Univerza v Ljubljani
Fakulteta za elektrotehniko

Borut Wagner

Izračun stacionarnega stanja električnih vezij in njegova uporaba pri optimizacijskih postopkih

DOKTORSKA DISERTACIJA

Ljubljana, 2008

Univerza v Ljubljani
Fakulteta za elektrotehniko

Borut Wagner

Izračun stacionarnega stanja električnih vezij in njegova uporaba pri optimizacijskih postopkih

DOKTORSKA DISERTACIJA

Mentor: prof. dr. Tadej Tuma

Ljubljana, 2008

Zahvala

Zahvaljujem se mentorju prof. dr. Tadeju Tumi, ki me je vodil skozi podiplomski študij, katerega rezultat je ta doktorska disertacija.

Sodelavcem iz Laboratorija za računalniške metode v elektroniki, še posebej Árpádu Bűrmenu, Janezu Puhanu in Branku Ždralu, se zahvaljujem za vse nasvete, namige in pripombe, ki so pripomogli k izdelavi in dosegu ciljev doktorske disertacije.

Zahvala tudi vsem domačim, ki so mi omogočili študij in me pri tem tudi moralno podpirali in vzpodbujali.

Zahvaljujem se tudi Ministrstvu za visoko šolstvo, znanost in tehnologijo, Javni agenciji za raziskovalno dejavnost RS za sofinanciranje podiplomskega študija in Fakulteti za elektrotehniko, kjer sem bil v času podiplomskega študija zaposlen kot mladi raziskovalec.

Povzetek

Doktorska disertacija obravnava izračun stacionarnega stanja električnih vezij z uporabo programskega paketa SPICE, ki ga uporabljamo kot orodje pri načrtovanju integriranih vezij. Prikazana je uporaba paketa SPICE pri optimizaciji električnih vezij, s poudarkom na problemih, ki izvirajo iz izračuna stacionarnega stanja. Stacionarno stanje vezja je stanje, ko vsi prehodni pojavi izzvenijo in je vezje v dinamičnem ravnovesju. S pojmom *stacionarno stanje* je mišljeno stanje vezja, katerega spremenljivke (frekvenca, amplituda, faza) se v časovnem prostoru ne spreminjanjo, in ne stanje, ko se trenutne vrednosti posameznih signalov ne spreminjajo v odvisnosti od časa. Pojem *stacionarno stanje* se v doktorski disertaciji torej nanaša na *periodično stacionarno stanje* električnega vezja.

Pri izračunu stacionarnega stanja električnih vezij se pri določenih vezjih hitro srečamo s problemom dolgotrajnega izračuna, ki lahko traja več minut ali celo ur. Delo obravnava metode, s pomočjo katerih stacionarno stanje izračunamo hitreje kot z uporabo običajne tranzientne analize, ki se mora izvajati tako dolgo, da vsi prehodni pojavi izzvenijo. Z uporabo teh postopkov se izračun stacionarnega stanja močno pospeši. Pospešitev je odvisna od vrste vezja, ki ga analiziramo. Tako je v doglednem času mogoče vezje tudi optimizirati, pri čemer je potreben izračun več 1.000 ali 10.000 analiz stacionarnega stanja.

Začetno poglavje obravnava matematičen opis električnih vezij in način izvajanja analiz. Naštete so vrste analiz, ki jih lahko izvajamo s pomočjo programskega paketa SPICE. Poudarek je na tranzientni analizi, ki je osnova za izračun stacionarnega stanja. Tranzientna analiza je od vseh analiz računsko najbolj zahtevna. Zaradi tega je za mnoga vezja izračun stacionarnega stanja zelo dolgotrajen proces. Prikazani so problemi, ki se lahko pojavijo pri izračunu stacionarnega stanja. Podani so primeri vezij, pri katerih se lahko ti problemi pojavijo. Na koncu poglavja je opisana optimizacija električnih vezij, kjer želimo s spreminjanjem lastnosti vezja doseči, da bi vezje kar najbolje izpolnjevalo postavljene zahteve. Znotraj optimizacijske zanke se lahko izvajajo različne analize, med drugim tudi izračun stacionarnega stanja. Nakazana in ocenjena je dolgotrajnost optimizacije, kadar je že sam izračun stacionarnega stanja dolgotrajen.

Tretje poglavje opisuje metode, s pomočjo katerih lahko na hitrejši način izračunamo staci-

onarno stanje. Med njimi so najpomembnejše metoda harmoničnega ravnovesja in razne ekstrapolacijske metode. Pri prvi metodi vezje, za katerega iščemo stacionarno stanje, razdelimo na linearni in nelinearni del. Na vozliščih, kjer se dela stikata, moramo uskladiti napetosti in tokove, da sta izpolnjena Kirchhoffova zakona. S pomočjo ekstrapolacijskih postopkov izračunano stacionarno stanje tako, da najprej vezje analiziramo nekaj period osnovnega signala. Na podlagi dobljenega odziva sklepamo na stacionarno stanje oz. postavimo začetno stanje vezja za novo iteracijo analize. Prikazana je posplošitev metode za vezja, ki niso vzbujana (npr. oscilatorji).

V naslednjem poglavju je utemeljena izbira metode za izračun stacionarnega stanja. Metode za izračun stacionarnega stanja so bile najprej simulirane v programskem jeziku NUTMEG paketa SPICE in s pomočjo klicev običajne tranzientne analize iz zunanjega programa. Za izbor metode, ki je bila vgrajena v programski paket SPICE, so bili izbrani trije kriteriji. Na petih testnih vezjih so bile iz rezultatov simulacije metod določene vrednosti treh kriterijev. Na podlagi teh vrednosti se je kot najboljši izkazal algoritem epsilon. Algoritem je bil nato implementiran v programski paket SPICE OPUS.

Peto poglavje je namenjeno testiranju metode za izračun stacionarnega stanja na več realnih primerih električnih vezij. Rezultati analize stacionarnega stanja so primerjani z rezultati, ki jih dobimo, če vezje analiziramo z uporabo običajne tranzientne analize, s čimer je potrjena tudi točnost izračuna. Primerjan je čas, potreben za izračun stacionarnega stanja po obeh postopkih. Metoda je bila testirana tudi na primeru optimizacije vezja.

Zadnje poglavje povzema namene in cilje doktorske disertacije, način izpolnitve le-teh in rezultate. Našteti so izvirni prispevki k znanosti, članek, s katerim je bila pridobljena pravica za neposredni prehod na doktorski študij, in članek, objavljen v reviji s faktorjem vpliva po SCI, ki vsebuje del rezultatov raziskav, ki so obravnavani v tej doktorski disertaciji.

Abstract

Computing Steady State of Electrical Circuits and its Application to Optimization Algorithms

The thesis focuses on the computation of the steady state of electrical circuits by means of the SPICE circuit simulator and its application to optimization of electrical circuits with emphasis on problems that require the computation of the steady state. The steady state of electrical circuits is reached when all the transients fade out. The term *steady state* is used for the state of a circuit where the variables (frequency, amplitude and phase) do not change in the time domain, and not for the state where the response of the circuit is constant with respect to time. In this thesis the term *steady state* is used for the *periodic steady state* of the electrical circuit.

For certain nonlinear electrical circits the steady state is difficult to evaluate. The computation can take minutes or even hours before accurate results are obtained. With special methods the steady state can be obtained faster than by means of direct transient analysis. Depending on the type of the circuit the computation can be accelerated many times. The purpose of the thesis is also to speed-up the optimization of electrical circuits for which steady state has to be computed. Typically, more than 1,000 or 10,000 iterations are needed in the optimization procedure so that fast steady-state analysis can also speed-up the optimization.

The thesis starts with the mathematical background of electrical circuits and analyses. The circuit simulator SPICE and all main analyses are considered. Extra attention is paid to transient analysis which is required for steady state computation through the direct approach. The transient analysis is computationally very demanding, so the steady state computation of certain circuits takes a considerable amount of time. The problems in the steady-state response evaluation of electrical circuits are exposed. Cases of electrical circuits where problems manifest are given. Finally the optimization of electrical circuits is described. The main goal of the optimi-

zation is to find the best circuit that satisfies all the design requirements. The optimization is performed by changing the circuit parameters. Different circuit analyses are performed within the optimization process. We demonstrate that if the steady-state response analysis is difficult to compute, the optimization takes a very long time.

With methods described in Chapter 3 the steady-state response can be computed faster than by using the direct transient analysis. Most notable are the harmonic balance method and various extrapolation methods. In the harmonic balance method the circuit is divided into a linear and a nonlinear subcircuit. The voltage and the current Kirchhoff laws have to be satisfied at the nodes common to both subcircuits. When the steady state is computed using extrapolation methods the circuit is analyzed for several periods of signals that are present in the circuits. From the obtained response a new initial condition is set for the next iteration of the extrapolation algorithm. The algorithm is generalized for autonomous circuits (i.e. oscillators).

In the next chapter the choice of the steady-state response method is explained. All candidate methods were first simulated using the SPICE NUTMEG programming language and an external program. The choice of the method that was later implemented in SPICE was determined using three criteria. The criteria were evaluated for five test circuits. The best choice for the implementation was the epsilon algorithm.

Chapter 5 is devoted to the testing of the steady-state evaluation method on several real-world cases of electrical circuits. Steady-state responses are compared to the results obtained by direct transient analysis. Comparisons confirm that the results are virtually identical. The time needed for computing the steady state by both methods was compared. The method was also tested on the optimization of a test circuit.

Chapter 6 summerizes the purpose and the goals of the thesis. The results of the thesis are given and the original scientific contributions are presented.

Kazalo

Slike

Tabele

1

Uvod

Pogosto nas pri analizi električnega vezja zanima stanje vezja po dovolj dolgem času, ko se vsi prehodni pojavi iznihajo. Tedaj v vezju ostanejo signali, katerih vrednosti (amplituda, frekvenca, faza) se v frekvenčnem prostoru ne spreminjajo. To stanje v splošnem imenujemo *stacionarno stanje električnega vezja*. V nadaljevanju bo s pojmom *stacionarno stanje* mišljeno stacionarno stanje glede na frekvenčni prostor oz. *periodično stacionarno stanje* (pri katerem imajo signali v vezju vsaj eno frekvenčno komponento), ne pa stanje vezja, kjer se s časom ne spreminjajo tudi trenutne vrednosti signalov (stacionarno stanje glede na časovni prostor) - signali v vezju imajo samo enosmerno komponento.

Vezje, ki ga analiziramo, je lahko vzbujano (npr. filtri, ojačevalniki) ali nevzbujano (npr. oscilatorji). Pri obeh tipih vezij imajo lahko napetosti, tokovi in druge količine v vezju enosmerno komponento, največkrat pa še eno ali več izmeničnih komponent. Za vezja, ki so vzbujana ali v katerih predpostavljamo t.i. majhne signale, lahko najprej izračunamo delovno točko, vezje okoli te točke lineariziramo in s pomočjo malosignalne izmenične analize poiščemo odziv vezja oz. izmenično prenosno karakteristiko vezja. S pomočjo te karakteristike lahko na podlagi določenega vzbujanja vezja določimo stacionarno stanje vezja, če vhodni signal vsebuje samo eno frekvenčno komponento. Opisan postopek velja samo za majhne signale. Amplituda majhnih signalov je odvisna od natančnosti, ki jo želimo kot rezultat analize. Če je nelinearnost vezja okoli delovne točke vezja velika, mora biti amplituda signalov ustrezno majhna, da dosežemo ustrezno natančnost izračuna.

Če vnaprej ne poznamo lastnosti vezja, ki ga analiziramo, če želimo večjo natančnost analize ali če v vezju nastopajo relativno veliki signali, je potrebno stacionarno stanje električnega vezja izračunati s pomočjo tranzientne analize. S tranzientno analizo moramo vezje analizirati tako dolgo, da vsi prehodni pojavi izzvenijo. Za končanje tranzientne analize je potrebno upoštevati določen ustavitveni kriterij s podano toleranco. Ustavitveni kriterij preverja stanje vezja zadnjih nekaj period signalov v vezju oz. zadnjih nekaj časovnih enot. Ko rezultat analize doseže ustavitveni kriterij, je vezje (z določeno toleranco) v stacionarnem stanju.

Izračun stacionarnega stanja vezja, ki ima glede na periode signalov v vezju dolge časovne konstante, je zelo dolgotrajen. Zaradi relativno majhne periode in relativno velike časovne konstante je potrebno izračunati dovolj časovnih točk na periodo in vezje analizirati preko veliko period. Tipično je za taka vezja potrebno izračunati več milijonov ali celo deset milijonov časovnih točk, da vezje doseže stacionarno stanje. Odvisno od vrste in obsežnosti vezja lahko tak izračun traja od nekaj minut do nekaj ur.

Pri vezjih, ki so vzbujana z več vhodnimi signali (npr. mešalniki), se situacija še poslabša, ker je osnovna frekvenca tako enaka razliki vhodnih frekvenc. Če je vezje vzbujano z dvema signaloma, katerih frekvenci sta zelo blizu, je osnovna frekvenca zelo majhna v primerjavi s frekvencama vzbujajočih signalov. V vezju imamo prisotne tako signale nizkih kot tudi visokih frekvenc. Časovni korak tranzientne analize moramo izbrati glede na signal z najvišjo frekvenco (vsak nekaj točk na periodo signala z najvišjo frekvenco), osnovna perioda pa je določena z najnižjo frekvenco signala.

Zgoraj opisani problemi pri analizi točno določenega vezja še ne predstavljajo tako velike ovire. Poženemo tranzientno analizo in počakamo, da računalnik izračuna stacionarno stanje. Problem nastopi pri računalniškem načrtovanju vezij, kjer želimo s spreminjanjem lastnosti vezja (topologija vezja, vrednosti elementov) doseči, da vezje izpolni določene zahteve (optimizacijski postopek). V tem primeru je treba velikokrat izračunati stacionarno stanje vezja, vselej z drugačnimi lastnostmi vezja. V vsaki iteraciji tega optimizacijskega postopka je potrebno oceniti, kako dobro vezje izpolnjuje zahteve. Postopek moramo ponavljati, dokler vezje ne izpolni vseh zahtev. Tipično tak optimizacijski postopek zahteva nekaj tisoč ali deset tisoč iteracij, dokler ne najdemo vezja, ki izpolnjuje vse zahteve.

Za vezja, ki potrebujejo za posamezen izračun staconarnega stanja nekaj deset minut, lahko tako poiščemo optimalno vezje v nekaj deset dneh ali več. Zaradi potrebe po čimvečji konkurenčnosti in čim hitrejšem razvoju integriranih vezij nas ta čas ne zadovolji. Pri tem je treba še poudariti, da večina danajšnjih orodij za optimizacijo izvaja t.i. parametrsko optimizacijo, t.j. spreminja samo parametre elementov, ne pa tudi topologije. V primeru, da s tako optimizacijo ne dosežemo vseh zahtev, moramo spremeniti še topologijo vezja in ponovno pognati parametrsko optimizacijo. Celotnen postopek iskanja ustreznega vezja (parametrska optimizacija s pomočjo računalnika in ročno spreminjanje topologije) lahko tako traja več mesecev.

Postopek optimizacije se da skrajšati na več načinov. Prvi način je uporaba zmogljivejših računalnikov, večprocesorskih računalnikov in/ali uporaba paralelnega procesiranja. Ta način v vsakem primeru zahteva več vložene procesorske moči za izračun določenega problema. Pri tem produkt *procesorske moči* in *časa izračuna* ne spreminja veliko. Drugi način pa je uporaba algoritmov, ki pri isti procesorski moči izračunajo stacionarno stanje hitreje kot z uporabo običajne tranzientne analize.

Doktorska disertacija je namenjena slednjemu načinu. S pomočjo posebnih postopkov je mogoče določiti stacionarno stanje vezja, ne da bi morali vezje analizirati z uporabo direktne tranzientne analize, kot je bilo opisano v prejšnjih odstavkih. Doktorska disertacija najprej obravnava analizo električnih vezij. Predstavljene so vrste analiz, ki jih lahko izvajamo s programskim paketom SPICE [1, 2, 3] za načrtovanje integriranih vezij. Poudarek je na tranzientni analizi, s katero na običajen način določimo odziv vezja v stacionarnem stanju. Prikazan je problem dolgotrajnega izračuna stacionarnega stanja. Sledi opis metod za hitrejši izračun stacionarnega stanja, utemeljitev izbire določene metode, ki je bila implementirana v programski paket SPICE OPUS. Metoda je bila testirana na več realnih električnih vezjih. Prikazana je računska zahtevnost in točnost izračuna stacionarnega stanja in optimizacije vezja.

2

Analiza električnih vezij in izračun stacionarnega stanja

2.1 Matematična obravnava in opis električnih vezij

Doktorska disertacija temelji na enem izmed najuspešnejših programov za analizo (integriranih) električnih vezij - SPICE3 [2], ki je bil razvit na Univerzi Berkeley, oz. na njegovi različici SPICE OPUS [1, 3].

Vezje, ki ga želimo analizirati, opišemo z vhodno datoteko, ki jo lahko napišemo s katerim koli urejevalnikom besedil ali pa ustvarimo s programom za shematski vnos vezij. Vhodna datoteka vsebuje modele elementov, elemente in njihove medsebojne povezave. Elementi vezja (tranzistorji, diode, induktivnosti, kapacitivnosti, upori, napetostni in tokovni viri...) so opisani z diferencialnimi enačbami. Vezje lahko analiziramo v časovnem ali frekvenčnem prostoru. V časovnem prostoru je sistem nelinearnih diferencialnih enačb rešen numerično z upoštevanjem začetnega stanja vezja. V frekvenčnem prostoru vezje linearziramo v delovni točki. Rezultat tovrstne analize je frekvenčna karakteristika vezja pri vzbujanju z majhnimi signali.

Osnova za izračun stacionarnega stanja električnih vezij je tranzientna oz. časovna analiza. Pri tem postopku rešujemo sistem navadnih diferencialnih enačb

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t), \tag{2.1}$$

ki jih dobimo po obdelavi vhodne datoteke analiziranega vezja [4, 5]. Pri reševanju teh enačb upoštevamo začetno stanje $\mathbf{x_0}(0)$. Vektor $\mathbf{x}(t)$ predstavlja vozliščne napetosti in vejne tokove vezja. Enačbe rešujemo numerično z uporabo Newton-Raphsonove iteracijske metode in integracijskih algoritmov [2, 6]. Rešitev predstavlja vektor $\mathbf{x}(t)$ za čase $t > 0$. Podaja časovni potek stanja vezja oz. vozliščne napetosti in vejne tokove kot funkcijo časa. Ta odziv predstavlja tudi osnovo za določitev stacionarnega stanja vezja.

V nadaljevanju bodo predstavljene analize vezja, ki jih lahko izvajamo s programskim paketom SPICE OPUS. Med njimi je računsko najbolj zahtevna tranzientna (časovna) analiza.

2.2 Vrste analiz električnih vezij

S programskim paketom SPICE lahko izvajamo različne analize vezja (analiza enosmerne delovne točke, izračun enosmerne prenosne funkcije, izračun diferencialne enosmerne prenosne funkcije, izmenična analiza, šumna analiza, analiza polov in ničel). Glavna analiza, na kateri temelji doktorska disertacija, je tranzientna analiza, in bo predstavljena v nadaljevanju. Podrobnejši opis ostalih analiz pa lahko najdemo v [7] in [8].

2.2.1 Tranzientna analiza

Časovna ali tranzientna analiza deluje v časovnem prostoru. Začetne pogoje ob času $t = 0$ podamo s stavkom *ic*. Če le-teh ne podamo, se avtomatično izračunajo z analizo enosmerne delovne točke. Reaktančni elementi se nato nadomestijo z diskretnimi modeli, tako da nastane nelinearno rezistivno vezje, katerega rešitev predstavlja odziv vezja po nekem kratkem časovnem intervalu. Za določitev diskretnih modelov se uporabi integracijski algoritem. Po vsakem časovnem koraku se le-ta ustrezno popravi in prilagodi dinamiki vezja.

Osnova tranzientne analize je stavek *tran*

$$tran \quad t_\Delta \quad t_k \quad [\ t_z\ [\ t_m\ ]] \quad [UIC]\ ,$$

ki določa časovni potek. Analiza traja od časa $t = 0$ do $t = t_k$. Izpisni korak je določen s t_Δ in sicer od t_z do t_k. Vgrajena vrednost za t_z je 0. Računski korak (ki ni enak izpisnemu koraku) ni konstanten, saj se avtomatsko prilagaja odzivu vezja, uporabnik pa lahko določa le njegovo zgornjo mejo t_m. Če zgornja meja računskega koraka ni podana, potem velja $t_m = \min(t_\Delta, (t_k - t_z)/50)$.

Rezervirana beseda *UIC* (Use Initial Conditions) se uporablja v povezavi s stavkom *ic* ter s parametri *ic* elementov vezja. Če uporabimo rezervirano besedo *UIC*, se slednji upoštevajo kot začetno stanje vezja.

Problem tranzientne analize je dejstvo, da vsak nadaljnji časovni korak izhaja iz rezultatov prejšnjega. To pomeni, da se napake akumulirajo. SPICE3 ima vgrajena dva mehanizma, ki se vsak na svoj način borita proti nabiranju numeričnih napak.

Prvi mehanizem se imenuje avtomatično nastavljanje računskega časovnega koraka. Če SPICE3 zazna, da je odziv vezja preživahen, skrajša časovni korak. Tako na eni strani natančneje sledi spremembam spremenljivk v vezju, na drugi strani se pa lahko zato poveča trajanje analize in seveda obremenitev računalnika. Uporabnik nima neposrednega vpliva na mehanizem avtomatskega nastavljanja časovnega koraka.

Drugi mehanizem, ki ga SPICE3 ponuja, je možnost izbire integracijskega algoritma. Osnovi trapezoidni algoritem se naslanja na predhodno rešitev vezja in na osnovi tega določi naslednjo. V stavku *options* lahko preklopimo s parametrom *method* na Gearov integracijski algoritem. V tem primeru SPICE3 določi nadomestno vezje na osnovi rezultatov iz zadnjih nekaj časovnih točk. To omogoča daljši časovni korak in večjo natančnost, vendar pa Gearov algoritem višjega reda zelo občutljivo reagira na časovne nezveznosti, saj sega globlje v zgodovino signala kot trapezoidni algoritem. Izbira Gearovega integracijskega algoritma je zelo zahtevna tudi zaradi začetnih vrednosti in je priporočljiva le v posebnih primerih.

2.3 Stacionarno stanje z uporabo tranzientne analize

Najprej definirajmo *periodično stacionarno stanje* vezja. Vezje vsebuje signale

$$\mathbf{x}(t) = \left[\, x_1(t) \quad x_2(t) \quad \cdots \quad x_n(t) \,\right]^T . \tag{2.2}$$

Vezje je po dovolj dolgem času $t > t_{ss}$ v *periodičem stacionarnem stanju*, če obstaja osnovna perioda signalov T in če velja

$$x_i(t) = x_i(t+T), \quad t > t_{ss}, \quad i = 1, 2, \ldots, n . \tag{2.3}$$

Poseben primer periodičnega stacionarnega stanja je *enosmerno stacionarno stanje*, za katerega velja

$$\lim_{T \to 0} x_i(t) = x_i(t+T), \quad t > t_{ss}, \quad i = 1, 2, \ldots, n . \tag{2.4}$$

Posledica definicije je, da signali vezja $x_i(t)$ vsebujejo je enosmerne komponente. Kot je bilo poudarjeno že v Uvodu, v doktorski disertaciji takih primerov stacionarnih stanj ne bomo obravnavali, ampak le primere periodičnih stacionarnih stanj. V nadaljevanju bo torej s pojmom stacionarno stanje mišljeno periodično stacionarno stanje.

Stacionarno stanje s programskim paketom SPICE določimo s pomočjo tranzientne analize. Vezje analiziramo dovolj dolgo, da prehodni pojavi izzvenijo. Za vezje, ki vsebuje signale s periodo T, bi morali izvesti tranzientno analizo s časovnim korakom t_Δ za zelo velik čas t

$$tran \quad t_\Delta \quad t \quad t - 2T \ . \tag{2.5}$$

Rezultat te tranzientne analize je vektor spremenljivk vezja

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) & x_2(t) & \cdots & x_n(t) \end{bmatrix}^T, \qquad 0 \leq t \leq 2T. \tag{2.6}$$

Vezje je v stacionarnem stanju, zato za vsak $i = 1, 2 \ldots n$ in za vse $t_{ss} - 2T \leq t \leq T_{ss} - T$ velja

$$x_i(t) = x_i(t + kT), \qquad k = 1, 2, 3, \ldots \ . \tag{2.7}$$

Pri računalniški analizi vezij se zadovoljimo s približkom stacionarnega stanja. Tranzientna analiza (2.5) se tako izvaja samo do določenega časa t_{ss}, pri katerem za vse $0 \leq t \leq T$ velja

$$|x_i(t) - x_i(t + T)| < \delta \qquad i = 1, 2 \ldots n, \tag{2.8}$$

pri čemer je $\delta > 0$ konstanta, ki določa toleranco oz. natančnost izračunanega stacionarnega stanja vezja.

Algoritem 2.1 predstavlja izračun stacionarnega stanja s pomočjo tranzientne analize.

izberi $\delta > 0$
izvedi tranzientno analizo za čas T
$t_{ss} := T$
do
    podaljšaj tranzientno analizo za dodaten čas T
    $t_{ss} := t_{ss} + T$
while $|x_i(t) - x_i(t + T)| > \delta$ za vsaj en $i = 1, 2 \ldots n$, $\quad t_{ss} - 2T \leq t \leq t_{ss} - T$

Algoritem 2.1. Določitev stacionarnega stanja

2.3.1 Analiza zahtevnosti in problemi pri izračunu stacionarnega stanja

Enačba (2.8) in Algoritem 2.1 določajo, da moramo vezje analizirati do časa t_{ss}, da dobimo stacionarno stanje v okviru natančnosti δ. Čas t_{ss} je odvisen od vrste vezja, za katerega računamo stacionarno stanje.

Za dosego zadovoljive numerične natančnosti moramo izračunati nekaj deset do nekaj sto časovnih točk na periodo T signalov v vezju. Če je časovna konstanta približevanja stacionarnemu stanju τ mnogo večja v primerjavi s periodo T, je treba izračunati več 100.000 časovnih točk, da dosežemo stacionarno stanje.

Predpostavimo, da lahko ovojnico $x(t)$ signala v vezju opišemo z enačbo

$$x(t) = x_0 + (x_\infty - x_0)(1 - e^{-t/\tau}), \tag{2.9}$$

če ovojnica signala narašča proti vrednosti x_∞, oz.

$$x(t) = x_\infty + (x_0 - x_\infty)e^{-t/\tau}, \tag{2.10}$$

če ovojnica signala pada proti vrednosti x_∞. Pogoja, ki določata, da je ovojnica signala z relativno natančnostjo δ_r oddaljena od stacionarnega stanja x_∞, sta podana z enačbama za prvi primer

$$\frac{x_\infty - x(t)}{x_\infty - x_0} < \delta_r \tag{2.11}$$

in za drugi primer

$$\frac{x(t) - x_\infty}{x_0 - x_\infty} < \delta_r. \tag{2.12}$$

V obeh primerih je potrebno za zahtevano relativno natančnost δ_r vezje analizirati

$$t/\tau > -ln\delta_r \tag{2.13}$$

časovnih konstant. Za relativno natančnost npr. 10^{-4} je tako število časovnih konstant enako 9,2.

Pri vezjih, kjer je časovna konstanta primerljiva z osnovno periodo signalov v vezju, je izračun stacionarnega stanja relativno hiter. Problem pa nastopi, če je časovna konstanta mnogo večja v primerjavi z osnovno periodo. V Poglavju 5.2 so predstavljeni primeri vezij, kjer je časovna konstanta tudi več tisočkrat večja od periode signalov.

Primer
Izračunati želimo sto časovnih točk na periodo,
časovna konstanta je tisočkrat večja od periode signalov,
želimo relativno natančnost izračuna 10^{-4}.
Potrebujemo izračun 100 · 1000 · 9,2 = 920.000 časovnih točk.

V Poglavju 5.2 so podani tudi časi, ki so bili na povprečnem osebnem računalniku potrebni za izračun stacionarnega stanja s podano natančnostjo. Ti časi imajo velikostni razred deset sekund oz. minute. V Poglavju 2.4 so predstavljeni optimizacijski postopki, kjer se za dosego zahtevanih lastnosti vezja v optimizacijski zanki izvajajo analize vezja. Če moramo v zanki izračunavati stacionarno stanje vezja, ki traja nekaj minut, je taka optimizacija zelo dolgotrajna. Z metodami, ki so predstavljene v Poglavju 3, lahko stacionarno stanje izračunamo mnogo hitreje.

2.4 Optimizacijski postopki pri električnih vezjih

V tem poglavju bomo na kratko opisali optimizacijske postopke in njihovo uporabo pri električnih vezjih. Optimizacijski postopki so podrobneje opisani v [9] v Poglavju 2, [10] in tudi v [11]–[21].

Optimizacijski postopek pri električnih vezjih je iskanje vezja, ki najbolje izpolnjuje določene zahteve in ima zahtevane lastnosti [22]. Postopek se običajno izvede z iskanjem minimuma kriterijske funkcije f

$$f : \mathcal{D} \mapsto \mathbb{R}, \quad \mathcal{D} \subseteq \mathbb{R}^n$$

na območju $\mathcal{D}$. n-terico realnih števil, ki določajo točko iz območja $\mathcal{D}$, imenujemo optimizacijski parametri. Pri analizi električnih vezij so lahko optimizacijski parametri vrednosti posameznih elementov vezja. Običajno kriterijska funkcija ni analitično izražena z optimizacijskimi parametri. V praksi se vrednost kriterijske funkcije izračuna iz rezultatov analize vezja. Glede na to, katere informacije o kriterijski funkciji imamo na voljo, lahko optimizacijske postopke razdelimo na dve veliki družini: na gradientne postopke in direktne postopke. Gradientni postopki zahtevajo poleg same vrednosti kriterijske funkcije tudi enega ali več odvodov kriterijske funkcije po vseh optimizacijskih parametrih. Ker v praksi dobimo vrednost kriterijske funkcije na podlagi analiz vezja, lahko vrednosti odvodov izračunamo numerično na osnovi diferenc, kar pa s sabo prinaša veliko numerično napako. Običajno nam ostanejo samo direktne metode, ki potrebujejo informacijo samo o vrednosti kriterijske funkcije v določeni točki, ne pa tudi odvodov.

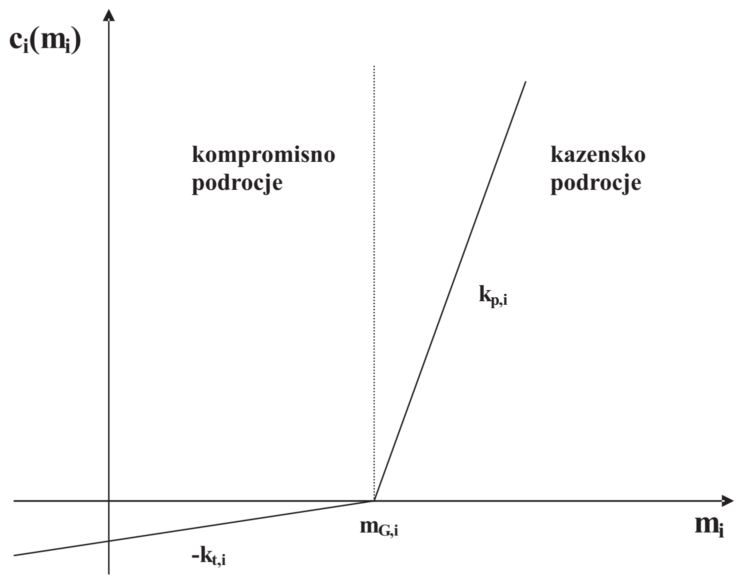
Kriterijska funkcija f je definirana kot vsota merilk c_i

$$f = \sum_i c_i(m_i(\mathbf{x}(\mathbf{p}))). \tag{2.14}$$

Merilke c_i so funkcije vrednosti meritev m_i. Meritve m_i določimo na podlagi analiz vezja $\mathbf{x}$ pri določenih vrednostih parametrov $\mathbf{p}$. Kriterijska funkcija f oz. merilke c_i so zapisane z uporabo kazenskih (penalty) in kompromisnih (trade-off) področij. Za vsako meritev m_i je podana načrtovalska zahteva $m_{G,i}$ (design goal) in podatek, ali naj bo meritev m_i manjša ali večja od načrtovalske zahteve $m_{G,i}$. Izberimo si primer, ko želimo, da je meritev manjša od načrtovalske zahteve (minimizacija načrtovalskih zahtev). Tedaj je merilka c_i je definirana kot

$$c_i(m_i) = \begin{cases} -k_{t,i} \cdot (m_{G,i} - m_i); & m_i < m_{G,i} \\ k_{p,i} \cdot (m_i - m_{G,i}); & m_i \geq m_{G,i} \,. \end{cases} \tag{2.15}$$

Del merilke, kjer velja $m_i < m_{G,i}$, imenujemo kompromisno področje (meritev leži v območju načrtovalske zahteve), drugi del ($m_i \geq m_{G,i}$) pa imenujemo kazensko področje (meritev leži zunaj načrtovalske zahteve). Slika 2.1 prikazuje odvisnost vrednosti merilke c_i od meritve m_i.



Slika 2.1. Odvisnost vrednosti merilke c_i v odvisnosti od meritve m_i

Naklon kompromisnega območja $k_{t,i}$ mora biti veliko manjši od naklona kazenskega področja $k_{p,i}$, kar pomeni, da v kriterijski funkciji prevladujejo doprinosi merilk v kazenskih področjih. To je tudi razumljivo, saj je naloga kazenske funkcije ustvariti zadovoljivo vezje, kompromisna funkcija pa le iz nabora zadovoljivih vezij izbere najboljšega. Naklon kompromisnega področja je tako velikokrat postavljen kar na 0 – zadovoljimo se že s prvim vezjem, ki izpolnjuje načrtovalske zahteve.

S tako definirano kriterijsko funkcijo f in na podlagi že izračunanih vrednosti kriterijske funkcije optimizacijski postopek išče nove točke prostora $\mathcal{D}$, ki imajo manjše vrednosti kriterijske funkcije. Iteracijski postopek se izvaja tako dolgo, dokler ne najdemo minimuma kriterijske funkcije. Minimum je lahko lokalen ali globalen, kar je odvisno od izbranega optimizacijskega postopka.

V vsaki iteraciji optimizacijskega postopka je potrebno za izračun vrednosti kriterijske funkcije f izračunati analize vezja $\mathbf{x}$ in na njih opraviti meritve m_i. Večino procesorske moči se porabi prav za analiziranje vezja, sama strategija optimizacijskega postopka potrebuje malo časa. Za skrajšanje optimizacijskega postopka moramo skrajšati čase, ki so potrebni, da vezje analiziramo in na rezultatih izvedemo določene meritve, ki so osnova za določitev kriterijske funkcije. V Poglavju 2.3.1 je bilo pokazano, da lahko traja določitev stacionarnega stanja vezja tudi do nekaj minut. Odvisno od števila optimizacijskih parametrov in izbranega optimizacijskega postopka potrebujemo tipično nekaj tisoč do nekaj deset tisoč iteracij optimizacijskega postopka. Zaradi dolgotrajnega izračuna stacionarnega stanja bo tako optimizacijski postopek trajal več deset dni.

S pomočjo optimizacijskih postopkov pri električnih vezjih dosežemo, da vezje izpolnjuje določene zahteve. Zgoraj opisanim optimizacijskim postopkom rečemo tudi parametrska optimizacija, saj spreminjamo samo parametre vezja, ne pa tudi njegove topologije. Če upoštevamo še dejstvo, da nam včasih pri določeni topologiji vezja optimizacijski postopek ne da zadovoljivih rezultatov, moramo spremeniti topologijo vezja (dodati, odvzeti kakšen element, spremeniti medsebojne povezave ...) in ponovno pognati parametrsko optimizacijo. Spreminjanje topologije vezja poteka zaenkrat še ročno, opravlja ga načrtovalec na podlagi znanja teorije vezij in prejšnjih izkušenj.

Celotni postopek, s katerim tako pridobimo delujoče vezje z željenimi lastnostmi, se zato podaljša še za faktor števila sprememb topologije vezja in se tako meri v mesecih. Za praktično uporabo je to predolgo, saj nas konkurenca sili k čim hitrejšem načrtovanju vezij. V Poglavju 3 so opisani algoritmi, ki pospešijo optimizacijske postopke s tem, da pospešijo analizo vezja, ki je računsko najbolj zahtevna. S tem postane optimizacijski postopek praktično uporaben.

Doktorska disertacija je specializirana za primere načrtovanja električnih vezij, kjer so izpolnjeni naslednji pogoji:

1. potrebna je optimizacija vezja,
2. kriterijska funkcija vsebuje meritve, ki slonijo na izračunu stacionarnega stanja,
3. vezje je neugodno - izračun stacionarnega stanja je dolgotrajen.

3

Stacionarno stanje električnih vezij

V tem poglavju bodo opisani postopki za hiter izračun stacionarnega stanja. Kot je bilo poudarjeno že v uvodu, se pojem *stacionarno stanje* nanaša glede na frekvenčni prostor. Uporaba postopkov za izračun stacionarnega stanje je zelo dobrodošla, če se vezje odziva zelo počasi glede na periodo vzbujanja in bi določitev stacionarnega stanja trajala zelo dolgo. Še posebno so ti postopki uporabni pri optimizaciji električnih vezij, kjer je znotraj optimizacijske zanke potreben izračun stacionarnega stanja takega vezja. V optimizacijskem postopku je potreben izračun več tisoč analiz stacionarnega stanja. Zato lahko sama optimizacija traja več tednov, kar nas pri načrtovanju električnih vezij zelo ovira. Na nek način moramo skrajšati celoten postopek optimizacije. To lahko storimo na več načinov. Uporabimo lahko hitrejše računalnike, večprocesorske računalnike ali vzporedno računanje z več računalniki hkrati [23]. V tem poglavju se bomo posvetili drugačnemu načinu pospešitve optimizacijskega postopka, t.j. uporabi postopkov, ki izračun staconarnega stanja pospešijo pri isti računski moči računalnika.

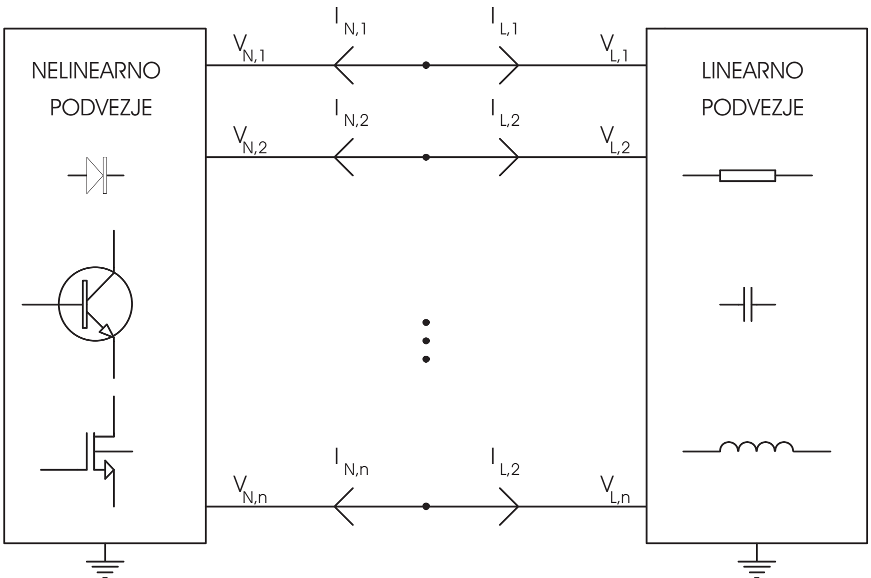
Vezje, ki vsebuje t.i. *majhne signale*, lahko analiziramo z malosignalno izmenično analizo. Rezultat je izmenična prenosna karakteristika. Iz vzbujanja vezja in karakteristike lahko določimo odziv vezja in s tem tudi stacionarno stanje vezja. Vezje lahko vzbujamo samo s signalom, ki ima eno frekvenčno komponento. Problem pri tem pristopu je, da se natančnost izračuna odziva manjša z amplitudo vzbujanega signala, saj je vezje linearizirano okrog delovne točke. Če so amplitude signalov v vezju velike, je ta linearizacija neprimerna oz. je tak izračun nenanančen. Tudi v primeru, če vezje vzbujamo s signalom, ki vsebuje več frekvenčnih komponent (npr. pravokotni signal), si z malosignalno izmenično analizo ne moremo pomagati, saj je vezje v splošnem nelinearno in zato ne moremo uporabiti superpozicije.

V teh primerih je za izračun stacionarnega stanja potrebno vezje analizirati s tranzientno analizo, ki pa je časovno in računsko zelo zahtevna.

V naslednjih poglavjih bomo predstavili, kako se pri izračunu stacionarnega stanja lahko izognemo uporabi računsko zahtevne direktne tranzientne analize, kljub temu pa ohranimo natančnost izračuna, ki jo ta analiza prinaša.

3.1 Metoda harmoničnega ravnovesja

Metoda harmoničnega ravnovesja (Harmonic Balance) [24]–[42] zahteva razdelitev vezja na del, v katerem so vsi nelinearni elementi, in del, ki vsebuje vse linearne elemente. Dela imenujemo nelinearno in linearno podvezje. Razdelitev prikazuje Slika 3.1.



Slika 3.1. Razdelitev vezja na nelinearni in linearni del

Vezje zaradi delitve navzven izkazuje n vozlišč. Število teh vozlišč je odvisno od števila elementov v vezju in medsebojnih povezav med elementi. Vezje v splošnem analiziramo z izbrano analizo. Na vozliščih nelinearnega dela so prisotni signali napetosti $V_{N,j}$ in tokov $I_{N,j}$, $j = 1, 2, 3 \ldots n$. Podobno je na vozliščih linearnega dela vezja, kjer so prisotne napetosti $V_{L,j}$ in tokovi $I_{L,j}$, $j = 1, 2, 3 \ldots n$. Zaradi kasnejše uporabe pri obravnavi metode harmoničnega ravnovesja so na Sliki 3.1 izbrane oznake, ki se uporabljajo v frekvenčnem prostoru (velike črke). Analogno bi lahko zamenjali oznake na Sliki 3.1 z oznakami v časovnem prostoru (npr. $V_{N,1} \to v_{N,1}(t)$), če bi vezje analizirali z uporabo tranzientne analize.

Signale v vezju splošno označimo z $x(t)$ in jih predstavimo v obliki Fourierjeve vrste

$$x(t) = \sum_{\omega_k \in \Omega} A_k cos(\omega_k t) + B_k sin(\omega_k t), \tag{3.1}$$

pri čemer so koeficienti A_k in B_k Fourierjevi koeficienti, ki pripadajo frekvencam ω_k. Ω je množica frekvenc, ki so vsebovane v signalih vezja. Vezje lahko v splošnem vzbujamo s signalom, ki vsebuje več frekvenc oz. z več signali. n frekvenc, ki so zastopane v vhodnih signalih, označimo z $\omega^1, \omega^2 \ldots \omega^n$. Predpostavimo, da je vezje, ki ga analiziramo, nelinearno. Zato množica Ω vsebuje tudi linearne kombinacije frekvenc $\omega^1, \omega^2 \ldots \omega^n$

$$\Omega = \left\{\omega, \quad \omega = k_1\omega^1 + k_2\omega^2 + \ldots + k_n\omega^n, \quad k_j \in \mathbb{Z}, \quad j = 1, 2 \ldots n\right\}. \tag{3.2}$$

Pri metodi harmoničnega ravnovesja moramo množico Ω omejiti, tako da vsebuje končno število elementov. Tako vsak k_i v (3.2) leži znotraj intervala

$$k_{j_{min}} \leq k_j \leq k_{j_{max}}, \quad j = 1, 2 \ldots n \tag{3.3}$$

(eksplicitne omejitve) ali pa so izbrani $k_i \ldots k_j$ povezani z m implicitnimi omejitvami

$$\begin{gathered} f_m(k_j \ldots k_l) = 0, \quad k_j \ldots k_l \in K_m \subseteq \mathbb{Z} \\ \text{oz.} \\ f_m(k_j \ldots k_l) \leq 0, \quad k_j \ldots k_l \in K_m \subseteq \mathbb{Z}. \end{gathered} \tag{3.4}$$

Primer omejitve množice Ω v dveh dimenzijah (koeficienta k_1 in k_2) prikazujeta Sliki 3.2 (eksplicitne omejitve) in 3.3 (implicitne omejitve). Pike na Slikah 3.2 in 3.3 prikazujejo pare koeficientov k_1 in k_2, s katerimi je določena množica Ω v enačbi (3.2). V [26] je še več primerov, kako omejiti množico Ω, tako da bo vsebovala končno število elementov oz. frekvenc.

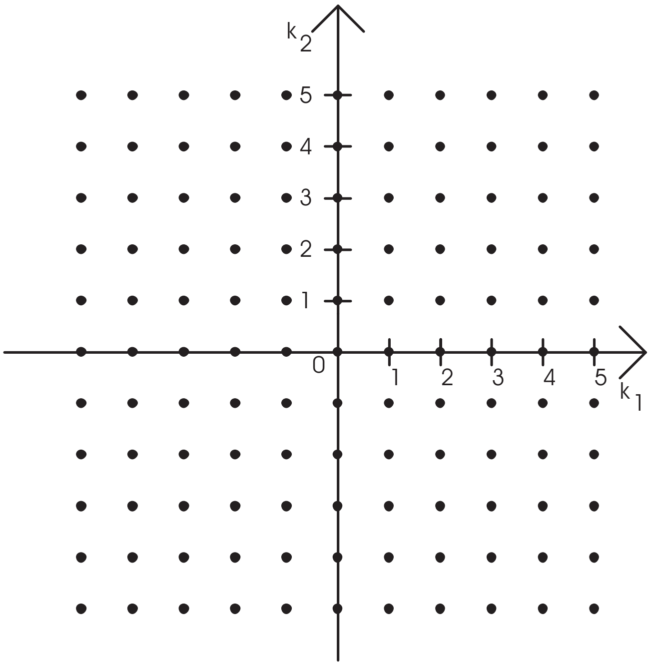
Signal, opisan z (3.1), je *skoraj periodičen*, če ga lahko predstavimo z vsoto števno mnogo frekvenčnih komponent in velja

$$\sum_{\omega_k \in \Omega} (A_k^2 + B_k^2) < \infty. \tag{3.5}$$

Pri metodi harmoničnega ravnovesja bomo na enačbi (3.1) oz. na signalih vezja izvajali Fourierjevo transformacijo

$$X(\omega) = \mathcal{F}\left\{x(t)\right\}, \qquad \omega \in \Omega, \tag{3.6}$$

saj večina postopka izračuna stacionarnega stanja z metodo harmoničnega ravnovesja poteka v frekvenčnem prostoru. Na vozliščih stičišča nelinearnega in linearnega podvezja mora veljati napetostni



Slika 3.2. Primer eksplicitne omejitve za k_1 in k_2 ($k_{1_{min}} = k_{2_{min}} = -5, \quad k_{1_{max}} = k_{2_{max}} = 5$)

$$V_{N,j} = V_{L,j}, \qquad j = 1, 2 \ldots n \tag{3.7}$$

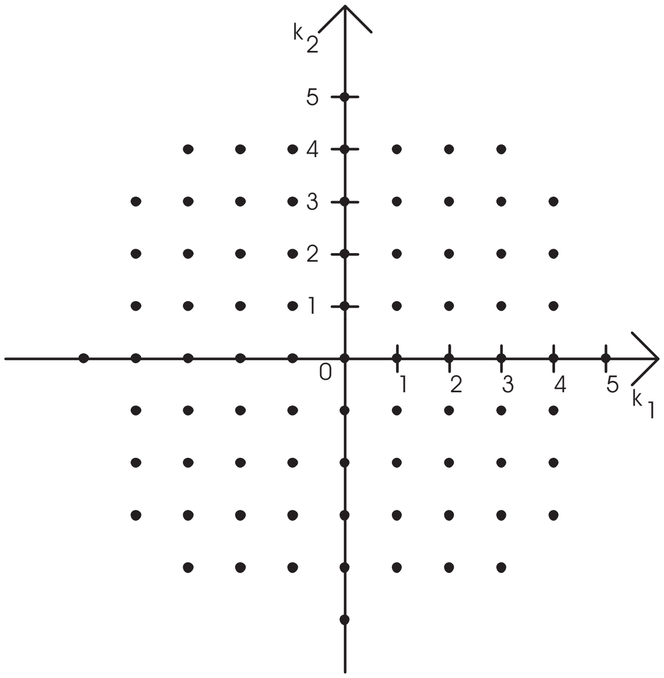
in tokovni

$$I_{N,j} = -I_{L,j}, \qquad j = 1, 2 \ldots n \tag{3.8}$$

Kirchhoffov zakon. Metoda harmoničnega ravnovesja je iteracijski postopek. Predpostavimo napetostni Kirchffov zakon in nato z iteracijskim postopkom izpolnimo še tokovni Kirchhoffov zakon oz. dosežemo, da je vrednost napake E_j enaka 0

$$E_j = I_{N,j} + I_{L,j} = 0, \qquad j = 1, 2 \ldots n \tag{3.9}$$

Linearno in nelinearno podvezje analiziramo ločeno. Na vozlišča obeh podvezij priključimo neodvisne napetostne vire $V_j = V_{L,j} = V_{N,j}, j = 1, 2 \ldots n$. Linearno podvezje analiziramo v frekvenčnem prostoru z malosignalno izmenično analizo. Ker je podvezje linearno, pri tem zaradi morebitne prisotnosti velikih signalov ne naredimo napake. Zaradi linearnosti lahko uporabimo tudi princip superpozicije. Rezultat analize so tokovi $I_{L,j}, j = 1, 2 \ldots n$, ki tečejo iz napetostnih virov v linearno podvezje (Slika 3.4).



Slika 3.3. Primer implicitne omejitve za k_1 in k_2 ($k_1^2 + k_2^2 - 25 \leq 0$)

Nelinearno podvezje moramo zaradi velikih signalov v vezju analizirati v časovnem prostoru. Vrednosti napetostnih virov so enake kot pri linearnem podvezju, $V_j = V_{L,j} = V_{N,j}, j = 1, 2 \ldots n$. Čas tranzientne analize je odvisen od izbire množice Ω v (3.2) oz. omejitev, podanih z (3.3) in/ali (3.4) in je sorazmeren obratni vrednosti najmanjšega skupnega delitelja elementov Ω. Ta zahteva je pogojena s kasnejšo uporabo Fourierjeve transformacije, ki zato poteka vedno preko celega števila period frekvenc v Ω. Kot rezultat tranzientne analize dobimo časovni potek tokov $i_j(t), j = 1, 2 \ldots n$. Izvedemo Fourierjevo transformacijo (3.6) in dobimo vrednost tokov $I_{N,j}, j = 1, 2 \ldots n$ v frekvenčnem prostoru, ki tečejo iz neodvisnih napetostnih virov $V_{L,j}, j = 1, 2 \ldots n$ v nelinearno podvezje.

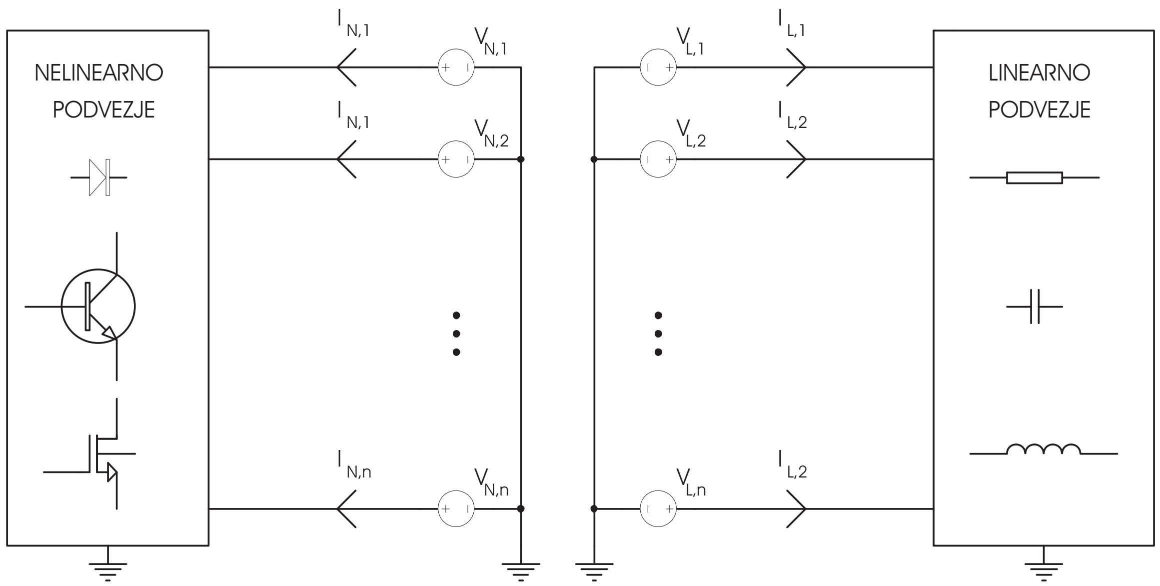
Celoten zgoraj opisani postopek analize linearnega in nelinearnega podvezja lahko predstavimo s funkcijo **F**, za katero želimo, da ima vrednost **0**

$$\mathbf{E} = \mathbf{F}(\mathbf{V}) = \mathbf{0}. \tag{3.10}$$

V (3.10) smo uporabili oznake vektorjev

$$\begin{aligned} \mathbf{E} = \mathbf{I_N} + \mathbf{I_L} = [I_{N,1} + I_{L,1} \ldots I_{N,n} + I_{L,n}]^T \\ \mathbf{V} = \mathbf{V_L} = \mathbf{V_N} = [V_{L,1} \ldots V_{L,n}]^T \qquad \text{in} \\ \mathbf{F} = [f_1(\mathbf{V}) \ldots f_n(\mathbf{V})]^T . \end{aligned} \tag{3.11}$$

Metoda harmoničnega ravnovesja se tako prevede v iskanje vektorja **V**, tako da bo izpol-



Slika 3.4. Priključitev napetostnih virov za iskanje stacionarnega stanja vezja pri metodi harmoničnega ravnovesja

njena enačba (3.10). Z vektorjem $\mathbf{V}$ so določeni tudi tokovi v linearno in nelinearno podvezje $\mathbf{I_L}$ in $\mathbf{I_N}$ in ostale količine v linearnem in nelinearnem podvezju (napetosti vozlišč, tokovi vej).

Rešitve enačbe (3.10) dobimo s pomočjo Newton-Raphsonovega iteracijskega postopka. Enačbo (3.10) najprej množimo s $\mathbf{K}(\mathbf{V}) \neq \mathbf{0}$, na obeh straneh prištejemo $\mathbf{V}$ in zamenjamo slevo in desno stran. Tako dobimo

$$\mathbf{V} = \mathbf{V} + \mathbf{K}(\mathbf{V})\mathbf{F}(\mathbf{V}). \tag{3.12}$$

Osnovna iteracijska enačba je podana z

$$\mathbf{V}^{(i+1)} = \mathbf{V}^{(i)} + \mathbf{K}(\mathbf{V}^{(i)})\mathbf{F}(\mathbf{V}^{(i)}), \tag{3.13}$$

(metoda fiksne točke), pri čemer zgornja indeksa (i) in $(i+1)$ pomenita številko iteracije. Iz enačbe (3.13) lahko izpeljemo Newton-Raphsonov postopek, če nadomestimo $\mathbf{K}(\mathbf{V}^{(i)})$ z $-\mathbf{J}^{-1}(\mathbf{V}^{(i)})$. $\mathbf{J}(\mathbf{V}^{(i)})$ predstavlja Jacobijevo matriko [43, 44], ki je splošno definirana kot

$$\mathbf{J}(\mathbf{V}) = \frac{\partial \mathbf{F}}{\partial \mathbf{V}} = \begin{bmatrix} \frac{\partial f_1(\mathbf{V})}{\partial V_1} & \cdots & \frac{\partial f_1(\mathbf{V})}{\partial V_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(\mathbf{V})}{\partial V_1} & \cdots & \frac{\partial f_n(\mathbf{V})}{\partial V_n} \end{bmatrix}. \tag{3.14}$$

Jacobijeva matrika mora biti nesingularna (obstajati mora $\mathbf{J}^{-1}(\mathbf{V})$).

Newton-Raphsonov postopek je tako določen z

$$\mathbf{V}^{(i+1)} = \mathbf{V}^{(i)} - \mathbf{J}^{-1}(\mathbf{V}^{(i)})\mathbf{F}(\mathbf{V}^{(i)}). \tag{3.15}$$

Postopek (3.15) ponavljamo, dokler napaka $\mathbf{E}^{(i)}$, ki je enaka $\mathbf{F}(\mathbf{V}^{(i)})$, ni manjša od vnaprej izbrane natančnosti izračuna δ

$$E = \left\|\mathbf{E}^{(i)}\right\| = \left\|\mathbf{F}(\mathbf{V}^{(i)})\right\| < \delta. \tag{3.16}$$

Velikost napake E je odvisna od izbire frekvenčnih komponent, ki so prisotne v vezju, oz. od elementov v Ω. Če izberemo prestroge omejitve (3.3) in (3.4) in premajhen δ, se lahko zgodi, da pogoj (3.16) tudi za poljubno velik i ne bo nikoli izpolnjen. V tem primeru moramo popustiti pri omejitvah in s tem povečati število elementov v Ω in/ali povečati δ.

Celotni algoritem iskanja stacionarnega stanja električnega vezja z metodo harmoničnega ravnovesja je prikazan v Algoritmu 3.1.

3.1.1 Modifikacija metode harmoničnega ravnovesja

Večina elementov vezja je v praksi, sploh pri analizi integriranih vezjih, nelinearnih. Zato lahko metodo harmoničnega ravnovesja modificiramo tako, da bo prilagojena temu dejstvu. Celotno vezje, za katerega računamo stacionarno stanje, proglasimo za nelinearni del. Nato v vsako od n vozlišč vezja priključimo dodaten neodvisni napetostni vir $V_{HB_i}, i = 1, 2 \ldots n$, vezan preko serijskega upora $R_{HB_i}, i = 1, 2 \ldots n$. Situacijo prikazuje Slika 3.5.

Stacionarno stanje vezja bo doseženo, ko bodo vsi tokovi $I_j = 0, \quad j = 1, 2 \ldots n$. Vektor napake $\mathbf{E}$ je sedaj definiran kot

$$\mathbf{E} = \mathbf{F}(\mathbf{V}_{HB}) = \mathbf{I} = \left[I_1, I_2 \ldots I_n\right]^T. \tag{3.17}$$

Newton-Raphsonov postopek za modificirano metodo harmoničnega ravnovesja je podoben kot pri (3.15), in sicer

$$\mathbf{V}_{HB}^{(i+1)} = \mathbf{V}_{HB}^{(i)} - \mathbf{J}^{-1}(\mathbf{V}_{HB}^{(i)})\mathbf{F}(\mathbf{V}_{HB}^{(i)}). \tag{3.18}$$

Upornosti $\mathbf{R}_{HB}$ morajo biti v vezju prisotne, da lahko z modificirano metodo harmoničnega ravnovesja na enak način analiziramo vsa vozlišča v vezju, tudi tista, kamor so priključeni neodvisni viri. Če uporov $\mathbf{R}_{HB}$ ne bi uporabili, bi s priključitvijo dodatnih vezij povzročili kratek stik med virom, ki je prisoten v vezju, in med dodatnim virom, ki ga priključimo za

1: Razdeli vezje na del, ki vsebuje vse nelinearne elemente,
in na del, ki vsebuje vse linearne elemente (Slika 3.1).
2: Izberi natančnost izračuna $\delta \in \mathbb{R}$ in maksimalno število iteracij $i_{max} \in \mathbb{N}^{+}$.
3: Izberi množico Ω (enačba (3.2)), ki predstavlja frekvenčne komponente signalov, ki nastopajo v vezju.
4: Nastavi $i = 0$ in $\mathbf{V}^{(0)} = \mathbf{0}$.
5: Na vozlišča nelinearnega podvezja in na vozlišča linearnega podvezja priključi neodvisne napetostne vire $\mathbf{V}^{(i)}$ (Slika 3.4).
6: Linearno podvezje analiziraj v frekvenčnem prostoru.
Rezultat analize je $\mathbf{I}_L^{(i)}$.
7: Nelinearno podvezje analiziraj v časovnem prostoru in uporabi Fourierjevo transformacijo. Rezultat analize je $\mathbf{I}_N^{(i)} = \mathcal{F}\left\{\mathbf{i}_N^{(i)}(t)\right\}$.
8: Izračunaj $\mathbf{E}^{(i)} = \mathbf{I}_L^{(i)} + \mathbf{I}_N^{(i)} = \mathbf{F}(\mathbf{V}^{(i)})$.
9: Izračunaj $\mathbf{J}(\mathbf{V}^{(i)}) = \frac{\partial \mathbf{F}(\mathbf{V}^{(i)})}{\partial \mathbf{V}^{(i)}}$.
10: Uporabi Newton-Raphsonovo iteracijsko formulo
$\mathbf{V}^{(i+1)} = \mathbf{V}^{(i)} - \mathbf{J}^{-1}(\mathbf{V}^{(i)})\mathbf{F}(\mathbf{V}^{(i)})$.
11: Povečaj i.
12: Če je $i > i_{max}$, nadaljuj pri koraku 2 s spremenjenimi vrednostmi δ, i_{max} in/ali Ω.
13: Če je $\left\|\mathbf{E}^{(i)}\right\| > \delta$, nadaljuj pri koraku 5.
14: Stacionarno stanje je doseženo z natančnostjo δ.
Iz $\mathbf{V}^{(i)}$ in $\mathbf{I}_L^{(i)}$ določi še ostale količine v nelinearnem in linearnem podvezju.

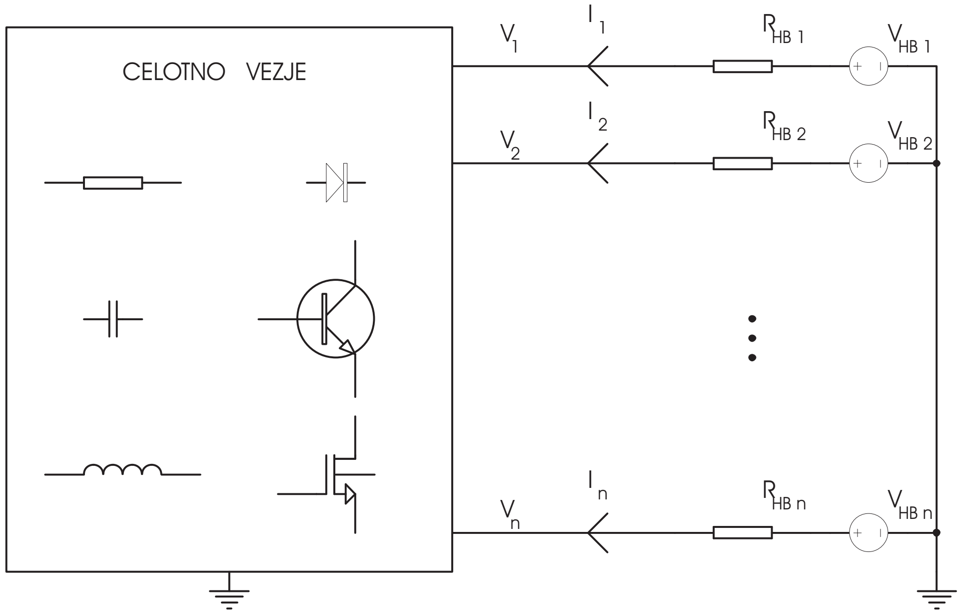
Algoritem 3.1. Izračun stacionarnega stanja električnega vezja z metodo harmoničnega ravnovesja

potrebe metode. S tem bi povzročili, da ne bi našli rešitve oz. ne bi mogli analizirati vezja niti za prvo iteracijo Newton-Raphsonovega postopka.

Začetne upornosti $\mathbf{R}_{HB}^{(0)}$ nastavimo na nekaj ali nekaj deset ohmov, odvisno od vezja, ki ga analiziramo. Nato z vsako iteracijo Newton-Raphsnovega postopka te upornosti manjšamo

$$\mathbf{R}_{HB}^{(i+1)} = \lambda \mathbf{R}_{HB}^{(i)}, \qquad i = 0, 1, 2 \ldots \quad 0 < \lambda < 1. \tag{3.19}$$

Z Newton-Raphsonovim postopkom (3.18) dosežemo, da bodo tokovi $\mathbf{I}$, ki tečejo iz dodanih neodvisnih virov v vozlišča analiziranega vezja, enaki $\mathbf{0}$ oz. dovolj majhni v okviru izbrane natančnosti izračuna δ, $||\mathbf{I}|| < \delta$. Hkrati z manjšanjem upornosti $\mathbf{R}_{HB}$ (enačba (3.19)) poskrbimo, da bodo napetosti v vozliščih vezja $\mathbf{V}$ enake napetostim neodvisnih generatorjev $\mathbf{V}_{HB}$:



Slika 3.5. Priključitev napetostnih virov za iskanje stacionarnega stanja vezja po modificirani metodi harmoničnega ravnovesja

$$\begin{array}{rcl} V_i = V_{HB_i} - R_{HB} \cdot I_{HB}, & & i = 1, 2 \ldots n \\ R_{HB_i} \to 0 \quad \&\& \quad I_{HB_i} \to 0 \quad \Rightarrow \quad V_i \to V_{HB_i} & & i = 1, 2 \ldots n. \end{array} \tag{3.20}$$

Opisani postopek modificirane metode harmoničnega ravnovesja je opisan tudi v [45], kjer je prikazana uporaba te metode na enostavnem primeru vezja (usmerniško vezje). Podana je primerjava odzivov, dobljenih z uporabo običajne tranzientne analize in z metodo harmoničnega ravnvesja.

3.2 Pospeševanje izračuna stacionarnega stanja z ekstrapolacijskimi metodami

Drug način izračuna stacionarnega stanja je uporaba ekstrapolacijskih metod [46]–[61]. Osnova metod je t.i. strelska metoda (angl. shooting method). Cilj metode je poiskati začetno stanje vezja (napetosti vozlišč, tokovi vej), ki bo enako stanju vezja po preteku analize ene periode vezja. Metoda je primerna tudi za zelo nelinearna vezja, manj primerna je pa za vezja, ki vsebujejo signale iz zelo širokega frekvenčnega področja [62, 63].

Metodo lahko v kratkem povzamemo v treh korakih:

1. analiza vezja v določenem časovnem intervalu,
2. vzorčenje časovnega odziva vezja oz. izbor točk, ki jih uporabimo v ekstrapolacijskem algoritmu,
3. določitev novega začetnega stanja z uporabo izbranega ekstrapolacijskega algoritma.

Postopek ponavljamo, dokler stanje vezja na začetku in na koncu osnovne periode signalov v vezju ni enako oz. ni v okviru zahtevanih toleranc.

3.2.1 Analiza vezja

V prvem koraku postopka za izračun stacionarnega stanja s pomočjo ekstrapolacijskih metod moramo vezje najprej analizirati z običajno tranzientno analizo preko časovnega intervala, ki je odvisen od zahtevnosti vezja, ki ga analiziramo, izbranega ekstrapolacijskega algoritma in natančnosti, ki jo zahtevamo pri končnem rezultatu, t.j. pri stacionarnem stanju.

Pri osnovni različici izračuna stacionarnega stanja, ki je primerna za vzbujana vezja, poznamo osnovno periodo T vzbujevalnega signala. Izvesti moramo tranzientno analizo dolžine $mT + t_{del}$, pri čemer odziva do začetnega časa t_{del} pri nadaljni obravnavi ne potrebujemo in ga zato ne vključimo v rezultate analize:

$$tran \quad t_{\Delta} \quad mT + t_{del} \quad t_{del} \ .$$

Rezultat zgornje tranzientne analize je vektor

$$\mathbf{x}(t) = \left[\, x_1(t) \quad x_2(t) \quad \cdots \quad x_n(t) \,\right]^T, \qquad t \in [t_{del}, t_{del} + mT]\,. \tag{3.21}$$

Pri tem je potrebno poudariti, da lahko točke oz. časi t, izbrani iz intervala $[t_{del}, t_{del} + mT]$, zavzemajo le diskretne vrednosti, saj je rezultat tranzientne analize (3.21) izračunan numerično.

Da dosežemo zadovoljivo natančnost pri izračunu stacionarnega stanja z uporabo ekstrapolacijskih metod, moramo interval ene periode razdeliti na vsaj nekaj sto (priporočljivo vsaj petsto) do tisoč točk in v teh točkah izračunati stanje vezja. Zato moramo pri tranzientni analizi vezja podati še četrti parameter t_{max}, ki podaja zgornjo mejo računskega koraka. Vrednost parametra t_{max} je lahko naprimer

$$t_{max} < \frac{T}{500}. \tag{3.22}$$

Ukaz za zagon tranzientne analize je tako

$$tran \quad t_\Delta \quad mT + t_{del} \quad t_{del} \quad t_{max} \ .$$

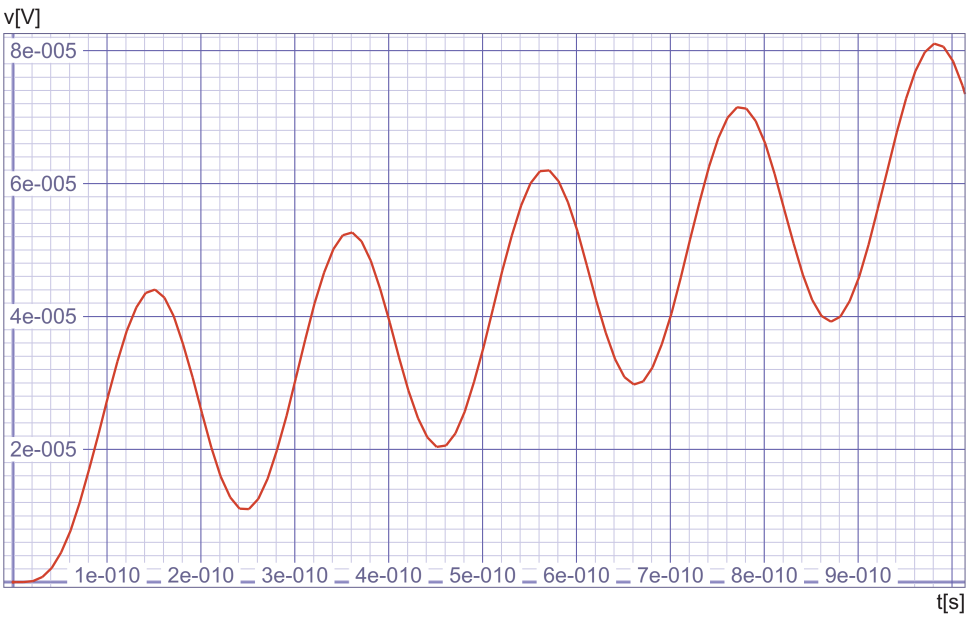
Celoten postopek izračuna stacionarnega stanja z uporabo ekstrapolacijskih metod temelji na postavitvi takih začetnih pogojev vezja, ki bodo dali po preteku analize ene periode isto stanje kot je bilo postavljeno na začetku. Zato je potrebno pri tranzientni analizi simulatorju vezij povedati, da upošteva nastavljeno začetno stanje, in sicer z besedo uic (Use Initial Conditions):

$$tran \quad t_\Delta \quad mT + t_{del} \quad t_{del} \quad t_{max} \quad uic \ .$$

Zgornja sintaksa uporabe tranzientne analize se uporablja pri vseh ekstrapolacijskih algoritmih, ki so obravnavani v tej doktorski disertaciji.

3.2.2 Vzorčenje odziva

Za potrebe ekstrapolacijskega algoritma moramo iz rezultata tranzientne analize (3.21) izbrati vrednosti vektorja $\mathbf{x}(\mathbf{t})$ pri določenih časih t. Tipični odziv vezja (napetost v enem od vozlišč) prikazuje Slika 3.6.



Slika 3.6. Napetost v enem od vozlišč kot tipični odziv vezja pri iskanju stacionarnega stanja

Signal na Sliki 3.6 ima osnovno frekvenco 4,8 GHz oz. osnovno periodo T=2,083 $\cdot 10^{-10}s$. Izbrati je potrebno točke, ki so medsebojno oddaljene za eno osnovno periodo, za zgornji primer je to 2,083 $\cdot 10^{-10}s$. Prvo točko izberemo ob času t_{del}.

Označimo izbrane točke z $\mathbf{x}_0^{(i)}, i = 0, 1, 2 \ldots m$:

$$\mathbf{x}_0^{(i)} = \begin{bmatrix} x_1(t_{del} + iT) \\ x_2(t_{del} + iT) \\ \cdots \\ x_n(t_{del} + iT) \end{bmatrix} \quad , \quad i = 0, 1, 2 \ldots m \tag{3.23}$$

Iz vektorjev (3.23) sestavimo matriko

$$\mathbf{X}_0 = \begin{bmatrix} x_1(t_{del}) & x_1(t_{del} + T) & x_1(t_{del} + 2T) & \cdots & x_1(t_{del} + mT) \\ x_2(t_{del}) & x_2(t_{del} + T) & x_2(t_{del} + 2T) & \cdots & x_2(t_{del} + mT) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_n(t_{del}) & x_n(t_{del} + T) & x_n(t_{del} + 2T) & \cdots & x_n(t_{del} + mT) \end{bmatrix}, \tag{3.24}$$

ki predstavlja vzorčene vrednosti signalov v vezju pri določenih časovnih točkah. Indeksa 0 v (3.23) in (3.24) določata, da gre za začetno oz. prvo analizo vezja, brez podanih začetnih pogojev.

V Poglavju 3.2.3 bomo s pomočjo matrike (3.24) izračunali novo začetno stanje vezja, na podlagi katerega bomo kasneje ponovno analizirali vezje.

3.2.3 Uporaba ekstrapolacijskih metod

Ekstrapolacijske metode se v splošnem uporabljajo za pospeševanje konvergence zaporedij, lahko pa jih uporabimo tudi pri analizi vezij oz. pri določevanju stacionarnega stanja vezij [47, 48]. Zaporedje, ki ga želimo ekstrapolirati, je generirano na podlagi začetnega vektorja $\mathbf{x}_0^{(0)}$ (največkrat ima vrednost **0**) in z uporabo t.i. generatorja zaporedja

$$\mathbf{x}_0^{(i+1)} = \mathbf{F}(\mathbf{x}_0^{(i)}), \quad i = 0, 1, 2 \ldots m_0 - 1. \tag{3.25}$$

Generator zaporedja **F** je lahko podan na različne načine, analitično ali pa numerično, kot je to v primeru analize električnih vezij (enačba (3.23)). Na ta način dobljeno zaporedje označimo z

$$\left\{ \mathbf{x}_0^{(i)} \right\} \ , \ \ i = 0, 1, 2 \ldots m_0 \ . \tag{3.26}$$

Če je zaporedje (3.26) konvergentno, označimo njegovo limito z

$$\mathbf{x} = \lim_{m_0 \to \infty} \mathbf{x}_0^{(m_0)} \ . \tag{3.27}$$

Z ekstrapolacijskim algoritmom **E** iz zaporedja (3.26) izračunamo nov začetni vektor $\mathbf{x}_1^{(0)}$:

$$\mathbf{x}_1^{(0)} = \mathbf{E}(\mathbf{x}_0^{(0)}, \mathbf{x}_0^{(1)}, \mathbf{x}_0^{(2)} \ldots \mathbf{x}_0^{(m_0)}) \ . \tag{3.28}$$

Začetni vektor $\mathbf{x}_1^{(0)}$ je izhodišče za izračun novega zaporedja, katerega členi so izračunani z uporabo istega generatorja zaporedja kot v (3.25):

$$\mathbf{x}_1^{(i+1)} = \mathbf{F}(\mathbf{x}_1^{(i)}), \quad i = 0, 1, 2 \ldots m_1 - 1. \tag{3.29}$$

Postopek, podan z (3.28) in (3.29), ponavljamo za $k = 1, 2, 3 \ldots$

$$\mathbf{x}_{k+1}^{(0)} = \mathbf{E}(\mathbf{x}_k^{(0)}, \mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)} \ldots \mathbf{x}_k^{(m_k)}), \tag{3.30}$$

kar nam generira novo zaporedje

$$\left\{\mathbf{x}_k^{(0)}\right\}, \quad k = 1, 2, 3 \ \ldots \tag{3.31}$$

Ekstrapolacijski algoritmi predpostavljajo lastnosti zaporedja, ki ga ekstrapolirajo. Zato je hitrost konvegence novega zaporedja (3.31) odvisna od izbire ekstrapolacijskega algoritma. Če izberemo zaporedju primeren ekstrapolacijski algoritem, lahko trdimo, da novo zaporedje konvergira k isti limiti $\mathbf{x}$ kot osnovno zaporedje (3.25) [48, 54, 55, 58, 61]:

$$\mathbf{x} = \lim_{k \to \infty} \mathbf{x}_k^{(0)} \ . \tag{3.32}$$

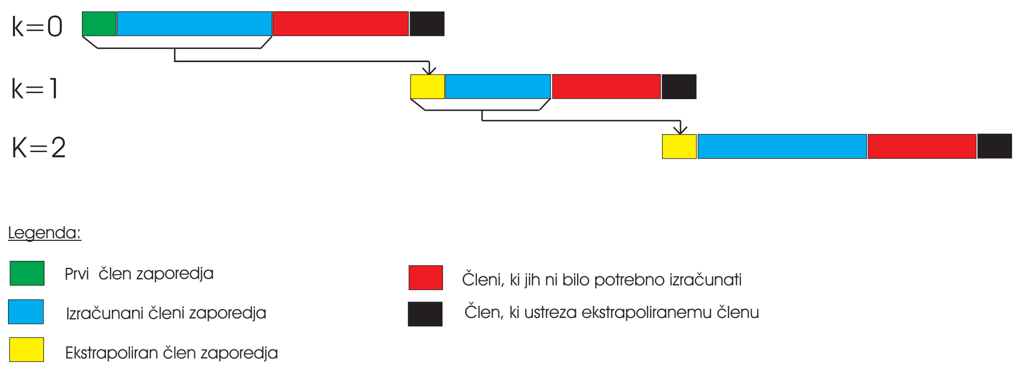
Celoten ekstrapolacijski algoritem je prikazan v Algoritmu 3.2.

1: Nastavi začetni vektor $\mathbf{x}_0^{(0)} = \mathbf{0}$
2: Nastavi $k = 0$
3: Izberi m_k
4: Izračunaj člene $\mathbf{x}_k^{(i)}, i = 0, 1, 2 \ldots m_k$
5: Izračunaj $\mathbf{x}_{k+1}^{(0)} = \mathbf{E}(\mathbf{x}_k^{(0)}, \mathbf{x}_k^{(1)} \ldots \mathbf{x}_k^{(m_k)})$
6: Povečaj k za 1
7: Pojdi na korak 3, če limita (3.32) še ni dosežena.

Algoritem 3.2. Ekstrapolacijski algoritem

Z ekstrapolacijskim postopkom, opisanim z enačbami (3.25), (3.28) in (3.29) (oz. z Algoritmom 3.2), smo hitreje določili limito zaporedja $\mathbf{x}$, kot če bi limito računali direktno, po enačbi (3.27). V vsaki iteraciji ekstrapolacijskega postopka (koraki 3, 4, 5 in 6 Algoritma 3.2) se izognemo izračunu nekaj členov v zaporedju (3.29).

Slika 3.7 prikazuje tri iteracije ekstrapolacijskega algoritma, pri čemer je prikazan princip pospeševanja izračunavanja členov zaporedja.



Slika 3.7. Prikaz treh iteracij ekstrapolacijskega algoritma

Najprej si izberemo vrednost vektorja za prvi člen (zelena barva) zaporedja prve iteracije ekstrapolacijskega algoritma ($k = 0$). Nato z uporabo enačbe (3.25) določimo $m_0 - 1$ naslednjih členov zaporedja (modra barva pri $k = 0$). Z ekstrapolacijskim algoritmom (3.28) izračunamo začetni vektor $x_1^{(0)}$ za naslednjo iteracijo ekstrapolacijskega algoritma (rumena barva pri $k = 1$). Novi začetni vektor v drugem koraku ustreza členu, označenem s črno barvo pri $k = 0$. V prvem koraku smo se tako izognili izračunu členov, označenih z rdečo barvo pri $k = 0$. Opisani postopek ponavljamo ($k = 2$, $k = 3 \ldots$), pri čemer je lahko število členov, ki jih pri vsakem k izračunamo z enačbo (3.25), različno ($m_k - 1$), kar je na Sliki 3.7 označeno z različno dolžino modrega področja.

V naslednjih razdelkih bo predstavljenih nekaj ekstrapolacijskih algoritmov, epsilon, theta, rho in topološki algoritem epsilon. Algoritmi se razlikujejo v definiciji funkcije **E** (enačba (3.28)).

Algoritem epsilon

Ekstrapolacijski algoritem (3.28) je pri algoritmu epsilon [47, 48, 52] definiran z rekurzivnim izračunom elementov v dvodimenzionalnem polju na Sliki 3.8.

Elemente s spodnjim indeksom -1 ($\epsilon_{-1}^{(1)}$, $\epsilon_{-1}^{(2)} \ldots \epsilon_{-1}^{(m_k)}$) postavimo na vrednost $\mathbf{0}$, elementom s spodnjim indeksom 0 pa priredimo vrednosti vhodnih podatkov v (3.30):

$$\epsilon_0^{(j)} = \mathbf{x}_k^{(j)}, \quad j = 0, 1, 2 \ldots m_k \,. \tag{3.33}$$

Ostale elemente na Sliki 3.8 izračunamo na naslednji način. Za vsak stoplec ($i = 0, 1, 2 \ldots m_k - 1$) izračunamo njegove elemente z enačbami

$$\epsilon_{i+1}^{(j)} = \epsilon_{i-1}^{(j+1)} + (\epsilon_i^{(j-1)} - \epsilon_i^{(j)})^{-1}, \quad j = 0, 1, 2 \ldots m_k - 1 - i. \tag{3.34}$$

$$
\begin{array}{cccccc}
 & \epsilon_0^{(0)} & & & & \\
\epsilon_{-1}^{(1)} & & \epsilon_1^{(0)} & & & \\
 & \epsilon_0^{(1)} & & \epsilon_2^{(0)} & & \\
\epsilon_{-1}^{(2)} & & \epsilon_1^{(1)} & & \ddots & \\
 & \epsilon_0^{(2)} & & \epsilon_2^{(1)} & \cdots & \epsilon_{m_k}^{(0)} \\
\epsilon_{-1}^{(3)} & & \epsilon_1^{(2)} & & \cdot^{\cdot^{\cdot}} & \\
\vdots & \vdots & \vdots & \vdots & & \\
 & \epsilon_0^{(m_k-1)} & & \epsilon_2^{(m_k-2)} & & \\
\epsilon_{-1}^{(m_k)} & & \epsilon_1^{(m_k-1)} & & & \\
 & \epsilon_0^{(m_k)} & & & &
\end{array}
$$

Slika 3.8. Dvodimenzionalno polje za izračun algoritma epsilon

V enačbi (3.34) nastopa inverzna vrednost vektorja $\epsilon_i^{(j-1)} - \epsilon_i^{(j)}$, ki je definirana kot

$$\mathbf{x}^{-1} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2^2}. \tag{3.35}$$

Elementi s sodim spodnjim indeksom ($\epsilon_{2n}^{(0)}, \epsilon_{2n}^{(1)}, \epsilon_{2n}^{(2)} \dots \ , n \in \mathbb{N}^+$) predstavljajo ekstrapolirano originalno zaporedje $\epsilon_0^{(0)}, \epsilon_0^{(1)}, \epsilon_0^{(2)} \dots$, elementi z lihim spodnjim indeksom ($\epsilon_{2n-1}^{(0)}, \epsilon_{2n-1}^{(1)}, \epsilon_{2n-1}^{(2)} \dots \ , n \in \mathbb{N}^+$) pa vmesno ekstrapolirano zaporedje, ki je inverzno glede na originalno zaporedje. Zaradi tega je končna vrednost ekstrapolacijskega algoritma (3.30) definirana z

$$\mathbf{x}_{k+1}^{(0)} = \epsilon_{m_k}^{(0)} , \tag{3.36}$$

pri čemer mora biti vrednost m_k soda ($m_k = 2 \cdot n, \quad n \in \mathbb{N}^+$).

Algoritem rho

Podobno kot pri algoritmu epsilon tudi pri algoritmu rho [53] zgradimo dvodimenzionalno polje, ki je prikazano na Sliki 3.9.

Izhodiščni elementi so določeni z

$$
\begin{aligned}
\rho_{-1}^{(j)} &= \mathbf{0}, \quad j = 1, 2, 3 \dots m_k \\
\rho_0^{(j)} &= \mathbf{x}_k^{(j)}, \quad j = 0, 1, 2 \dots m_k \ .
\end{aligned} \tag{3.37}
$$

Ostale elemente na Sliki 3.9 izračunamo tako, da posebej izračunamo elemente vsakega stolpca ($i = 0, 1, 2 \dots m_k - 1$) s formulami

$$
\begin{array}{cccccc}
 & \rho_0^{(0)} & & & & \\
\rho_{-1}^{(1)} & & \rho_1^{(0)} & & & \\
 & \rho_0^{(1)} & & \rho_2^{(0)} & & \\
\rho_{-1}^{(2)} & & \rho_1^{(1)} & & \ddots & \\
 & \rho_0^{(2)} & & \rho_2^{(1)} & \cdots & \rho_{m_k}^{(0)} \\
\rho_{-1}^{(3)} & & \rho_1^{(2)} & & \cdot^{\cdot^{\cdot}} & \\
\vdots & \vdots & \vdots & \vdots & & \\
 & \rho_0^{(m_k-1)} & & \rho_2^{(m_k-2)} & & \\
\rho_{-1}^{(m_k)} & & \rho_1^{(m_k-1)} & & & \\
 & \rho_0^{(m_k)} & & & &
\end{array}
$$

Slika 3.9. Dvodimenzionalno polje za izračun algoritma rho

$$\rho_{i+1}^{(j)} = \rho_{i-1}^{(j+1)} + (j+1)\left[\rho_i^{(j-1)} - \rho_i^{(j)}\right]^{-1}, \quad j = 0, 1, 2 \ldots m_k - 1 - i. \tag{3.38}$$

Inverzno vrednost vektorja $\rho_i^{(j-1)} - \rho_i^{(j)}$ izračunamo enako kot pri algoritmu epsilon (enačba (3.35)). Rezultat ekstrapolacijskega algoritma je za sode vrednosti m_k definiran z

$$\mathbf{x}_{k+1}^{(0)} = \rho_{m_k}^{(0)} \, . \tag{3.39}$$

Algoritem theta

Pri algoritmu theta [53] se izračun elementov na Sliki 3.10 razlikuje za lihe in za sode stolpce oz. spodnje indekse.

Izhajamo iz vrednosti prvih dveh stolpcev, ki jih določimo z izrazi

$$\begin{aligned} \vartheta_{-1}^{(j)} &= \mathbf{0}, \quad j = 1, 2, 3 \ldots m_k \\ \vartheta_0^{(j)} &= \mathbf{x}_k^{(j)}, \quad j = 0, 1, 2 \ldots m_k. \end{aligned} \tag{3.40}$$

Ostale vrednosti izračunamo za vsak $i = 0, 1, 2 \ldots \frac{m_k}{2} - 1$ s formulami

$$\vartheta_{2i+1}^{(j)} = \vartheta_{2i-1}^{(j+1)} + \left[\Delta\vartheta_{2i}^{(j)}\right]^{-1}, \quad j = 0, 1, 2 \ldots m_k - 1 - 2i \tag{3.41}$$

za lihe spodnje indekse elementov in

$$\vartheta_{2i+2}^{(j)} = \vartheta_{2i}^{(j+1)} + (\Delta\vartheta_{2i}^{(j+1)}, \Delta\vartheta_{2i+1}^{(j+1)}) \cdot \left[\Delta^2\vartheta_{2i+1}^{(j+1)}\right]^{-1}, \quad j = 0, 1, 2 \ldots m_k - 2 - 2i \tag{3.42}$$

$$
\begin{array}{cccccc}
 & \vartheta_0^{(0)} & & & & \\
\vartheta_{-1}^{(1)} & & \vartheta_1^{(0)} & & & \\
 & \vartheta_0^{(1)} & & \vartheta_2^{(0)} & & \\
\vartheta_{-1}^{(2)} & & \vartheta_1^{(1)} & & \ddots & \\
 & \vartheta_0^{(2)} & & \vartheta_2^{(1)} & \cdots & \rho_{m_k}^{(0)} \\
\vartheta_{-1}^{(3)} & & \vartheta_1^{(2)} & & \cdot^{\cdot^{\cdot}} & \\
\vdots & \vdots & \vdots & \vdots & & \\
 & \vartheta_0^{(m_k-1)} & & \vartheta_2^{(m_k-2)} & & \\
\vartheta_{-1}^{(m_k)} & & \vartheta_1^{(m_k-1)} & & & \\
 & \vartheta_0^{(m_k)} & & & &
\end{array}
$$

Slika 3.10. Dvodimenzionalno polje za izračun algoritma theta

za sode spodnje indekse elementov na Sliki 3.10. V (3.41) in (3.42) so uporabljene naslednje oznake za operator Δ :

$$
\begin{aligned}
\Delta\vartheta_i^{(j)} &= \vartheta_i^{(j+1)} - \vartheta_i^{(j)}\,, \qquad \forall\, i,\ \forall\, j \\
\Delta^2\vartheta_i^{(j)} &= \Delta\vartheta_i^{(j+1)} - \Delta\vartheta_i^{(j)}\,, \qquad \forall\, i,\ \forall\, j \quad .
\end{aligned}
\tag{3.43}
$$

V enačbi (3.42) je uporabljen tudi skalarni produkt $(\,\cdot\,,\,\cdot\,)$ med vektorjema $\Delta\vartheta_{2i}^{(j+1)}$ in $\Delta\vartheta_{2i+1}^{(j+1)}$.

Rezultat algoritma theta je definiran za sode m_k z izrazom

$$\mathbf{x}_{k+1}^{(0)} = \vartheta_{m_k}^{(0)}. \tag{3.44}$$

Topološki algoritem epsilon

Topološki algoritem epsilon [46] poteka na podoben način kot algoritem epsilon, pri čemer upoštevamo drugačno definicijo inverzne vrednosti vektorja v formulah algoritma epsilon (3.34). Inverzna vrednost vektorja (3.35) je tako definirana z

$$\mathbf{x}^{-1} = \frac{\mathbf{y}}{(\mathbf{y},\mathbf{x})}, \tag{3.45}$$

kar imenujemo *inverz vektorja* $\mathbf{x}$ *glede na vektor* $\mathbf{y}$. Pri topološkem algoritmu epsilon izračunamo elemente v (3.34) s sodim in lihim spodnjim indeksom glede na različen vektor $\mathbf{y}$ v (3.45). Topološki algoritem epsilon je tako definiran s postavitvijo začetnih vrednosti elementov

$$
\begin{aligned}
\epsilon_{-1}^{(j)} &= \mathbf{0}, \quad j = 1,2,3\ldots m_k \\
\epsilon_0^{(j)} &= \mathbf{x}_k^{(j)}, \quad j = 0,1,2\ldots m_k
\end{aligned}
\tag{3.46}
$$

in za vsak $i = 0,1,2\ldots \frac{m_k}{2} - 1$ z enačbami

$$\epsilon_{2i+1}^{(j)} = \epsilon_{2i-1}^{(j+1)} + \frac{\mathbf{y}}{(\mathbf{y},\Delta\epsilon_{2i}^{(j)})}, \quad j = 0,1,2\ldots m_k - 1 - 2i \tag{3.47}$$

$$\epsilon_{2i+2}^{(j)} = \epsilon_{2i}^{(j+1)} + \frac{\Delta\epsilon_{2i}^{(j)}}{(\Delta\epsilon_{2i}^{(j)}, \Delta\epsilon_{2i+1}^{(j)})}, \quad j = 0, 1, 2 \ldots m_k - 2 - 2i. \tag{3.48}$$

Pri elementih z lihim spodnjim indeksom (3.47) je inverz vektorja izračunan glede na vektor $\mathbf{y}$, pri elementih s sodim spodnjim indeksom (3.48) pa glede na vektor $\Delta\epsilon_{2i}^{(j)}$. Izbira vektorja $\mathbf{y}$ je poljubna, vendar mora biti taka, da za vse $j = 0, 1, 2 \ldots m_k - 1 - 2i$ obstaja $\epsilon_{2i+1}^{(j)}$.

Končna vrednost ekstrapolacijskega algoritma z uporabo topološkega algoritma epsilon je podana za sode vrednosti m_k z izrazom

$$\mathbf{x}_{k+1}^{(0)} = \epsilon_{m_k}^{(0)}. \tag{3.49}$$

3.2.4 Iteracijski postopek

Za izračun stacionarnega stanja električnega vezja uporabimo eno od zgoraj predstavljenih ekstrapolacijskih metod. Rezultat ekstrapolacijske metode uporabimo za nastavitev začetnega stanja vezja za naslednjo iteracijo analize vezja. Nato preverimo, če je vezje že doseglo stacionarno stanje, oz. če se je le-temu približalo dovolj, t.j. znotraj izbranih meja. Če vezje še ni doseglo stacionarnega stanja, opisani postopek ponavljamo, dokler vezje ne doseže stacionarnega stanja ali dokler ne izvedemo izbranega maksimalnega števila iteracij [47].

Za preverjanje, ali je vezje doseglo stacionarno stanje, si izberemo določeno odstopanje od stacionarnega stanja, ki ga ima lahko odziv vezja. V ta namen določimo števili $\delta_a > 0$ in $\delta_r > 0$, ki predstavljata stopnjo absolutnega in relativnega odstopanja od stacionarnega stanja odziva vezja. V odzivu vezja (3.21) moramo preveriti, če za vse $0 < t < T$ velja enačba

$$\mathbf{x}(t) = \mathbf{x}(t + T). \tag{3.50}$$

Rezultat analize vezja (3.21) je pridobljen numerično z računalniškim programom za analizo vezij, zato zaradi numeričnih napak enačba (3.50) ne more biti natančno izpolnjena. Za ustavitev iteracijskega postopka oz. za preverjanje, če je odziv vezja v stacionarnem stanju, uporabimo naslednjo neenačbo

$$|x_i(t) - x_i(t + T)| \le \delta_a + \max\left[|x_i(t)|, |x_i(t + T)|\right]\delta_r, \tag{3.51}$$

ki mora veljati za vse točke znotraj ene periode (za vse $0 < t < T$) in za vse komponente vektorja $\mathbf{x}$, t.j. za $i = 1, 2, 3 \ldots n$. Če je pogoj v zgornji neenačbi izpolnjen, trdimo, da je vezje *doseglo stacionarno stanje v okviru izbrane absolutne in relativne natančnosti δ_a in δ_r*. V praksi je zadovoljiva izbira vrednosti δ_a in δ_r od 10^{-5} do 10^{-3}, odvisno od natančnosti, ki jo želimo pri izračunu stacionarnega stanja in analize vezja nasploh. Teh dveh vrednosti ne smemo poljubno zmanjšati, saj lahko dosežemo relativno natančnost števil, ki je uporabljena pri izračunavanju odziva vezja v programu za analizo vezij. V primeru orodja SPICE OPUS so to števila z dvojno natančnostjo (standard za števila s plavajočo vejico IEEE 754, 64 bit double precision), ki imajo relativno natančnost približno 10^{-14} ter absolutno natančnost približno 10^{-320}. Vrednosti δ_a in δ_r ni smiselno zmanjševati tudi zaradi modeliranja elementov vezja, ki ga analiziramo. Elementi vezja so namreč opisani s posameznimi modeli. Glede na realne elemente so ti modeli opisani z določeno natančnostjo. Če bi vrednosti parametrov δ_a in δ_r zmanjšali pod mejo natančnosti modelov elementov, bi sicer dobili stacionarno stanje, ki pa zaradi omejene natančnosti modelov ne bi bilo nič boljše (glede na realne elemente), kot če bi izbrali večje vrednosti parametrov δ_a in δ_r.

Iteracijski postopek lahko ponazorimo z Algoritmom 3.3.

1: Izbira absolutne in relativne natančnosti δ_a in δ_r.
2: Izbira maksimalnega števila iteracij k_{max} .
3: Izbira ekstrapolacijskega algoritma (epsilon, rho, theta, topološki algoritem epsilon).
4: Nastavitev začetnega stanja vezja na $\mathbf{0}$.
5: Nastavitev $k = 0$.
6: Izbira števila period m_k.
7: Analiza vezja za čas števila period m_k.
8: Izračun ekstrapolacijskega algoritma (enačba (3.30)).
9: Začetno stanje vezja $\leftarrow$ rezultat ekstrapolacijskega algoritma.
10: Preveri pogoj (3.51).
  Pogoj izpolnjen $\Rightarrow$ stacionarno stanje je doseženo. Nadaljuj na 14.
11: Preveri pogoj $k > k_{max}$.
  Pogoj izpolnjen $\Rightarrow$ stacionarno stanje ni doseženo. Nadaljuj na 14.
12: Povečaj k.
13: Nadaljuj na 6.
14: Konec.

Algoritem 3.3. Iteracijski postopek za izračun stacionarnega stanja električnega vezja z ekstrapolacijskimi metodami

3.2.5 Vrstni red izračuna členov ekstrapolacijske sheme

Pri velikih vezjih, če je dimenzija vektorjev $\mathbf{x}_k^{(j)}, j = 0, 1, 2 \ldots m_k$ v ekstrapolacijskem algoritmu (3.30) velika, lahko z določenim vrstnim redom izračunavanja členov v ekstrapolacijskih shemah (algoritem epsilon, rho, theta, topološki algoritem epsilon) prihranimo precej pomnilniškega prostora [47].

Ponazorimo vrstni red izračuna členov pri algoritmu epsilon, pri čemer si izberimo vrednost parametra $m_k = 4$. Za ostale opisane ekstrapolacijske algoritme je postopek določitve vrstnega reda izračuna členov podoben.

Za izbrano vrednost parametra m_k sestavimo dvodimenzionalno shemo, ki je prikazana na Sliki 3.11 (primerjaj s splošno shemo na Sliki 3.8).

$$
\begin{array}{cccccc}
 & \epsilon_0^{(0)} & & & & \\
\epsilon_{-1}^{(1)} & & \epsilon_1^{(0)} & & & \\
 & \epsilon_0^{(1)} & & \epsilon_2^{(0)} & & \\
\epsilon_{-1}^{(2)} & & \epsilon_1^{(1)} & & \epsilon_3^{(0)} & \\
 & \epsilon_0^{(2)} & & \epsilon_2^{(1)} & & \epsilon_4^{(0)} \\
\epsilon_{-1}^{(3)} & & \epsilon_1^{(2)} & & \epsilon_3^{(1)} & \\
 & \epsilon_0^{(3)} & & \epsilon_2^{(2)} & & \\
\epsilon_{-1}^{(4)} & & \epsilon_1^{(3)} & & & \\
 & \epsilon_0^{(4)} & & & &
\end{array}
$$

Slika 3.11. Dvodimenzionalno polje za izračun algoritma epsilon ($m_k = 4$)

Za izračun končnega rezultata algoritma epsilon ($\epsilon_4^{(0)}$) ni potrebno imeti shranjenih vseh vrednosti elementov na Sliki 3.11, ampak lahko člene izračunavamo postopno, z vsakim novim členom $\epsilon_0^{(j)}, j = 0, 1, 2, 3, 4$, ki vstopi v algoritem epsilon. Člene, ki jih po izračunu ne potrebujemo več, lahko odstranimo in pri tem sprostimo nekaj pomnilnika.

Združimo elemente $\epsilon_i^{(j)}$ iz Slike 3.11, za katere velja $i + j = N$ v *verigo N*. Na ta način dobimo 5 verig, *verigo 0*, *verigo 1*, *verigo 2*, *verigo 3* in *verigo 4*. Elementi *verige 0* so že znani ($\epsilon_{-1}^{(1)} = \mathbf{0}$, $\epsilon_0^{(0)} = \mathbf{x}_k^{(0)}$). V *verigi 1* je potrebno z uporabo (3.34) izračunati samo element $\epsilon_1^{(0)}$, ostala dva elementa sta vhodna podatka ($\epsilon_{-1}^{(2)} = \mathbf{0}$, $\epsilon_0^{(1)} = \mathbf{x}_k^{(1)}$). Po izračunu elementa v *verigi 1* elementov v *verigi 0* ne potrebujemo več in zato lahko sprostimo pomnilnik, ki so ga ti elementi zasedali.

Podoben postopek ponovimo še za *verigo 2, verigo 3* in *verigo 4.* Pomemben je tudi vrstni red izračuna elementov znotraj posamezne verige, saj so elementi z večjim spodnjim indeksom izračunani iz elementov z manjšim spodnjim indeksom. Splošno moramo elemente v *verigi N* izračunati po naslednjem vrstnem redu: $\epsilon_1^{(N-1)}$, $\epsilon_2^{(N-2)}$, $\epsilon_3^{(N-3)} \ldots \epsilon_N^{(0)}$.

3.3 Stacionarno stanje nevzbujanih vezij

V Poglavju 3.1 smo v (3.1) predpostavili poznavanje frekvenčnih komponent Ω signalov v vezju, ki ga analiziramo oz. ki mu določamo odziv v stacionarnem stanju. Te frekvenčne komponente smo izračunali na podlagi znanih frekvenc vzbujanja vezja oz. njihovih period. Prav tako smo v Poglavju 3.2 v (3.21) predpostavili osnovno periodo signalov T, ki so prisotni v vezju. Na podlagi znane periode smo vezje analizirali za čas, ki je enak mnogokratniku te periode.

Iz zgoraj napisanega sledi, da sta osnovni varianti metod za določanje stacionarnega stanja (metoda harmoničnega ravnovesja in ekstrapolacijske metode) primerni samo za vezja, ki so vzbujana, oz. neavtonomna vezja.

Vezjem, ki niso vzbujana (avtonomna vezja), lahko določimo odziv v stacionarnem stanju z uporabo metode harmoničnega ravnovesja in ekstrapolacijskih metod, če poznamo osnovno periodo oz. frekvenčne komponente signalov odziva vezja.

Če torej na nek način določimo frekvence signalov v odzivu vezja, lahko uporabimo postopke, ki so bili opisani v Poglavjih 3.1 in 3.2. Eden od načinov določitve frekvenčnih komponent signalov odziva vezja je uporaba Fourierjeve transformacije. Drugi način je detekcija prehoda signala preko določenega nivoja oz. preverjanje, ali signal izpolnjuje določen pogoj. Razlika časov, pri katerih je ta pogoj izpolnjen, je povezana s frekvenčnimi komponentami opazovanega signala. V naslednjih dveh podpoglavjih sta podrobneje opisana dva načina za določanje frekvenčnih komponent odziva vezja.

Pri računanju stacionarnega stanja vezij s pomočjo ekstrapolacijskih metod je pomembno, da se frekvenca signalov ne spreminja, sploh med posameznimi iteracijami ekstrapolacijskega postopka. Če je npr. frekvenca signalov v vezju odvisna od določenega signala v vezju, ima lahko ta metoda težave. Primer takega vezja je npr. fazno sklenjena zanka, ki jo tipično sestavlja napetostno krmiljeni oscilator. Frekvenca oscilatorja je odvisna od napetosti, ki je v razmerju s signalom napake. Preden tako vezje doseže stacionarno stanje, se lahko frekvenca izhodnega signala spreminja na širokem frekvenčnem pasu, kar pa za uporabo ekstrapolacijskih metod ni primerno. Ta problem bomo natančnejše opisali v Poglavju 3.4. Fazno sklenjeno zanko bi sicer lahko obravnavali kot vzbujano vezje (običajno ima na vhodu signal z točno določeno frekvenco), a dokler vezje ni v zaklenjenem stanju, se mu frekvence in faze signalov spreminjajo in ga moramo obravnavati kot nevzbujano vezje (podobno kot npr. pri oscilatorjih).

Pri analizah tako nevzbujanih kot vzbujanih vezij se izkaže, da odstopanje od končne frekvence odziva (v stacionarnem stanju) pri določanju stacionarnega stanja s postopki, ki so bili opisani v Poglavju 3.2, ne vpliva na učinkovitost oz. uspešnost postopkov. V kakšnih mejah se lahko frekvenca spreminja med postopkom izračuna stacionarnega stanja, je težko vnaprej določiti ali napovedati. Meja je zelo odvisna od samega vezja, njegovih lastnosti in tudi od izbire ekstrapolacijskega algoritma.

3.3.1 Fourierjeva transformacija

Frekvenčni spekter signala lahko določimo s Fourierjevo transformacijo [64]. Če je funkcija $x(t) \in \mathbb{C}$ integrabilna, lahko izračunamo Fourierjevo transformiranko $X(\omega)$

$$X(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} x(t) e^{-i\omega t}\, dt, \tag{3.52}$$

ki je definirana za $\omega \in \mathbb{R}$.

Funkcija $x(t)$ je podana na celotni realni osi, frekvenčni spekter oz. $X(\omega)$ je zvezen. Ker je v primeru analize vezij funkcija $x(t)$ izračunana numerično, za diskretne čase t in na omejenem časovnem intervalu, za določitev frekvenčnega spektra tega signala ne moremo uporabiti Fourierjeve transformacije, ampak diskretno Fourierjevo transformacijo.

Diskretna Fourierjeva transformacija

Množica $\{x_n,\ n = 0, 1, 2 \ldots N-1\}$ naj predstavlja podmnožico vzorčenega signala $x(t)$. V splošnem so elementi te množice kompleksna števila. Diskretna Fourierjeva transformacija je definirana kot

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, 1, 2 \ldots N-1. \tag{3.53}$$

X_k predstavlja frekvenčni spekter vzorčenega signala $x(t)$ pri normalizirani krožni frekvenci $\omega_k = \frac{2\pi k}{N}$. Če je signal $x(t)$ ekvidistantno vzorčen z vzorčno frekvenco f_s, lahko normalizirano krožno frekvenco zapišemo še kot $\omega_k = \frac{2\pi f_k}{f_s}$, iz česar sledi izraz za običajno frekvenco $f_k = \frac{k}{N} f_s$. Če to izpeljavo vstavimo v (3.53), dobimo izraz za odvisnost frekvenčnega spektra vzorčenega signala $x(t)$ od frekvence f_k:

$$X(f_k) = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{f_s} f_k n} \qquad k = 0, 1, 2 \ldots N-1. \tag{3.54}$$

Frekvenčni spekter (3.54) je diskreten. Frekvenčne komponente, ki so v tem spektru zastopane, so podmnožica frekvenc zveznega spektra (3.52) Fourierjeve transformiranke signala $x(t)$, saj je bil signal $x(t)$ vzorčen, in sicer na omejenem časovnem intervalu (oknenje). Zaradi končne računske zmogljivosti, natančnosti in s tem omejitev pri izbiri računskega koraka, signal, opisan z (3.54), ni popolna predstavitev originalnega signala $x(t)$, še manj pa signala, ki bi ga imelo realno vezje, ki smo ga analizirali s programom za analizo vezij. Zaradi tega je uspešnost oz. učinkovitost metod, opisanih v Poglavjih 3.1 in 3.2, odvisna od izbire frekvence vzorčenja f_s in števila točk N v (3.54), sploh pri vezjih, katerih frekvence signalov ne poznamo (avtonomna vezja) in jo moramo poiskati npr. z diskretno Fourierjevo transformacijo.

V praksi učinkovito in hitro izračunamo diskretno Fourierjevo transformacijo s pomočjo hitre Fourierjeve transformacije (FFT - Fast Fourier Transform). Na ta način zmanjšamo število matematičnih operacij iz velikostnega razreda N^2 na $N\ logN$. Algoritmi za izračun hitre Fourierjeve transformacije so opisani v [64].

3.3.2 Detekcija prehoda skozi nivo

Enostavnejša oblika določanja osnovne frekvence signala odziva vezja je detekcija prehoda signala skozi nivo.

Predpostavimo, da lahko signal vezja $x(t)$, ki se približuje stacionarnemu stanju in ki mu želimo določiti osnovno periodo, zapišemo v obliki

$$x(t) = \sum_{n=0}^{N} f_n(t)(A_n cos\omega_n t + B_n sin\omega_n t). \tag{3.55}$$

Množica $\Omega = \{\omega_n,\ n = 0, 1, 2 \dots N\}$ predstavlja frekvenčne komponente signala $x(t)$. Če za frekvenco ω_0 izberemo vrednost 0, predstavlja vrednost A_0 enosmerno komponento odziva vezja v stacionarnem stanju. Prva frekvenca ω_1 v množici Ω, ki je različna od 0, naj bo manjša od ostalih elementov te množice

$$\omega_1 < \omega_n, \qquad \forall n = 2, 3, 4 \dots N. \tag{3.56}$$

Funkcija $f_n(t)$ podaja, kako se spreminja amplituda frekvenčne komponente n v odvisnosti od časa. To odvisnost lahko največkrat podamo v eksponentni obliki $f_n(t) = 1 - e^{-t/\tau_n}$. Zahtevajmo še, da se vse funkcije $f_n(t)$ za dovolj velike čase t približujejo vrednosti 1

$$\lim_{t\to\infty} f_n(t) = 1, \qquad \forall n = 0, 1, 2 \dots N. \tag{3.57}$$

Pri tako podanem signalu $x(t)$ predstavljajo vrednosti A_n in B_n amplitude frekvenčnih komponent signala v stacionarnem stanju oz. po dovolj dolgem času t. Ker je bil signal $x(t)$ v enačbi (3.55) predstavljen kot vsota sinusnega in kosinusnega nihanja, ima običajna amplituda ($A_n^{'}$ v zapisu oblike $A_n^{'}\ sin(\omega_n t + \varphi_n)$) vrednost $A_n^{'} = \sqrt{A_n^2 + B_n^2}$, faza φ_n pa vrednost $\varphi_n = arctg(A_n/B_n)$.

Največkrat je signal $x(t)$ tak, da so elementi množice Ω podani v obliki $\omega_n = n\omega_1, \quad n = 2, 3, 4 \ldots N$. V tem primeru je osnovna frekvenca $\omega_1{}'$ enaka frekvenci ω_1. V splošnem je pa osnovna perioda $T^{'}$ enaka najmanjšemu skupnemu večkratniku period frekvenčnih komponent iz množice Ω.

Naloga je poiskati periodo T najnižje frekvenčne komponente ω_1 oz. periodo $T^{'}$ osnovne frekvence $\omega_1^{'}$ ob predpostavki, da so amplitude ostalih frekvenčnih komponent v množici Ω mnogo manjše od amplitude najnižje frekvenčne komponente

$$\sqrt{A_1^2 + B_1^2} \gg \sqrt{A_n^2 + B_n^2}, \qquad n = 2, 3, 4 \ldots N. \tag{3.58}$$

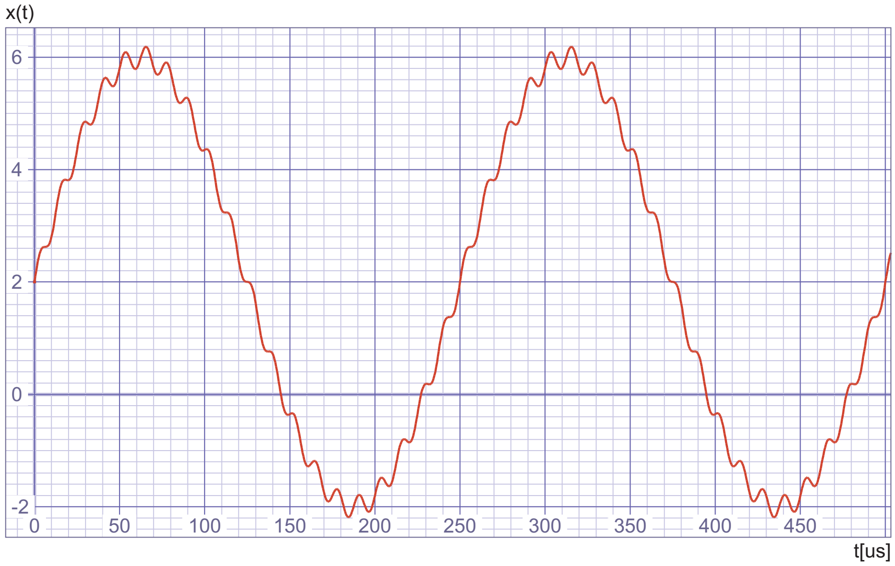
Signal $x(t)$ je odziv vezja, ki ima zelo dolge prehodne pojave, oz. vse časovne konstante τ_n v predstavitvi signala $x(t)$ (3.55) so mnogo večje od iskane osnovne periode signala $T^{'}$. Zaradi tega lahko signal $x(t)$ zapišemo kot približek signala

$$\tilde{x}(t) = \sum_{n=0}^{N} (\tilde{A}_n cos\omega_n t + \tilde{B}_n sin\omega_n t). \tag{3.59}$$

Vzemimo za primer signal $x(t)$ z naslednjimi vrednostmi parametrov: $N = 2, \quad \omega_0 = 0,$ $\omega_1 = 2\pi \cdot 4\, kHz, \quad \omega_2 = 2\pi \cdot 80\, kHz, \quad \tilde{A}_0 = 2, \quad \tilde{A}_1 = 4, \quad \tilde{A}_2 =$0,2 , $\quad \tilde{B}_0 = \tilde{B}_1 = \tilde{B}_2 = 0.$ Tak signal prikazuje Slika 3.12.

Za določitev osnovne periode s pomočjo detekcije prehoda skozi nivo si najprej izberemo določen nivo signala $x(t)$ in ga označimo z L. Vrednost nivoja mora biti med minimalno in maksimalno vrednostjo signala $x(t)$ na intervalu, na katerem želimo določiti osnovno periodo. Poleg tega moramo poznati še približno vrednost periode signala $x(t)$. Nato na intervalu, ki ima dolžino nekaj predpostavljenih osnovnih period, analiziramo vezje (tranzientna analiza). Tako bo interval signala $x(t)$ vseboval vsaj eno osnovno periodo.

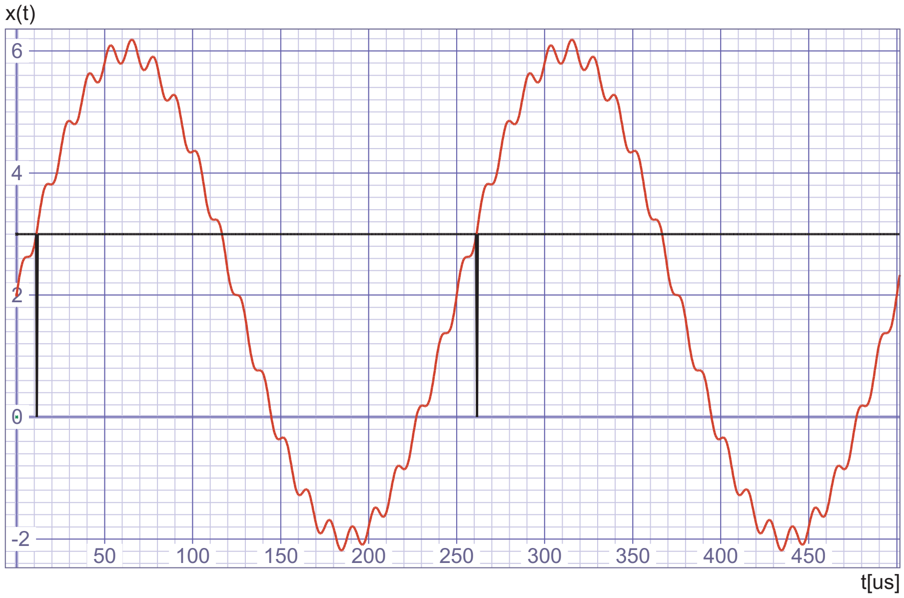
Postopek določevanja osnovne periode signala ponazorimo na Sliki 3.13. Predpostavljena je bila približna perioda signala $\tilde{T} = 100\ \mu s$. Za analizo signala smo vzeli interval petkratne dolžine približne periode, t.j. $500\ \mu s$. Signal $x(t)$ na tem intervalu vsebuje dve periodi, kar je več kot zahtevana ena perioda signala. Nato si izberimo vrednost nivoja npr. $L = 3$ in poiščimo presečišča signala $x(t)$ skozi nivo $L = 3$. Presečišča označimo z $t_{p_1}, t_{p_2}\ \ldots$ Tako velja, da je $x(t_{p_i}) = L, \quad \forall i$. Izberemo samo presečišča t_{p_i}, za katera velja, da sta za vsak t iz neke okolice točke t_{p_i} izpolnjeni neenačbi



Slika 3.12. Signal $x(t)$ v (3.59) za vrednosti $N = 2$, $\omega_0 = 0$, $\omega_1 = 2\pi \cdot 4\, kHz$, $\omega_2 = 2\pi \cdot 80\, kHz$, $\tilde{A}_0 = 2$, $\tilde{A}_1 = 4$, $\tilde{A}_2 =$0,2 , $\tilde{B}_0 = \tilde{B}_1 = \tilde{B}_2 = 0$

$$\begin{aligned} x(t) < L, \quad t < t_{p_i}, \quad \forall i, \\ x(t) > L, \quad t > t_{p_i}, \quad \forall i. \end{aligned} \tag{3.60}$$

Ta presečišča imenujmo *presečišča s pozitivnim naklonom signala* $x(t)$. Signal $x(t)$, nivo $L = 3$ in presečišča s pozitivnim naklonom t_{p_i} prikazuje Slika 3.13. Presečišči z nivojem $L = 3$ sta pri



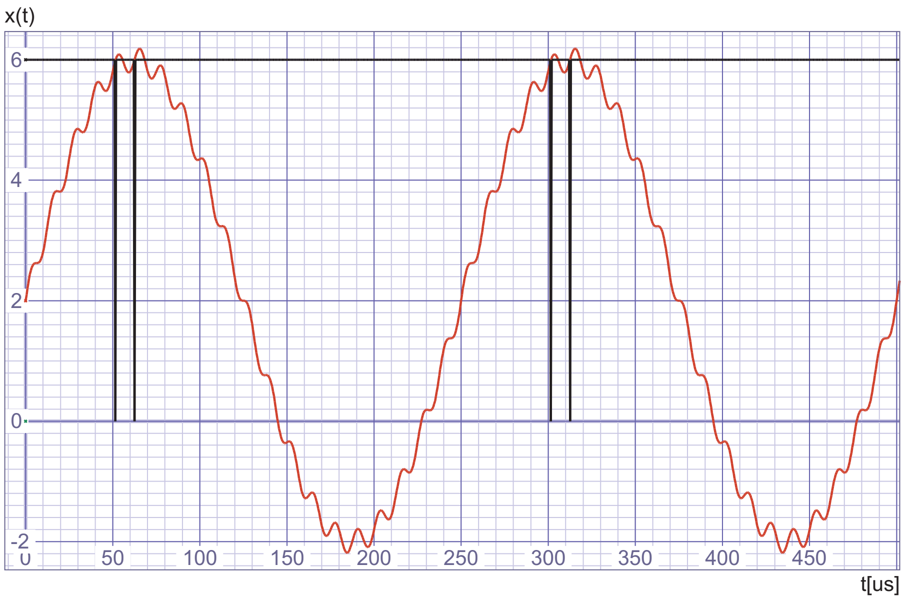
Slika 3.13. Signal $x(t)$, nivo $L = 3$ in presečišča t_{p_i}

vrednostih t_{p_1} = 11,5 μs in pri t_{p_2} = 261,5 μs. Osnovna perioda signala $x(t)$ je enaka razliki zaporednih časov, pri katerih signal $x(t)$ seka nivo L:

$$T^{'} = t_{p_i} - t_{p_{i-1}}, \quad i = 2, 3, 4 \ \ldots \tag{3.61}$$

Za primer s Slike 3.13 dobimo za $i = 2$ vrednost osnovne periode $T^{'}$ = 261,5 μs−11,5 μs = 250 μs. Osnovna frekvenca signala $x(t)$ je tako $\nu^{'} = 1/T^{'} = 1/250\ \mu s = 4\ kHz$, kar ustreza izbiri parametrov signala (3.59) na Sliki 3.12.

Dopolnimo opisano metodo tako, da bo primerna tudi za poljubno izbiro nivoja L med maksimalno in minimalno vrednostjo signala $x(t)$ na izbranem intervalu. V primeru, če za nivo L izberemo vrednost $L = 6$, bi dobili več vrednosti presečišč s tem nivojem, t_{p_1} = 51,5 μs, t_{p_2} = 62,5 μs, t_{p_3} = 301,5 μs in t_{p_4} = 312,5 μs. Ta presečišča signala $x(t)$ prikazuje Slika 3.14.



Slika 3.14. Signal $x(t)$, nivo $L = 6$ in presečišča x_{p_i}

Dopolnitev metode izvedemo na tak način, da za čas ocenjene periode $\tilde{T}$ po prvem presečišču vsa morebitna dodatna presečišča ignoriramo. Za zgornji primer je prvo presečišče t_{p_1} =51,5 μs, ocenjena perioda je $100\ \mu s$. Vsa presečišča na intervalu 51,5 $\mu s < t_{p_i} <$ 151,5 μs, t.j. presečišče t_{p_2}, zanemarimo oz. ga ne upoštevamo v izračunu osnovne periode (3.61).

Celotni postopek iskanja osnovne periode $T^{'}$ lahko povzamemo v naslednjih korakih:

1. **Iskanje presečišč t_{p_i} s pozitivnim naklonom signala $x(t)$ skozi nivo L:**
   Določi vrednosti t_{p_i}, za katere velja:

$x(t_{p_i}) = L$ in
obstaja okolica $\mathcal{O}$ točke t_{p_i}, da za vsak $t \in \mathcal{O}$ velja:
$x(t) < L, \quad t < t_{p_i}, \quad \forall i$ in
$x(t) > L, \quad t > t_{p_i}, \quad \forall i$.

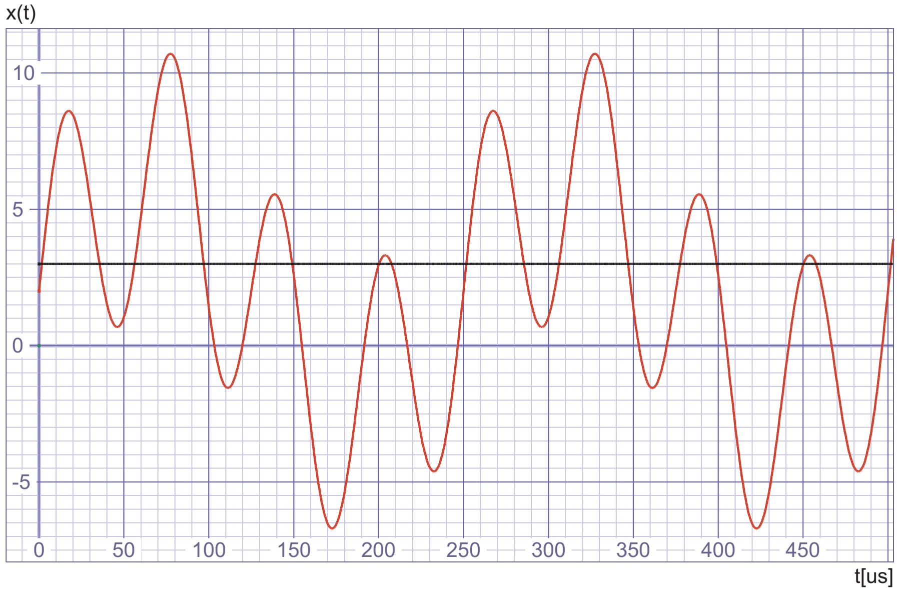
2. **Izbiranje ustreznih presečišč:**
   Izmed presečišč t_{p_i}, določenih pri 1. točki, glede na ocenjeno periodo $\tilde{T}$ izberimo tista presečišča $t_{p_{i'}}$, za katera velja:
   $t_{p_i} < t_{p_{i'}} - \tilde{T}, \quad i < i'$
   oz.
   $t_{p_{i'}} > t_{p_i} + \tilde{T}, \quad i < i'$
   kar pomeni, da morajo biti od izbranega presečišča $t_{p_{i'}}$ vsa predhodna presečišča ($t_{p_i}, i < i'$) oddaljena več kot je ocenjena perioda $\tilde{T}$.

3. **Izračun osnovne periode T':**
   Osnovna perioda T' je definirana kot razlika zaporednih presešišč $t_{p_{i'}}$, določenih v 2. točki.

Opisana metoda je uporabna, ker ne zahteva velike računske moči. Slabost metode pa je omejenost na določene tipe signalov, kjer je osnovna frekvenčna komponenta mnogo manjša v primerjavi z naslednjo frekvenčno komponento (4 kHz v primerjavi z 80 kHz na Sliki 3.12) in amplituda prve frekvenčne komponente mnogo večja od amplitud naslednjih frekvenčnih komponent (4 v primerjavi z 0,2 na Sliki 3.12). Na Sliki 3.15 je prikazan primer, kjer z opisano metodo ne moremo pravilno določiti osnovne frekvence, ampak moramo uporabiti metodo, opisano v Poglavju 3.3.1, ki pa je računsko bolj zahtevna. Z metodo detekcije prehoda skozi nivo bi tako za signal $x(t)$ in izbiro nivoja $L = 3$ dobili vrednost osnovne frekvence T' od 30 μs do 50 μs, odvisno od tega, katere točke presečišča z nivojem L bi upoštevali pri izračunu.

Z opisano metodo bi lahko določili osnovno periodo, če bi izbrali vrednost nivoja $L = 10$. Izbira nivoja igra pri tej metodi pomembno vlogo. Od nje je odvisna uspešnost in točnost izračuna osnovne periode.

Metodo lahko posplošimo tako, da spreminjamo vrednost nivoja, dokler ne dobimo zadostnega števila presešišč. Ena možnost spreminjanja nivoja je izbira neke velike vrednosti za nivo L. Za primer s Slike 3.15 lahko za nivo L izberemo vrednost 20, saj je signal $x(t)$ na opazovanem intervalu $[0, 500\ \mu s]$ povsod pod tem nivojem. Nato postopoma znižujemo vrednost nivoja,



Slika 3.15. Signal $x(t)$ in nivo $L = 3$. Za določitev osnovne frekvence moramo uporabiti metodo, opisano v Poglavju 3.3.1

dokler ne najdemo prvih presečišč signala $x(t)$ z nivojem L. Za zgornji primer se to zgodi pri vrednosti L =10,71. Pri tej vrednosti nivoja dobimo vrednosti presečičš 77,5 μs in 327,5 μs. Ker ima signal $x(t)$ pri teh dveh presečiščih maksimum, je položaj presečišč zelo odvisen od numeričnih napak pri izračunu signala $x(t)$. Zato je priporočljivo, da nivo L še zmanjšamo do take mere, da dobimo enako število presešišč, vendar je odvod signala $x(t)$ v presečiščih večji. V primeru s Slike 3.15 bi bil tako določen nivo L =8,7 in vrednosti presečišč 68,276 μs in 318,276 μs.

Izbira nivoja pri metodi detekcije prehoda skozi nivo

Največji problem pri določevanju osnovne frekvence z metodo prehoda skozi nivo je odstopanje točk presečišč od pravilnih vrednosti. To odstopanje je posledica numeričnih napak pri izračunu oz. analizi vezja. Če ima npr. signal osnovno periodo 250 μs in se pojavi prvo presečišče pri npr. 100 μs, mora biti naslednje presečišče točno pri 350 μs. Zaradi numeričnih napak se lahko drugo presečišče pojavi npr. na intervalu med 249,9 μs in 250,1 μs. Temu se lahko izognemo tako, da poiščemo več presečišč z izbranim nivojem L in iz njih izračunamo povprečno osnovno periodo preko daljšega časvnega intervala. Potem izračunamo, koliko posamezno presečišče odstopa od pričakovane vrednosti oz. od položaja presečišča, ki je določen s povprečno osnovno periodo preko daljšega časovnega intervala.

Če je osnovna perioda enaka T in želimo določiti periodo preko n osnovnih period, bi brez numeričnih napak dobili vrednosti presečišč t_{p_0}, t_{p_1}, t_{p_2} $\ldots t_{p_n}$. Zaradi numeričnih napak dobimo vrednosti presečišč pri časih $t_{p_{0'}}$, $t_{p_{1'}}$, $t_{p_{2'}}$ $\ldots t_{p_{n'}}$. Celotno povprečno absolutno odstopanje je definirano kot

$$\Delta t_{p_{abs}} = \frac{1}{n} \sum_{i=1}^{n} \left| t_{p_i} - t_{p_{i'}} \right|, \tag{3.62}$$

celotno povprečno relativno odstopanje pa kot

$$\Delta t_{p_{rel}} = \frac{\Delta t_{p_{abs}}}{T}. \tag{3.63}$$

S spreminjanjem nivoja L lahko vplivamo na $\Delta t_{p_{abs}}$ in s tem tudi na $\Delta t_{p_{rel}}$. Izogibati se je potrebno takim nivojem, kjer je odvod signala $x(t)$ majhen ali enak 0. V teh primerih se zaradi majhnega spreminjanja signala $x(t)$ ustrezno poveča računski korak metode, s katero je izračunan $x(t)$. Zaradi velikega računskega koraka so lahko razlike $\left| t_{p_i} - t_{p_{i'}} \right|$ v (3.62) tudi velike, s čimer se zmanjša natančnost izračuna osnovne periode in posledično tudi metode za izračun stacionarnega stanja.

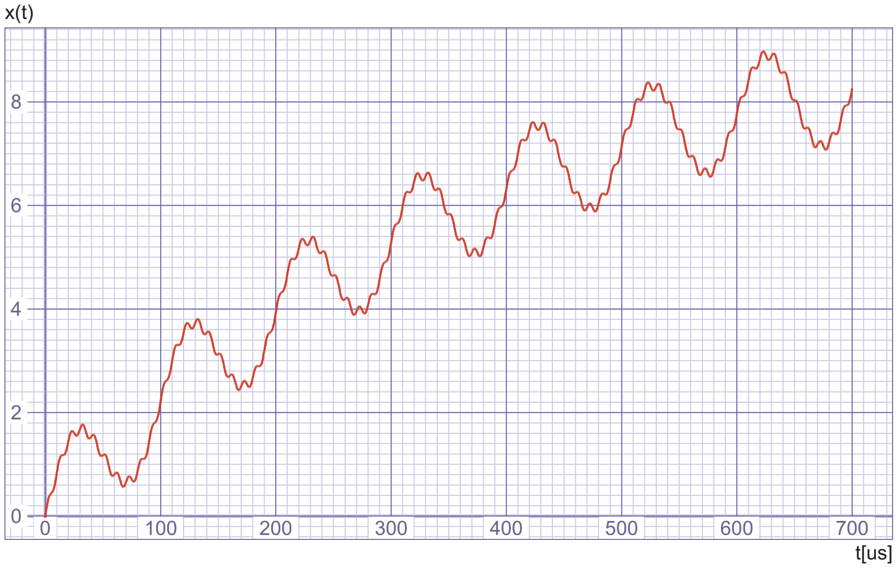
3.3.3 Detekcija kota krivulje

Naslednja možnost določitve periode signala je določitev točk, kjer krivulja oz. signal $x(t)$ z abscisno (časovno) osjo oklepa izbran kot [56]. Razlika časov med temi točkami določa osnovno periodo signala $x(t)$. Primer določevanja osnovne periode z detekcijo kota prikazuje Slika 3.16. Za ta primer bi s prej opisanima metodama prehoda skozi nivo in postopnega zniževanja nivoja težko določili osnovno periodo oz. bi bila na ta način določena osnovna perioda nenatančna.

Metoda temelji na iskanju točk, kjer krivulja oklepa z abscisno osjo določen kot. Zaradi tega moramo najprej poiskati odvod krivulje, ki predstavlja tangens kota med krivuljo in abscisno osjo. Za primer signala s Slike 3.16 je odvod prikazan na Sliki 3.17.

Enak naklon signala $x(t)$ z abscisno osjo pomeni enako vrednost odvoda, zato lahko z metodo prehoda skozi nivo in postopnega zniževanja nivoja, ki sta bili opisani v prejšnih podpoglavjih, določimo točke, kjer signal $x(t)$ z abscisno osjo oklepa enake kote. Z metodo postopnega zniževanja nivoja tako za primer s Slike 3.17 dobimo vrednost nivoja $L = 140.000$ in s tem vrednost odvodov $dx(t)/dt$ =140.000 pri t_1 =99,6 μs in t_2 =199,6 μs. Razlika časov t_1 in t_2 predstavlja osnovno periodo signala $x(t)$ na Sliki 3.16

$$T = t_2 - t_1, \tag{3.64}$$



Slika 3.16. Primer signala $x(t)$, za katerega želimo izračunati osnovno periodo z detekcijo kota krivulje

ki je enaka $100\ \mu s$.

V splošnem lahko metodo določanja osnovne periode signala z iskanjem točk enakega naklona povzamemo v naslednjih korakih:

1. Določi signal $x(t)$ (analiza električnega vezja).

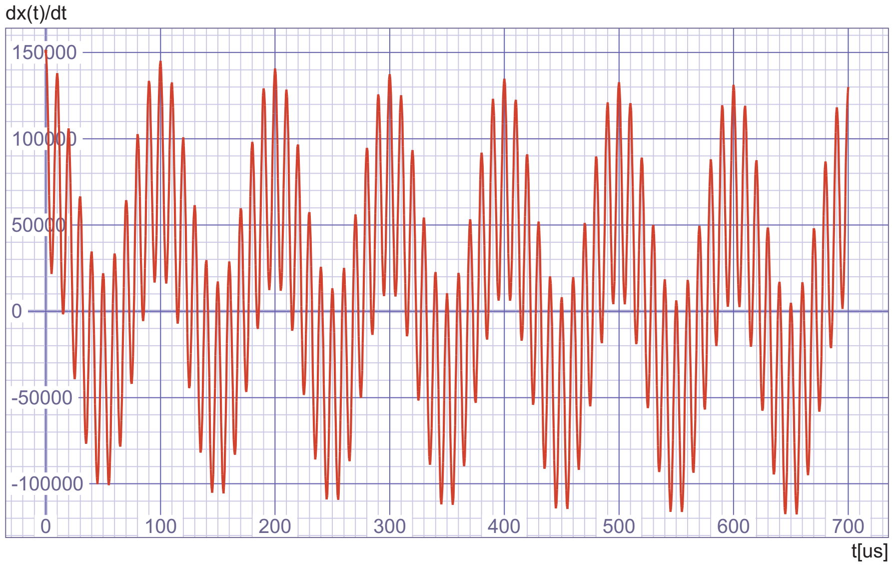
2. Izračunaj odvod signala $x(t)$.

3. Izberi neko vrednost odvoda. To vrednost označimo z $x_0^{'}$. Običajno izberemo največjo vrednost odvoda $dx(t)/dt$ oz. dovolj veliko vrednost, da dobimo zadostno število presečišč odvoda $dx(t)/dt$ z izbrano vrednostjo $x_0^{'}$.

4. Poišči točke, za katere velja $dx(t)/dt = x_0^{'}$ in jih označi s $t_1, t_2, t_3 \ldots$.

5. Izračunaj osnovno periodo signala $x(t)$ kot razliko zaporednih časov, pri katerih signal $x(t)$ z abscisno osjo oklepa enak kot:
   $T = t_{i+1} - t_i, \quad i = 1, 2 \ldots$.

Če se vrnemo na izračun stacionarnega stanja nevzbujanih vezij, lahko s pomočjo opisanih metod določimo osnovno frekvenco odziva vezja. S tem je znana tudi perioda signala. Problem izračuna stacionarnega stanja nevzbujanih vezij smo torej razdelili na dva dela: določanje periode odziva vezja in izračun stacionarnega stanja po eni od metod, opisanih v Poglavju 3.1 ali 3.2.



Slika 3.17. Odvod signala $x(t)$ s Slike 3.16

3.4 Ekstrapolacijske metode in vezja s signali s širokega frekvenčnega področja

V Poglavjih 3.2 in 3.3 je bilo že poudarjeno, da je metoda izračuna stacionarnega stanja z uporabo ekstrapolacijskih algoritmov manj primerna za vezja, ki vsebujejo signale iz zelo širokega frekvenčnega področja. Primer takega vezja je npr. mešalnik, ki ima na vhoda priključena signala frekvenc, ki sta si relativno blizu, npr. 900,0 *MHz* in 900,1 *MHz*. Na izhodu mešalnika so signali frekvenc, ki so enaki večkratnikom razlike in večkratnikom vsote vhodnih signalov. Za zgornji primer bi bila najnižja izhodna frekvenca enaka razliki vhodnih frekvenc, t.j. 100 *kHz*. Razpon frekvenčnih komponent tega vezja je od 100 *kHz* do velikostnega razreda nekaj *GHz*, oz. več 10.000-krat več kot najnižja frekvenca 100 *kHz*. Zaradi prisotnosti frekvenčnih komponent moramo pri izračunu signalov v vezju nastaviti ustrezno majhen računski korak, da zajamemo tudi signale z najvišjo frekvenco (nekaj *GHz*). Ker je osnovna frekvenca 100 *kHz*, moramo za določitev poteka signala ene same periode izračunati več deset ali sto tisoč časovnih točk. V prejšnjih poglavjih je bilo pokazano, da potrebujemo za izračun stacionarnega stanja z ekstrapolacijskimi metodami v vsaki iteraciji izračun nekaj osnovnih period, postopek pa moramo ponavljati toliko iteracij, da dosežemo stacionarno stanje. Zaradi tega ekstrapolacijske metode za tovrstna vezja niso primerne oz. z njimi ne pridobimo na hitrosti izračuna stacionarnega stanja.

Problem je prisoten tudi pri vezjih, pri katerih se v prehodnem pojavu, ko vezje še ni v stacionarnem stanju, spreminja frekvenca in/ali razmerje med trajanjem visokega in nizkega nivoja (*duty cycle*). Primer takih vezij so fazno sklenjene zanke (PLL - Phase Locked Loop) in preklopni napajalniki. V prehodnem pojavu je težko določiti frekvenco signalov, ker le-ta še ni ustaljena. Zaradi tega uspešnost ekstrapolacijskih metod, ki zahtevajo poznavanje osnovne frekvence signalov v stacionarnem stanju, ni zagotovljena. Fazno sklenjene zanke in preklopne napajalnike lahko tudi uvrstimo v skupino vezij s signali iz širokega frekvenčnega področja.

Za vezja, opisana v tem podpoglavju, je primernejša izbira metode harmoničnega ravnovesja, ki temelji na predstavitvi signalov z vsoto frekvenčnih komponent. Pri tem ni pomembna širina frekvenčnega področja, ampak število frekvenčnih komponent.

Izbira metode je torej odvisna ne samo od vrste vezja, ampak tudi od signalov, ki so v tem vezju prisotni. Računsko manj zahtevne so vsekakor ekstrapolacijske metode, enostavnejša je tudi sama izvedba, vendar so v določenih primerih neuspešne. Tedaj je potrebno uporabiti druge metode za izračun stacionarnega stanja, npr. metodo harmoničnega ravnovesja, ki je bila opisana v Poglavju 3.1.

4

Izračun stacionarnega stanja s programskim paketom SPICE

V tem poglavju bo utemeljen izbor metode za izračun stacionarnega stanja, ki je bil implementiran v programski paket SPICE. Metoda je bila izbrana glede na hitrost in točnost izračuna stacionarnega stanja, zahtevnost implementacije in same specifike programskega paketa SPICE. V nadaljevanju je predstavljena simulacija izbranih metod, preden bi bile le-te vgrajene v programski paket. Metode, ki so se uvrstile v ožji izbor, so bile simulirane kot skripta v programskem jeziku paketa SPICE NUTMEG (metoda harmoničnega ravnovesja) in kot samostojni program v programskem jeziku C, ki je obdeloval rezultate analize vezja in na podlagi le-teh ustrezno spreminjal parametre za nove analize vezja (računanje stacionarnega stanja z ekstrapolacijskimi metodami). Na podlagi treh kriterijev je bil kot najustreznejši izbran algoritem epsilon. Na koncu poglavja je opisana sama implementacija tega algoritma v programski paket SPICE.

4.1 Merila za izbor metode za implementacijo v programski paket SPICE

Pri izbiri primerne metode je bil glavni kriterij specifika programskega paketa SPICE, kamor je bila metoda za izračun stacionarnega stanja vgrajena. Upoštevane so bile zahteve za implementacijo posameznih metod, zahtevnost sprememb programskega paketa SPICE, enostavnost uporabe in potreba po ohranitni kompatibilnosti simulatorja za nazaj.

Prva metoda, predstavljena v Poglavju 3.1, je metoda harmoničnega ravnovesja. Pri izračunu je potrebno izračunati Jacobijevo matriko (enačba (3.14)), za kar potrebujemo izračun parcialnih odvodov po vozliščnih napetostih V_i. Za izračun le-teh je potrebna korenita sprememba jedra simulatorja oz. dopolnitev modelov elementov električnih vezij, na podlagi katerih bi lahko izračunali vrednosti parcialnih odvodov, ki nastopajo v Jacobijevi matriki. Drugi, enostavnejši, a žal manj natančen in numerično manj primeren način je numerični izračun parcialnih odvodov. Izračunamo ga z diferencami glede na vrednost vozliščnih napetosti v prejšnji iteraciji Newton-Raphsonovega postopka (3.15). Omenjena možnost izračuna Jacobijeve matrike je bila uporabljena pri simulaciji primernosti metode v [45], kar bo opisano tudi v Poglavju 4.2.

Ekstrapolacijske metode, ki so bile predstavljene v Poglavju 3.2, so enostavnejše za vgradnjo v obstječi programski paket, saj ne posegajo v samo jedro simulatorja. Za izračun stacionarnega stanja z ekstrapolacijskimi metodami je potrebno poznavanje odziva vezja v nekaj periodah vezja. Zato je izračun po tej metodi v bistvu nadgradnja obstoječe tranzientne analize. Izračun stacionarnega stanja z uporabo ekstrapolacijskih metod lahko izvedemo kot novo analizo, znotraj katere se v okviru iteracijskega postopka izvaja več tranzientnih analiz.

Drugi kriterij, ki je bil upoštevan pri izboru metode za izračun stacionarnega stanja, je točnost izračuna oz. razlika odzivov med metodo hitrega izračuna stacionarnega stanja in tranzientno analizo, ki naj bo manjša od 1 %. Razlike, ki bi bile manjše od 0,1 %, sicer kažejo na natančen izračun stacionarnega stanja, vendar moramo upoštevati še dejstvo, da so tudi modeli električnih elementov aproksimacija realnih elementov. Zaradi tega ne pričakujemo natačnosti, ki bi bila boljša od 0,1 %.

Pospešitev izračuna stacionarnega stanja v primerjavi z direktno tranzientno analizo je tretji kriterij za izbor najustreznejše metode. Glavni cilj doktorske disertacije je optimizacija vezja v doglednem času (nekaj ur, največ nekaj dni). Zato je ob časih izračuna stacionarnega stanja tipičnih vezij (nekaj minut ali nekaj ur) in ob tipično nekaj tisoč iteracijah optimizacijske zanke potrebno pospešiti izračun stacionarnega stanja za vsaj desetkrat.

4.2 Simulacija metod za izračun stacionarnega stanja električnih vezij

Pred samo implementacijo metode za izračun stacionarnega stanja sta bili metodi za izračun stacionarnega stanja simulirani oz. preizkušeni, tako da smo dodatno preverili njuno ustreznost. Metoda harmoničnega ravnovesja je bila simulirana v skriptnem jeziku NUTMEG, ki je del programskega paketa SPICE in omogoča programiranje in izvajanje ukazov, ki jih omogoča programsko orodje SPICE. Vse različice ekstrapolacijskih metod za izračun stacionarnega stanja, ki so bile opisane v Poglavju 3.2 (algoritem epsilon, rho, theta in topološki algoritem epsilon), so bile zaradi obsežnejšega algoritma izvedene v samostojnem programu v programskem jeziku C, znotraj katerega so se izvajali klici programskega paketa SPICE, ki je izvajal zahtevane analize vezja.

4.2.1 Simulacija metode harmoničnega ravnovesja

Metoda harmoničnega ravnovesja je opisana v [45]. Temelji na razdelitvi električnega vezja na nelinearni in linearni del, za katera ločeno določimo stanje vezja in nato z iteracijskim postopkom ta dva dela vezja med seboj uskladimo, tako da celotno vezje izpolnjuje Kirchhoffova zakona. Za iteracijski postopek je potrebno izračunati Jacobijevo matriko (3.14). Izračun parcialnih odvodov v Jacobijevi matriki je bil izveden na osnovi diferenc med zaporednima iteracijama (k) Newton-Raphsonovega postopka

$$\frac{\partial f_i}{\partial V_j} \doteq \frac{f_i^{(k+1)} - f_i^{(k)}}{V_j^{(k+1)} - V_j^{(k)}}. \tag{4.1}$$

Izračun parcialnih odvodov je potrebno izračunati na zgoraj opisan način, ker sam simulator ne ponuja direktne informacije o odvodu vozliščnih napetosti in vozliščnih tokov glede na spremembe le-teh v prejšnji iteraciji. Zaradi tega je bilo potrebno izračunati parcialne odvode numerično, kar v rezultate vnaša dodatne numerične napake in s tem tudi zmanjšuje natančnost samega izračuna stacionarnega stanja vezja, ki ga analiziramo.

Postopek simulacije metode je potekal v naslednjih korakih:

1. Razdelitev vezja na nelinearni in linearni del. Vsak od obeh delov vezja je bil združen v podvezje, katerim lahko kot vhodni podatek (parameter) nastavljamo lastnosti.

2. Analiza nelinearnega dela vezja v časovnem prostoru (tranzientna analiza).

3. Analiza linearnega dela vezja v frekvenčnem prostoru. Izvesti je bilo potrebno malosignalno frekvenčno analizo, za kar v splošnem uporabimo ukaz *ac*. Ker smo uporabili modificirano metodo harmoničnega ravnovesja (Poglavje 3.1.1), zaradi enostavnosti linearnega dela vezja le-tega ni bilo treba posebej analizirati, ampak lahko njegovo karakteristiko določimo analitično, saj vsebuje samo napetostne vire (V_{HBi}) in upore (R_{HBi}), $i = 1, 2 \ldots n$.

4. Fourierjeva transformacija rezultatov analize nelinearnega dela vezja.

5. Newton-Raphsonov iteracijski postopek (3.15) in sprememba vrednosti parametrov linearnega dela vezja (V_{HB_i} in R_{HB_i}, $i = 1, 2 \ldots n$, Slika 3.5).

6. Preverjanje izpolnjevanja zahteve (3.16). Če je zahteva izpolnjena, je postopek izračuna stacionarnega stanja z metodo harmoničnega ravnovesja uspešno zaključen.

Razdelitev vezja

Po modificirani metodi harmoničnega ravnovesja (Poglavje 3.1.1) je bilo celotno vezje, ki ga analiziramo, označeno kot nelinearno podvezje. Nato v vsako vozlišče celotnega vezja (kar ustreza tudi nelinearnemu delu vezja) dodamo linearni del vezja. Ta je sestavljen iz zaporedno vezanega upora R_{HBi} in napetostnega vira V_{HBi}.

Ta metoda je bila izbrana zato, ker je za splošno vezje težje izvesti algoritem, ki bo avtomatsko razdelil vezje na nelinearni in linearni del. Metoda je zaradi tega počasnejša, ker analiziramo vse elemente vezja v časovnem prostoru, tudi linearne elemente. Pridobimo pa na splošnosti, saj je postopek "delitve vezja" uporaben na kakršnem koli vezju. Druga utemeljitev takšnega načina delitve vezja pa je dejstvo, da je večina elementov v vezjih, sploh v integriranih vezjih, nelinearnih (tranzistorji, diode ...), zelo malo elementov pa je res linearnih (linearni upori, kondenzatorji ...). V integriranih vezjih upore in kondenzatorje realiziramo na način, ki ne zagotavlja linearnosti njihovih karakteristik na širšem območju vhodnih parametrov, zato so ti elementi v splošnem tudi nelinearni in je način delitve vezja, ki je bil uporabljen, še bolj upravičen.

Analiza nelinearnega dela vezja

Nelinearno podvezje (v je v našem primeru delitve vezja kar celotno vezje) analiziramo v časovnem prostoru s tranzientno analizo (analiza *tran* programskega paketa SPICE). S parametroma t_Δ in t_m (Poglavje 2.2.1) moramo zagotoviti, da je računski korak tranzientne analize ustrezno majhen, da lahko kasneje pri računanju Fourierjeve transformacije glede na natančnost izračuna določimo zahtevano število frekvenčnih komponent (enačba (3.2)).

Analiza linearnega dela vezja

Linearni del vezja vsebuje upore R_{HB_i} in napetostne vire V_{HB_i}, $i = 1, 2, 3 \ldots n$ (Slika 3.5). Zato lahko ta del analiziramo analitično. Za vsak par R_{HB_i} in V_{HB_i} na Sliki 3.5 upoštevamo napetostni in tokovni Kirchhoffov zakon v frekvenčnem prostoru in dobimo enakost

$$-V_i(j\omega) - R_{HB_i} I_i(j\omega) + V_{HB_i}(j\omega) = 0, \quad i = 1, 2, 3 \dots n. \tag{4.2}$$

Odvisnosti tokov $I_i(j\omega)$ od napetosti neodvisnih virov $V_{HB_i}(j\omega)$, ki jih izpeljemo iz (4.2)

$$I_i(j\omega) = \frac{V_{HB_i}(j\omega) - V_i(j\omega)}{R_{HB_i}}, \quad i = 1, 2, 3 \dots n, \tag{4.3}$$

predstavljajo tok $\mathbf{I}_L$ (enačba (3.11)) linearnega dela vezja

$$\mathbf{I}_L(j\omega) = \begin{bmatrix} I_1(j\omega) \\ I_2(j\omega) \\ I_3(j\omega) \\ \vdots \\ I_n(j\omega) \end{bmatrix}. \tag{4.4}$$

Z vsako iteracijo se spreminjajo upornosti R_{HB_i} (enačba (3.19)), zato se spreminja tudi odvisnost (4.2) in posledično (4.3).

Fourierjeva transformacija

Rezultati analize nelinearnega dela vezja, ki so izračunani v časovnem prostoru, so s Fourierjevo transformacijo prenešeni v frekvenčni prostor. Preden transformacijo izvedemo, je potrebno rezultate tranzientne analize linearizirati, t. j. interpolirati, da so njihove vrednosti porazdeljene ekvidistantno v času. Že pri izvajanju tranzientne analize moramo paziti, da je časovni korak dovolj majhen, da rezultat vsebuje dovolj frekvenčnih komponent za zahtevano natančnost izračuna stacionarnega stanja. Pred klicem ukaza *spec*, ki izvede Fourierjevo transformacijo, je treba določiti še vrsto okenske funkcije. Uporabljeno je bilo Hannovo okno (Hanning window).

Newton-Raphsonov postopek

Newton-Raphsonov postopek (3.15) poteka v frekvenčnem prostoru. Za simulacijo izračuna stacionarnega stanja z metodo harmoničnega ravnovesja je bila upoštevana osnovna in osem višjeharmonskih komponent frekvence vhodnega signala. Začetne vrednosti upornosti R_{HB_i} so bile enake 1 $k\Omega$, parameter λ v (3.19) pa 0,5.

Izpolnjevanje ustavitvenega kriterija

Za končanje Newton-Raphsonovega postopka mora biti doseženo maksimalnego število iteracij (izračun stacionarnega stanja je neuspešen) ali pa dosežen pogoj v (3.16) (uspešen izračun stacionarnega stanja). Maksimalno število iteracij iteracijskega postopka je bilo nastavljeno na 50, parameter δ v (3.16), ki določa natančnost izračuna, pa je imel vrednosti med 10^{-7} in $5 \cdot 10^{-4}$ (odvisno od vrste vezja).

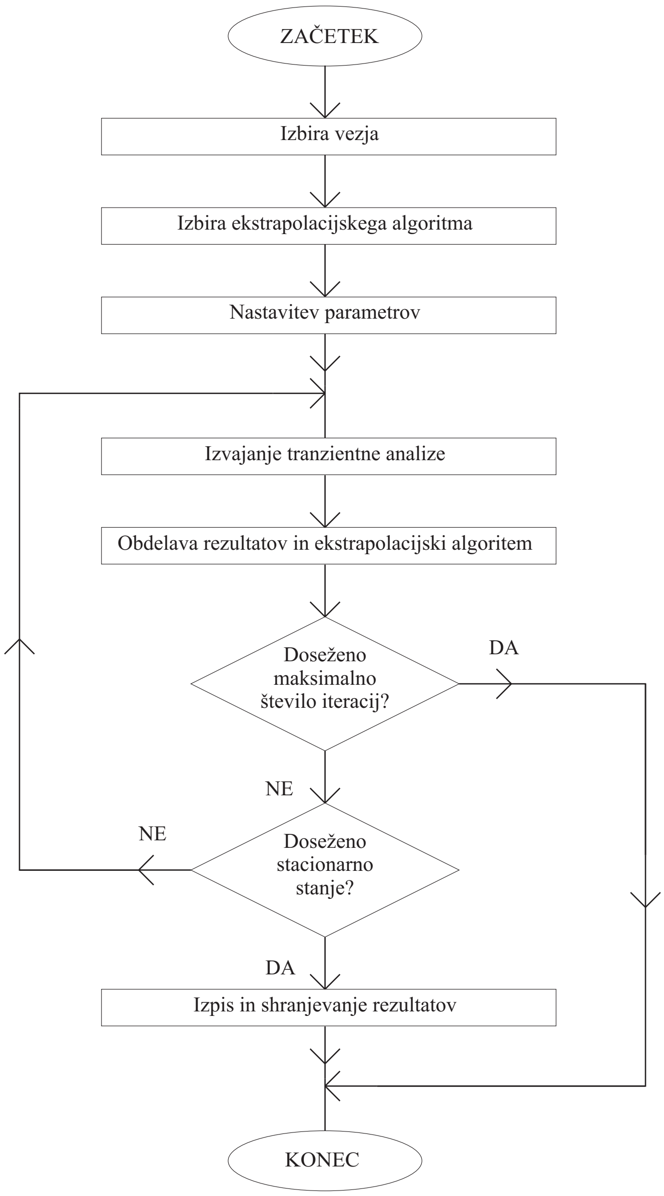
4.2.2 Simulacija ekstrapolacijskih metod za izračun stacionarnega stanja

Simulacija izračuna stacionarnega stanja z uporabo ekstrapolacijskih metod je bila izvedena s pomočjo posebnega programa v programskem jeziku C. S programom najprej izberemo vezje, ki mu želimo izračunati stacionarno stanje, vrsto ekstrapolacijskega algoritma (epsilon, rho, theta ali topološki algoritem epsilon) in nastavimo parametre algoritma. Nato se v zanki izvajajo klici programskega paketa SPICE OPUS, ki izvaja tranzientno analizo vezja. Rezultati se shranjujejo v binarno datoteko. Rezultate analize iz datoteke program obdela, oz. glede na izbran ekstrapolacijski algoritem nastavi začetno stanje vezja za novo tranzientno analizo. Na koncu zanke preveri, če je izpolnjen ustavitveni kriterij (če je bilo izračunano stacionarno stanje ali če je bilo doseženo maksimalno število iteracij). Če ustavitveni kriterij ni izpolnjen, program ponovi vse korake v zanki. Po izhodu iz zanke program izriše rezultate analize (stacionarno stanje vezje), izpiše število iteracij, ki so bile potrebne za izračun in vse rezultate shrani v datoteko za morebitno nadaljnjo uporabo.

Diagram poteka programa za simulacijo izračuna stacionarnega stanja s pomočjo ekstrapolacijskih metod prikazuje Slika 4.1.

Izbira vezja

Vezja, ki jih želimo analizirati, je potrebno prej pripraviti, da jih lahko analiziramo z uporabo ekstrapolacijskih metod. Poskrbeti je potrebno, da se pri tranzientni analizi izračunajo vozliščne napetosti in vejni tokovi točno v časovnih točkah, ki jih potrebujemo pri ektrapolacijskem algoritmu (enačba (3.23)). V ta namen v vezje dodamo vzporedno vezana upor in neodvisen tokovni vir pravokotnega signala. Ob časih, kjer želimo izračun stanja vezja, ima tokovni vir ostro spremembo izhodnega toka, s čimer prisilimo simulator vezja, da se v teh časovnih točkah stanje vezja res izračuna. Če izračuna vezja v zahtevanih časovnih točkah ne bi imeli, bi te vrednosti sicer lahko izračunali naknadno z interpolacijo, a zaradi numeričnih napak in zaradi dejstva, da vezje ni v resnici izračunano v teh točkah, pri ekstrapolacijskem algoritmu ne bi dobili pravilnih rezultatov.



Slika 4.1. Diagram poteka programa za simulacijo izračuna stacionarnega stanja s pomočjo ekstrapolacijskih metod

Izbira ekstrapolacijskega algoritma

Izbiramo lahko med ekstrapolacijskimi algoritmi, opisanimi v Poglavju 3.2: algoritmi epsilon, rho, theta in topološkim algoritmom epsilon. Glede na izbrani algoritem je potrebno nastaviti tudi ustrezne parametre tega algoritma.

Nastavitev parametrov

Pred izvajanjem tranzientne analize je potrebno glede na izbran ekstrapolacijski algoritem, vezje in zahtevano natančnost nastaviti parametre ekstrapolacijskega algoritma. Ti parametri so:
frequency je frekvenca vzbujanja vezja. Če vezje ni vzbujano, je potrebno ustrezno nastaviti parameter *DetFreq*.
DetFreq določa, ali program iz odziva vezja določi frekvenco ali ne. Če je vezje vzbujano, frekvence ne določamo in nastavimo vrednost parametra na 0; če je vezje avtonomno, moramo vezju določiti frekvenco. V tem primeru nastavimo vrednost parametra na 1.
tdel in ***tdelSec*** predstavljata začetna časa, pri katerem vzamemo prvi vzorec odziva vezja in ustreza vrednosti t_{del} v (3.23), pri čemer se *tdel* nanaša na t_{del} za prvo iteracijo ekstrapolacijskega algoritma, *tdelSec* pa na t_{del} za vse naslednje iteracije.
Parametri ***NumPointsMin***, ***NumPointsMax*** in ***PointsStep***. Ti parametri določajo število period in s tem čas analize vezja oz. vrednost m v (3.24). Začetno število period m je enako parametru *NumPointsMin*. Z vsako naslednjo iteracijo ekstrapolacijskega algoritma se število period poveča za *PointsStep*, dokler število period ne doseže vrednosti *NumPointsMax*. Število period ostane nato nespremenjeno do konca ekstrapolacijskega algoritma.
PeriodsPerPoint določa število period med dvema zaporednima vzorcema oz. točkama, ki jih potrebujemo pri ekstrapolacijskem algoritmu. Vrednost tega parametra je naravno število in ima običajno vrednost 1, kar pomeni, da je signal vzorčen vsako osnovno periodo. Povečanje parametra *PeriodsPerPoint* efektivno pomeni zmanjšanje osnovne frekvence za faktor *PeriodsPerPoint*. Spreminjanje tega parametra pride do izraza pri vezjih, pri katerih je sprememba odziva vezja v eni periodi zelo majhna, v npr. štirih periodah pa je sprememba že dovolj velika, da lahko z ekstrapolacijskim algoritmom izračunamo novo začetno stanje vezja za naslednjo analizo.
Parameter ***FreqMax*** določa maksimalno frekvenčno komponento v odzivu vezja. Vrednosti parametrov t_Δ in t_m pri tranzientni analizi (glej Poglavje 2.2.1) sta podana z vrednostjo parametra *FreqMax*, in sicer: $t_\Delta = t_m = \frac{1}{2 \cdot FreqMax}$.
Vrednosti parametrov δ_a in δ_r v (3.51) sta enaki parametroma ***EpsStopAbsTol*** in ***EpsStopRelTol*** in podajata ustavitveni kriterij ekstrapolacijskega algoritma oz. natančnost izračuna sta-

cionarnega stanja.
S parametrom ***MaxIters*** nastavimo maksimalno število iteracij ekstrapolacijskega algoritma. Če stacionarno stanje ni doseženo v številu iteracij, podanim s tem parametrom, se ekstrapolacijski algoritem zaključi (glej Sliko 4.1 - "Doseženo maksimalno število iteracij?").

Izvajanje tranzientne analize

Na začetku izvajanja paketne datoteke, ki izvede tranzientno analizo, vključimo datoteko *settings.dat*, ki vsebuje vrednosti zgoraj opisanih parametrov. Nato naložimo (ukaz *load*) datoteko *initial.raw*, kjer je podano začetno stanje vezja za trenutno iteracijo ekstrapolacijskega algoritma. S stavkom *ic* (Initial Conditions) določimo vektor začetnega stanja, ki ga bo upoštevala tranzientna analiza. Na koncu glede na vrednosti parametrov poženemo ustrezno nastavljeno tranzientno analizo. Rezultate analize v binarni obliki shranimo (ukaz *write*) v datoteko *result.raw*.

Ekstrapolacijski algoritem

Na rezultatih tranzientne analize, ki so shranjeni v datoteki *result.raw*, glede na vrednosti izbranih parametrov izvedemo vzorčenje (3.23). Uporabimo izbrani ekstrapolacijski algoritem. Nastavimo novo začetno stanje vezja in ga shranimo v datoteko *initial.raw* za naslednjo tranzientno analizo vezja.

Preverjanje zaustavitvenega kriterija

Preverimo, ali odziv vezja izpolnjuje zaustavitvene kriterije (3.51). Če je ta pogoj izpolnjen, smo izračunali stacionarno stanje vezja, v nasprotmem primeru pa izvedemo naslednjo iteracijo ekstrapolacijskega algoritma. Del zaustavitvenega kriterija je tudi preverjanje, če smo dosegli maksimalno število iteracij ekstrapolacijskega algoritma (parameter *MaxIters*).

Izpis in shranjevanje rezultatov

V primeru uspešnega izračuna stacionarnega stanja vezja se izpišejo podatki o številu iteracij ekstrapolacijskega algoritma in skupnem številu period, ki so bile potrebne za izračun stacionarnega stanja vezja. S temi podatki lahko primerjamo stopnjo pospešitve izračuna stacionarnega stanja v primerjavi z običajno tranzientno analizo. Izrišejo se še izbrani signali vezja, ki nas zanimajo (npr. izhodna napetost, napajalni tok). Vse napetosti in tokovi, ki so rezultat zadnje tranzientne analize, se shranijo v končno datoteko *result.raw*, ki jo lahko uporabljamo za nadaljnjo uporabo (izris ostalih signalov vezja, primerjava z običajno tranzientno analizo...).

V tem podpoglavju smo predstavili simulacijo metod za izračun stacionarnega stanja električnih vezij. S primerjavo rezultatov simulacije na nekaj primerih vezij iz Poglavja 5 in na podlagi meril za izbor, podanih v Poglavju 4.1, bomo v naslednjem poglavju utemeljili izbor metode izračuna stacionarnega stanja, ki smo jo vgradili v programski paket SPICE OPUS.

4.3 Izbor najustreznejše metode

Izbor najustreznejše metode, ki je bila vgrajena v programski paket SPICE OPUS, je bil določen na podlagi treh kriterijev. Vsakemu od teh treh kriterijev smo določili utež, ki podaja vlogo kriterija pri celotnem izboru metode.

1. kriterij: Specifika programskega paketa SPICE (utež $w_1 = 10$ - najpomembnejše)
(zahtevnost sprememb programskega paketa SPICE pri vgradnji izbrane metode)
Ocenjena je zahtevnost pri implementaciji metode v programski paket SPICE OPUS. Vrednosti tega kriterija so lahko med 1 in 5, in sicer:
$k_1 = 1$ - zelo nezahtevna implementacija, zelo malo sprememb in dodatkov programske kode (do 10 ur [1]),
$k_1 = 2$ - nezahtevna implementacija (10 do 30 ur),
$k_1 = 3$ - srednje zahtevna implementacija (30 do 100 ur),
$k_1 = 4$ - zahtevna implementacija (100 do 500 ur),
$k_1 = 5$ - zelo zahtevna implementacija, potreben temeljit poseg v jedro simulatorja, popravljanje oz. dodajanje funkcionalnosti modelov električnih elementov (več kot 500 ur).

2. kriterij: Točnost izračuna (utež $w_2 = 5$ - srednja pomembnost)
Primerjana je absolutna vrednost relativne razlike zgornje vrednosti ovojnice izhodnega signala pri izračunu z uporabo direktne tranzientne analize v primerjavi z metodo za hiter izračun stacionarnega stanja. Kriterij ima lahko vrednosti med 1 in 5, in sicer:
$k_2 = 1$ - popolnoma identičen rezultat - absolutna vrednost relativne razlike je enaka 0,00 %,
$k_2 = 2$ - absolutna vrednost relativne razlike je med vključno 0,01 % in vključno 0,20 %,

[1] Ocenjen čas za programiranje in testiranje metode

$k_2 = 3$ - absolutna vrednost relativne razlike je med vključno 0,21 % in vključno 0,50 %,
$k_2 = 4$ - absolutna vrednost relativne razlike je med vključno 0,51 % in vključno 1,00 %,
$k_2 = 5$ - absolutna vrednost relativne razlike je večja kot 1,00 %.

3. kriterij: Hitrost izračuna (utež $w_3 = 1$ - manjša pomembnost)
Pri enaki zahtevani natančnosti izračuna je bil primerjan čas izračuna stacionarnega stanja pri direktni tranzientni analizi in z uporabo metode za hiter izračun stacionarnega stanja. Pri ekstrapolacijskih metodah je bilo merilo hitrosti izračuna je razmerje števila osnovnih period signalov, pri metodi harmoničnega ravnovesja pa razmeje med časom izračuna stacionarnega stanja. Vrednosti kriterija so med 1 on 5, in sicer:
$k_3 = 1$ - več kot 20-kratna pospešitev izračuna stacionarnega stanja,
$k_3 = 2$ - pospešitev izračuna za faktor med vključno 20 in 5,
$k_3 = 3$ - pospešitev izračuna za faktor med vključno 5 in 2,
$k_3 = 4$ - pospešitev izračuna za faktor med vključno 2 in 1,
$k_3 = 5$ - upočasnitev izračuna stacionarnega stanja (faktor manjši ali enak 1).

Celotno vrednost kriterija (4.5) za določeno metodo sestavimo tako, da pomnožimo vrednost posameznega kriterija z njegovo utežjo. Prvi člen (1. kriterij) moramo še normirati (oz. pomnožiti s številom vezij n), da s številom testnih vezij ne pada vpliv 1. kriterija. Nato seštejemo tako dobljene produkte preko vseh treh kriterijev in vseh testnih vezij, ki jih uporabimo za izbor metode. Vrednost kriterija posamezne metode izračunamo z enačbo

$$K_{metode} = k_1 \cdot w_1 \cdot n + \sum_{i=1}^{n} k_{2,i} \cdot w_2 + \sum_{i=1}^{n} k_{3,i} \cdot w_3, \qquad (4.5)$$

pri čemer predstavljajo indeksi i vezja, na katerih preizkušamo ustreznost metode. Metodo preizkušamo na n vezjih.

Kriteriji so zastavljeni tako, da manjša vrednost posameznega kriterija predstavlja boljšo vrednost. Prav tako manjša vrednost celotnega kriterija metode K_{metode} predstavlja boljšo vrednost za izbor metode.

V nadaljevanju bomo podali rezultate oz. vrednosti posameznih kriterijev za obravnavane metode hitrega izračuna stacionarnega stanja za več testnih vezij iz Poglavja 5. Izbranih je bilo pet ($n = 5$) testnih vezij (Vezje 3, 4, 5, 6 in 7).

V spodnji tabeli so najprej za izbrana testna vezja za ekstrapolacijske algoritme podana števila period v primerjavi s tranzientno analizo, za metodo harmoničnega ravnovesja pa časi analiz v primerjavi s časi tranzientnih analiz.

	rel. nat. δ_r	Število period tran-zientna analiza	algo-ritem epsilon	algo-ritem rho	algo-ritem theta	topološki algoritem epsilon	Čas izračuna tran-zientna analiza	ravot. harm. analiza
Vezje 3: Napetostni množilnik	$5\cdot 10^{-6}$	90	30,1	30,1	78,1	30,1	150 s	38 s
Vezje 4: Ozkopasovni filter	10^{-5}	496	24	94	182,25	24	48 s	10 s
Vezje 5:Super-ozkopasovni filt. s kvarcom	$5\cdot 10^{-4}$	48.332	54	87	144	54	1.110 s	4,9 s
Vezje 6: Preklopni napajalnik	10^{-7}	15.300	30	30	72	30	152 s	3,8 s
Vezje 7: Greinacherjev usmernik	10^{-6}	73.800	312	220	594	1.644	460 s	25 s

Tabela 4.1. Število period in čas izračuna metod hitrega izračuna stacionarnega stanja v primerjavi s tranzientno analizo

Tabela 4.2 pa za vse algoritme prikazuje vrednosti treh kriterijev, na podlagi katerih smo določili najustreznejši algoritem. V oklepajih so navedene še vrednosti, iz katerih smo posameznemu kriteriju določili eno od vrednosti kriterija (1, 2, 3, 4 ali 5).

		algoritem epsilon	algoritem rho	algoritem theta	topološki alg. eps.	ravn. harm. analiza
Kriterij 1 (k_1; $w_1 = 10$)		3	3	3	3	5
Kriterij 2 (k_2; $w_2 = 5$)	Vezje 3	4 (0,96 %)	3 (0,46 %)	4 (0,52 %)	4 (0,68 %)	3 (0,31 %)
	Vezje 4	3 (0,22 %)	3 (0,25 %)	2 (0,19 %)	4 (0,57 %)	3 (0,37 %)
	Vezje 5	3 (0,26 %)	3 (0,38 %)	3 (0,30 %)	3 (0,38 %)	3 (0,29 %)
	Vezje 6	1 (0,00 %)	2 (0,12 %)	3 (0,23 %)	2 (0,09 %)	2 (0,18 %)
	Vezje 7	1 (0,00 %)	2 (0,08 %)	2 (0,11 %)	1 (0,00 %)	2 (0,14 %)
Kriterij 3 (k_3; $w_3 = 1$)	Vezje 3	3 (2,99)	3 (2,99)	4 (1,15)	3 (2,99)	3 (3,95)
	Vezje 4	1 (20,7)	2 (5,28)	3 (2,72)	1 (20,7)	3 (4,80)
	Vezje 5	1 (895)	1 (556)	1 (336)	1 (895)	1 (227)
	Vezje 6	1 (510)	1 (510)	1 (213)	1 (510)	1 (40,0)
	Vezje 7	1 (237)	1 (335)	1 (124)	1 (44,9)	2 (18,4)
K_{metode}		217	223	230	227	325

Tabela 4.2. Vrednosi treh kriterijev za pet izbranih testnih vezij in vrednosti celotnega sestavljenega kriterija posamezne metode po enačbi (4.5). Pri kriteriju 2 in 3 so v oklepaju navedeni podatki, iz katerih so bile določene vrednosti posameznega kriterija.

Ekstrapolacijske metode so se glede kriterija, sestavljenega na podlagi zahtevnosti implementacije, natančnosti in hitrosti izračuna stacionarnega stanja (enačba (4.5)), izkazale za najustreznejše za uporabo in vgradnjo v programski paket SPICE OPUS. Pri vseh ekstrapolacijskih metodah je vrednost kriterija K_{metode} manjša kot pri ravnotežni harmonski analizi. Izmed ekstrapolacijskih metod je bil pa najprimernejši algoritem epsilon (vrednost kriterija $K_{metode} = 217$ je najmanjša).

Na podlagi opisanih meril za izbor najustreznejše metode se je za najboljšega izkazal torej algoritem epsilon. Izbrani algoritem je bil zato vgrajen v programski paket SPICE. V naslednjem podpoglavju bo opisana implementacija te metode v programski paket SPICE in njena uporaba pri analizi vezij.

4.4 Implementacija metode v programski paket SPICE

V programski paket SPICE OPUS smo vgradili analizo za izračun stacionarnega stanja električnih vezij *ssse* (Steady-State Shooting by Extrapolation). Vgrajena je ekstrapolacijska metoda z uporabo algoritma epsilon, ki uspešno določi stacionarno stanje vezij, opisanih v Poglavju 5. Za določevanje frekvence avtonomnih vezij je uporabljena metoda detekcije prehoda skozi nivo, s katero lahko določimo tudi frekvenco neavtonomnih vezij. Sintaksa analize *ssse* je podana z

```
ssse v(<posnode>[,<negnode>]) [<level> [<step> [<skip> [<periods>]]]] [history]
```

Prvi obvezni parameter *v(<posnode>)* oz. *v(<posnode>[,<negnode>])* definira signal $x(t)$ v Poglavju 3.3.2. Signal $x(t)$ je definiran kot napetost vozlišča *<posnode>*
$x(t) = v(< posnode >)$
oz. kot razlika med vozliščema *<posnode>* in *<negnode>*
$x(t) = v(< posnode >) - v(< negnode >)$.
Če je vezje vzbujano (ni avtonomno), za signal $x(t)$ definiramo signal vzbujanja, katerega frekvenco poznamo. Pri avtonomnih vezjih kot $x(t)$ ponavadi podamo izhodni signal vezja oz. tisti signal, ki nas zanima kot rezultat analize *ssse*. Na podlagi tega signala namreč analiza *ssse* določi osnovno frekvenco, ki jo potrebujemo pri ekstrapolacijskem postopku za vzorčenje signala.

Ostali parametri analize *ssse* so neobvezni. Parameter *<level>* podaja nivo signala $x(t)$, pri katerem metoda nivo išče presečišča, s pomočjo katerih se določi frekvenca signala (Poglavje 3.3.2). Če ta vrednost ni podana, je privzeta vrednost enaka *<level>*$= 0$.
S parametrom *<step>* določimo časovni korak pri izračunu odziva vezja. Parameter ustreza prvemu parametru $t_{\triangle}$ tranzientne analize *tran*. Privzeta vrednost parametra je *<step>*$= 1$.

Začetni čas t_{del} v (3.23) je enak parametru <*skip*>. Če paramatra ne podamo, je začetni čas enak <*skip*>$= 0$ oz. $t_{del} = 0$.
Parameter <*periods*> določa število period za analizo vezja oz. vrednost m v (3.23). Če parameter ni podan, je njegova vrednost enaka najmanjši vrednosti, ki je potrebna za pravilno delovanje ekstrapolacijskega algoritma, t.j. $m = 2$.
Če na koncu stravka analize *ssse* dodamo besedo *history*, dobimo kot rezultat analize časovne poteke signalov v vezju v vseh iteracijah ekstrapolacijskega algoritma.
Maksimalno število iteracij ekstrapolacijskega postopka k_{max} v Algoritmu (3.3) določimo s postavitvijo vrednosti *itl2* v okolju SPICE OPUS z ukazom *.options*.
Absolutna δ_a in relativna δ_r natančnost (enačba (3.51)) izračuna stacionarnega stanja vezja z ekstrapolacijsko metodo sta podani z vrednostmi *sssetol*, *abstol* in *reltol*, in sicer:
$\delta_a =$*sssetol* * *abstol*
in
$\delta_r =$*sssetol* * *reltol* .
Vrednosti *sssetol*, *abstol* in *reltol* nastavimo s stavkom *.options*.
Pred izvajanjem analize *ssse* je priporočljivo izklopiti funkcijo programskega paketa SPICE OPUS za detekcijo numeričnih oscilacij, kar storimo s stavkom
set xmumult=1 .

Spodnji primer kaže nastavitve parametrov in analizo *ssse* za testno vzbujano vezje. Vhodni signal je priključen med vozliščema *inp* in *inn*. Začetni časovni korak je enak $t_{\Delta} = 2\ ps$, začetni čas $t_{del} = 0$, število period pa je $m = 6$. Zahtevamo shranjevanje vseh iteracij ekstrapolacijskega algoritma. Maksimalno število iteracij je $k_{max} = 1000$. Absolutna in relativna natančnost izračuna sta enaki $\delta_a = 10^{-6}$ in $\delta_r = 10^{-6}$. Izklopljena je detekcija numeričnih oscilacij.

Spodaj je izpis datoteke *TestnoVezje*, ki najprej vsebuje opis vezja, nastavitev parametrov, nato pa v bloku *.control* zagon analize *ssse*.

```
Testno vezje
.lib 'ts0u18/kay/top.kay' typical
.subckt recti com1 com2 in1 in2 out1 out2 bulk param:  w1=100u w2=100u cap=100p
mn_d11 in1  in1  1    bulk nch w=w1 l=0.5u m=1
mn_d12 2    2    in2  bulk nch w=w1 l=0.5u m=1
mn_d21 1    1    out1 bulk nch w=w2 l=0.5u m=1
mn_d22 out2 out2 2    bulk nch w=w2 l=0.5u m=1
c11 com1 1 cap
c12 com1 2 cap
c21 com2 out1 cap
c22 com2 out2 cap
.ends

.subckt 2_recti 2 5 6 bulk param:  w1=100u w2=100u w3=100u w4=100u cap=100p
x1 2 0 0 0 3 4 bulk recti param:  w1=w1 w2=w2 cap=cap
x2 2 0 3 4 5 6 bulk recti param:  w1=w3 w2=w4 cap=cap
.ends

vs inp inn dc 0 ac 1 sin 0 1.0 2.4g
rinp inp 2 r=25
rinn inn 5 r=25
xnrecti 2 5 6 6 2_recti param:  w1=10u w2=10u w3=8u w4=8u cap=100p
iload 5 6 dc=10u

Nastavitev parametrov:
.options itl2=1000
.options sssetol=1e-2
.options abstol=1e-4
.options reltol=1e-4

Analiza ssse:
.control
set xmumult=1
ssse v(inp,inn) 0 2p 0 6 history
.endc
.end
```

Datoteko poženemo v okolju NUTMEG programskega paketa SPICE OPUS, tako da vpišemo ime datoteke `TestnoVezje`. Izpis programa je prikazan spodaj:

```
Welcome to Program:
SpiceOpus (c), version:  2.24 Professional Revision:  30
Date built:  Nov 2 2006
Copyright (C) 1996-2006

SpiceOpus (c) 1 -> TestnoVezje
SpiceOpus (c) 2 -> setplot
        new      New plot
Current ssse5    Testno vezje (Steady State Analysis)
        ssse4    Testno vezje (Steady State Analysis)
        ssse3    Testno vezje (Steady State Analysis)
        ssse2    Testno vezje (Steady State Analysis)
        ssse1    Testno vezje (Steady State Analysis)
        const    Constant values (constants)
SpiceOpus (c) 3 ->
```

V Poglavju 5 bo uporabljena analiza *ssse*, kjer bo na realnih primerih vezij opisano testiranje metode za izračun stacionarnega stanja. Predstavljenih bo več tipov vezij, pri katerih so se pokazale prednosti metode v primerjavi z izračunom stacionarnega stanja z uporabo običajne tranzientne analize.

5

Testiranje metode na realnih primerih vezij

V tem poglavju bomo pokazali uspešnost metode za izračun stacionarnega stanja na realnih primerih vezij. Uspešnost se kaže v hitrejšem izračunu stacionarnega stanja v primerjavi z običajno tranzientno analizo, pri čemer ohranimo točnost izračuna in dosežemo enake rezultate oz. enak odziv kot pri direktni tranzientni analizi.

Najprej bodo predstavljena testna vezja, ki bodo uporabljena za prikaz uspešnosti analize za izračun stacionarnega stanja. Za ta vezja bo opravljena primerjava računske zahtevnosti izračuna stacionarenga stanja s tranzientno analizo in z analizo za izračun stacionarnega stanja. Primerjali bomo odziv vezja, dobljenega z tranzientno analizo in z analizo za izračun stacionarnega stanja, iz česar bo razvidna točnost izračuna analize stacionarnega stanja.

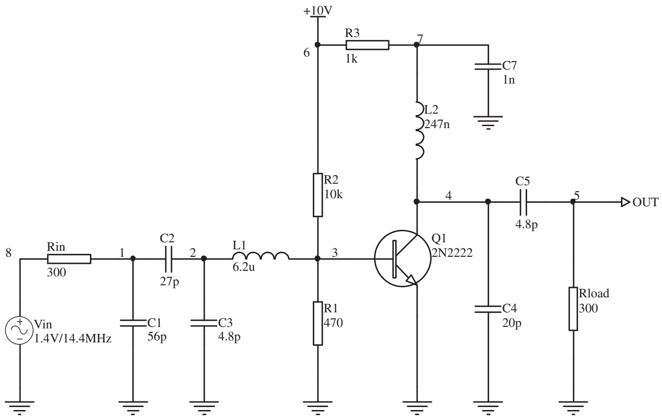
Zadnje podpoglavje bo pokazalo uporabnost analize za izračun stacionarnega stanja pri optimizaciji električnih vezij. Pri tem postopku je potrebno velikokrat ponoviti analize vezja. Če je potreben izračun stacionarnega stanja vezja, lahko z uporabo analize, opisane v doktorski disertaciji, zelo pospešimo optimizacijo električnega vezja.

5.1 Testna vezja

5.1.1 Vezje 1: Frekvenčni množilnik

Vezje na Sliki 5.1 predstavlja frekvenčni množilnik ([65], stran 124, slika 2.76).

Na vhod vezja v vozlišče 8 je priključen sinusni signalni generator frekvence 14,4 MHz in amplitude 1,4 V. Izhod frekvenčnega množilnika je v vozlišču 5 in vsebuje frekvenčne komponente štirikratnika vhodnega signala, torej frekvence 57,6 MHz. Vhodno impedanco vezja

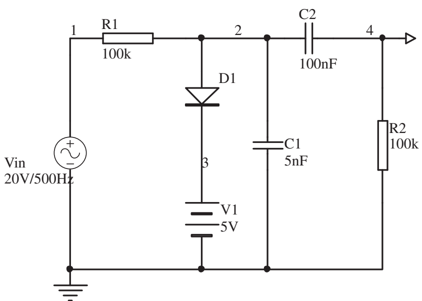


Slika 5.1. Frekvenčni množilnik

določa upor R_{in}. Na izhod vezja je priključeno breme oz. upor R_{load} z upornostjo 300 Ω. Tuljava L_2 določa lastnosti frekvenčnega množilnika. Njena vrednost je nastavljena na vrednost 247 nH, ki je izbrana tako, da skupaj z ostalim delom vezja ne duši četrte harmonske komponente vhodnega signala (57,6 MHz), ostale harmonske komponente pa izloči oz. reflektira nazaj v vezje frekvenčnega množilnika.

5.1.2 Vezje 2: Nelinearno usmerniško vezje RC

Slika 5.2 prikazuje usmerniško vezje ([66], slika 1).

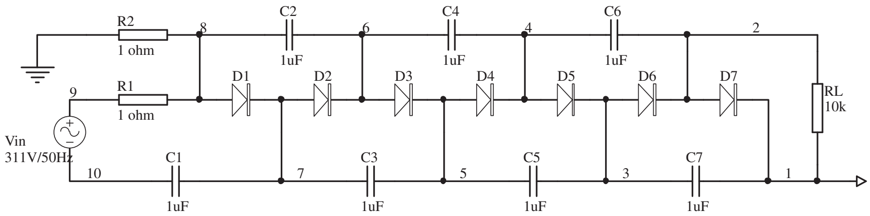


Slika 5.2. Nelinearno usmerniško vezje RC

Kondenzator C_2 se polni preko obeh uporov R_1 in R_2 s časovno konstanto $20\ ms$. Prazni se preko diode D_1 in upora R_2 s časovno konstanto $10\ ms$. Posledica tega je, da ob vključitvi signala V_{in} celotno vezje potrebuje od 100 do $200\ ms$, da prehodni pojavi izzvenijo in da dobimo periodičen odziv.

5.1.3 Vezje 3: Napetostni množilnik

Naslednje testno vezje je prikazano na Sliki 5.3 ([67], stran 9, slika 4) in predstavlja kaskadni napetostni množilnik.

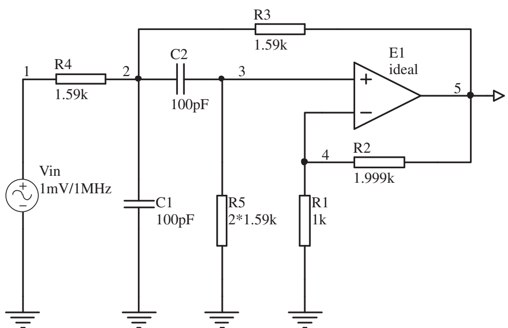


Slika 5.3. Napetostni množilnik

Napetostni sinusni vir V_{in} v pozitivnih polperiodah preko diod D_1, D_3, D_5 in D_7 polni kondenzatorje C_1, C_3, C_5 in C_7, v negativnih polperiodah pa preko diod D_2, D_4 in D_6 kondenzatorje C_2, C_4 in C_6. Zato na izhodu vezja v vozlišču 1 dobimo enosmerno napetost, ki je odvisna od števila kondenzatorjev v verigi kaskade in od amplitude vhodnega sinusnega vira V_{in}.

5.1.4 Vezje 4: Ozkopasovni filter

Vezje na Sliki 5.4 je ozkopasovni filter ([68], stran 16-30, slika 16-33) topologije Sallen-Key [69].

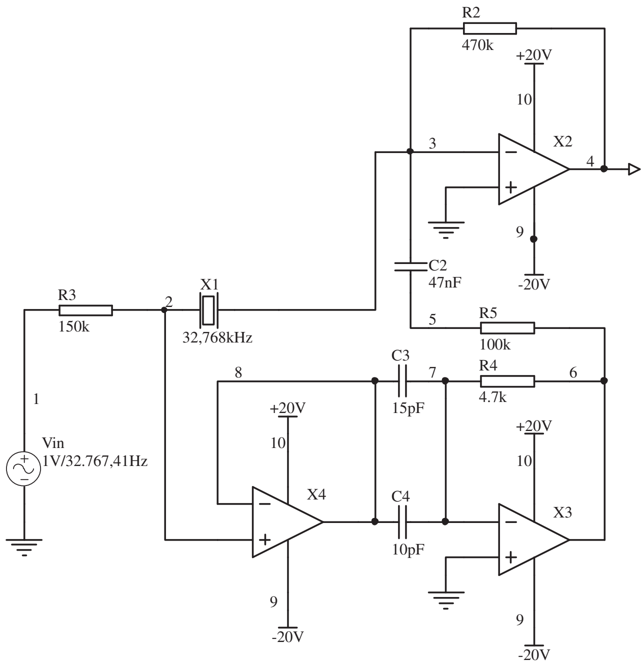


Slika 5.4. Ozkopasovni filter

Vrednosti elementov so izbrane tako, da je srednja frekvenca oz. frekvenca, ki jo filter prepušča, enaka 1 MHz, kvaliteta filtra pa $Q = 100$.

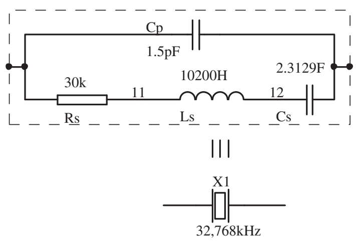
5.1.5 Vezje 5: Superozkopasovni filter s kvarcom

Superozkopasovni filter ([70], stran N79, slika 6) je prikazan na Sliki 5.5.



Slika 5.5. Superozkopasovni filter

Filter je realiziran s pomočjo kvarčnega kristala X_1, ki filtru določa veliko kvaliteto (približno 10.000). Na Sliki 5.6 je prikazano vezje, s katerim modeliramo kristal X_1.

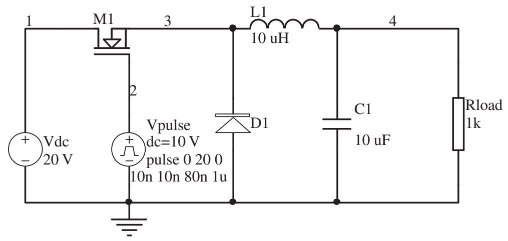


Slika 5.6. Modeliranje kristala X_1 na Sliki 5.5.

5.1.6 Vezje 6: Preklopni napajalnik

Na Sliki 5.7 je prikazan poenostavljen model preklopnega napajalnika (t. i. "Buck switching regulator", [71], stran 9).

Enosmerni napetostni vir je priključen preko tranzistorja M_1, ki ga krmili pulzni vir V_{pulse}. Ko je tranzistor odprt, se preko tuljave L_1 polni kondenzator C_1, na katerega je priključen bremenski upor $R_{load} = 1\ k\Omega$. Ko tranzistor M_1 ne prevaja, se tok skozi tuljavo L_1 zaključuje preko diode D_1. Izhodna enosmerna napetost na bremenu je odvisna od razmerja med trajanjem

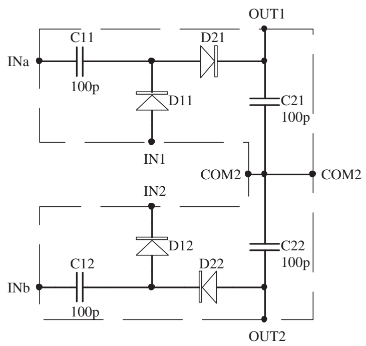


Slika 5.7. Preklopni napajalnik

visokega in nizkega nivoja (*duty cycle*) pulznega vira V_{pulse}. Vezje je poenostavljen model preklopnega napajalnika. V praksi tak preklopni napajalnik potrebuje še povratno vezavo, ki spreminja vrednost *duty cycle-a* pulznega vira V_{pulse} v odvisnosti od razlike izhodne napetosti in zahtevane izhodne napetosti.

5.1.7 Vezje 7: Greinacherjev usmernik

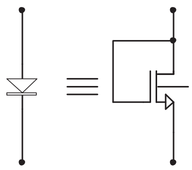
Naslednje testno vezje je usmerniško vezje, ki se med drugim uporablja za napajanje sistemov za radijsko identifikacijo (RFID - Radio Frequency Identification Systems). Na Sliki 5.8 je prikazana ena stopnja Greinacherjevega usmerniškega vezja [72].



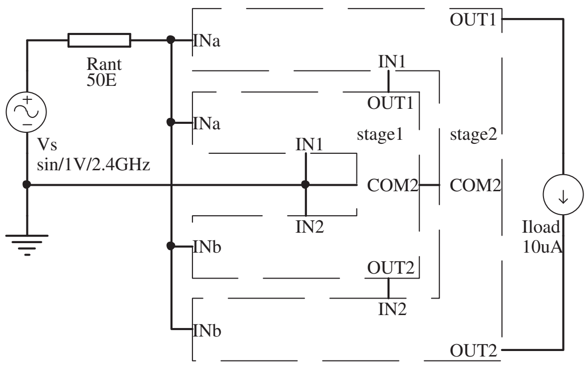
Slika 5.8. Ena stopnja Greinacherjevega usmernika

Vse diode so realizirane z uporabo n-kanalnih MOS tranzistorjev, kar prikazuje Slika 5.9. Uporabljena je bila 0,18 μm tehnologija TSMC. Dolžina vseh tranzistorjev obeh stopenj, s katerimi modeliramo diode, so enake 0,5 μm, širine tranzistorjev prve stopnje so enake 10 μm in druge stopnje 8 μm.

Za testno vezje smo uporabili dvostopenjsko Greinacherjevo usmerniško vezje. Slika 5.10 prikazuje medsebojno vezavo dveh stopenj, vhodni signalni vir V_s frekvence 2,4 GHz in amplitude 1 V, ki skupaj z uporom R_{ant} predstavlja anteno. Izhod vezja je obremenjen v tokovnim virom $I_{load} = 10\ \mu A$.



Slika 5.9. Modeliranje diode s tranzistorjem

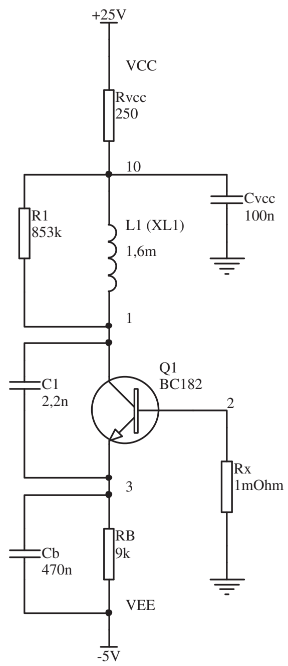


Slika 5.10. Testno vezje: dvostopenjski Greinacherjev usmernik

5.1.8 Vezje 8: Oscilator

Slika 5.11 predstavlja enostavno oscilatorsko vezje. Vezje je avtonomno, osnove frekvence signalov vezja v stacionarnem stanju ne poznamo vnaprej in je zato primerno pa prikaz delovanja metode za določevanje frekvence s pomočjo prehoda skozi nivo (Poglavje 3.3.2).

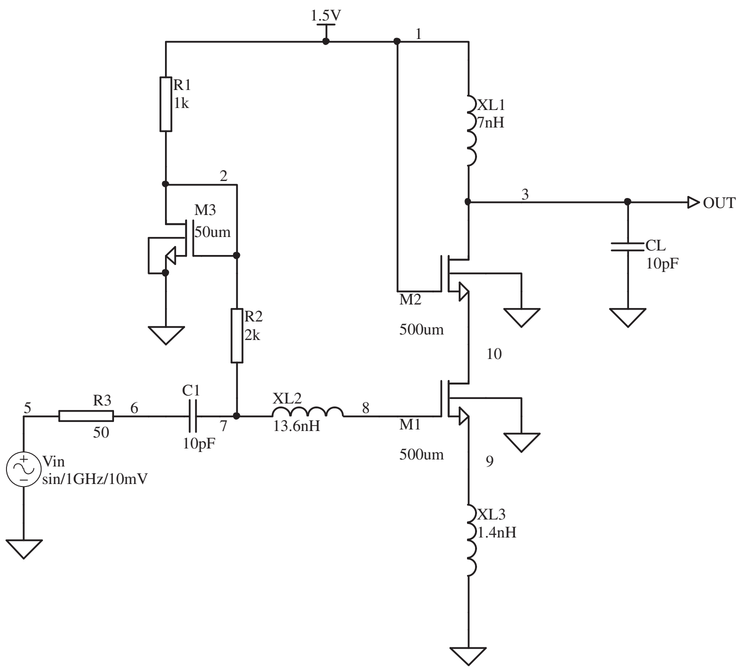
Glavni aktivni element v oscilatorju je tranzistor Q_1, ki je vezan kot ojačevalnik s skupno bazo. Izhodna sponka je kolektor, ki je priključen na nihajni krog. Ta je sestavljen iz tuljave L_1 in kondenzatorja C_1. Kondenzator C_b skupaj s C_1 tvori kapacitivni delilnik, ki vodi signal na emitor tranzistorja Q_1. Delilnik ima veliko razmerje zato, da je izkrmiljenje tranzistorja majhno. S tem dosežemo majhno harmonično popačenje izhodnega signala. Amplitudo izhodne napetosti spreminjamo z enosmernim emitorskim tokom oz. s spreminjanjem upornosti R_B.



Slika 5.11. Oscilator

5.1.9 Vezje 9: Nizkošumni ojačevalnik

Na Sliki 5.12 je shema nizkošumnega ojačevalnika ([76], stran 290 in 80).



Slika 5.12. Nizkošumni ojačevalnik

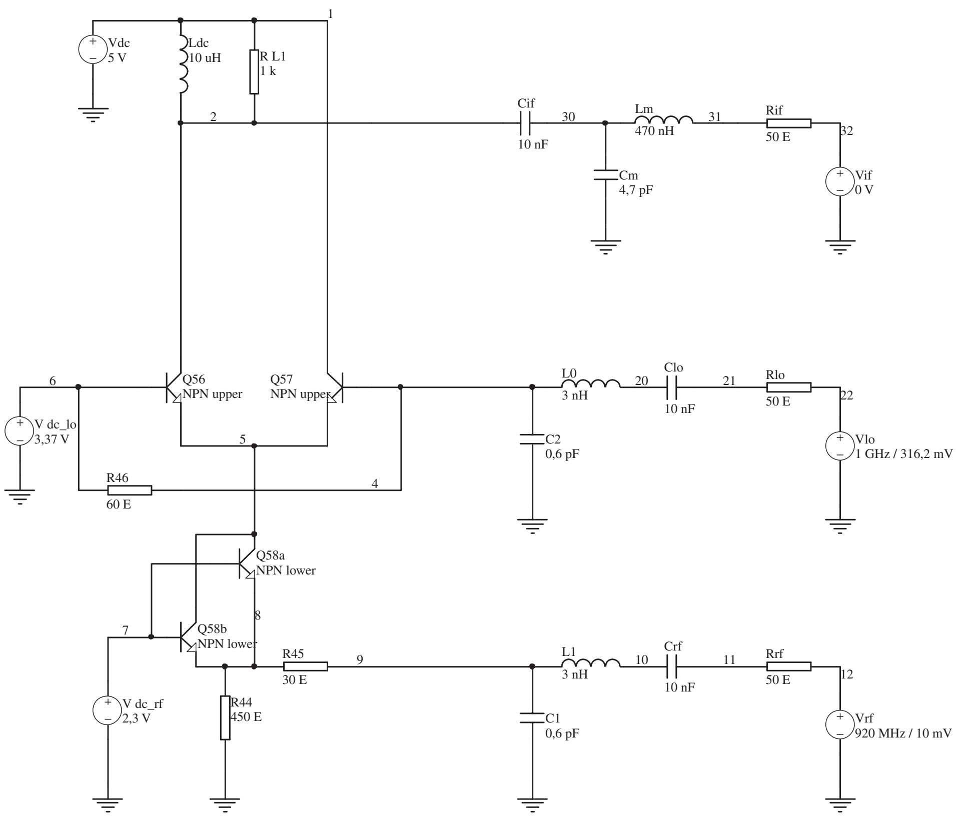
Kaskodni tranzistor M_2 preprečuje medsebojno vplivanje vhoda in izhoda in zmanjšuje učinke kapacitivnosti C_{gd} tranzistorja M_1. Skupna kapacitivnost vozlišča ponora tranzistorja M_2 je v resonanci s tuljavo XL_1. S tem povečamo ojačenje pri centralni frekvenci. Vhod in izhod nihata pri isti resonančni frekvenci, če pa želimo širši frekvenčni odziv, ti dve resonančni frekvenci razmaknemo.

Tranzistor M_3 predstavlja tokovno zrcalo s tranzistorjem M_1. Tok skozi tranzistor M_3 je odvisen od napajalne napetosti, upornosti R_1 in od V_{gs} tranzistorja M_3. Vrednost upora R_2 mora biti dovolj velika, da je njegov ekvivalenten šumni tok dovolj majhen, da ga lahko zanemarimo. Kondenzator C_1 blokira enosmerno napetost. Njegova vrednosti mora biti taka, da je reaktanca pri frekvenci signala v vezju zanemarljiva.

Vrednosti elementov na Sliki 5.12 so izbrane tako, da je vhodna upornost enaka $50\ \Omega$, mirovni tok je enak $5\ mA$. Uporabljena je 0,5 μm tehnologija.

5.1.10 Vezje 10: Mešalnik

Vezje na Sliki 5.13 predstavlja mešalnik.



Slika 5.13. Mešalnik

Vhodni signal oz. napetostni vir V_{rf} je preko vhodnega upora R_{rf} in vhodnega filtra priključen na tokovni vir, ki ga sestavljata tranzistorja Q_{58a} in Q_{58b}, vir $V_{dc\ rf}$ in upor R_{44}. Z opisano vezavo dosežemo modulacijo vrednosti tokovnega vira v odvisnosti od vhodnega signala V_{rf}.

Podobna vezava je pri signalu lokalnega oscilatorja V_{lo}, le da je signal priključen na bazo tranzistorja Q_{57}.

Izhodni medfrekvenčni signal iz vozlišča 2 vodimo preko izhodnega filtra, kjer ga na njegovem izhodu v vozlišču 31 obremenimo z bremenom R_{if}.

5.2 Izračun stacionarnega stanja

Stacionarno stanje je bilo za vsa vezja iz Poglavja 5.1 najprej izračunano z direktno tranzientno analizo (neprekinjena tranzientna analiza, dokler prehodni pojavi ne izzvenijo), nato pa še z uporabo ekstrapolacijske metode (algoritem epsilon), s katero pospešimo tranzientno analizo (Poglavje 3.2).

Pri izračunu stacionarnega stanja z direktno tranzientno analizo smo za testna vezja iz začetnega odziva (časi od $t = 0$ naprej) najprej ocenili časovne konstante vezij τ. Časovno konstanto smo določili na enega od dveh načinov. Po prvem načinu je časovna konstanta določena kot presečišče med tangento na krivuljo odziva pri času $t = 0$ in končno vrednostjo odziva pri dovolj velikem času. Pri drugem načinu je pa časovna konstanta enaka času, pri katerem odziv doseže 63,2 % ($1 - e^{-1}$) končnega odziva; oz. polovica časa, pri katerem odziv doseže 86,4 % ($1 - e^{-2}$) končnega odziva, itd.

Nato smo na podlagi časovnih konstant in želenih relativnih natančnosti δ_r z uporabo enačbe (2.13) določili minimalni čas t direktnih tranzientnih analiz, ki so potrebne za dosego zahtevanih natančnosti stacionarnih odzivov. Kot primer natančnosti naj navedemo natančnost $\delta_r = 10^{-4}$. Za ta primer moramo po (2.13) vezje analizirati 9,2 časovni konstanti vezja.

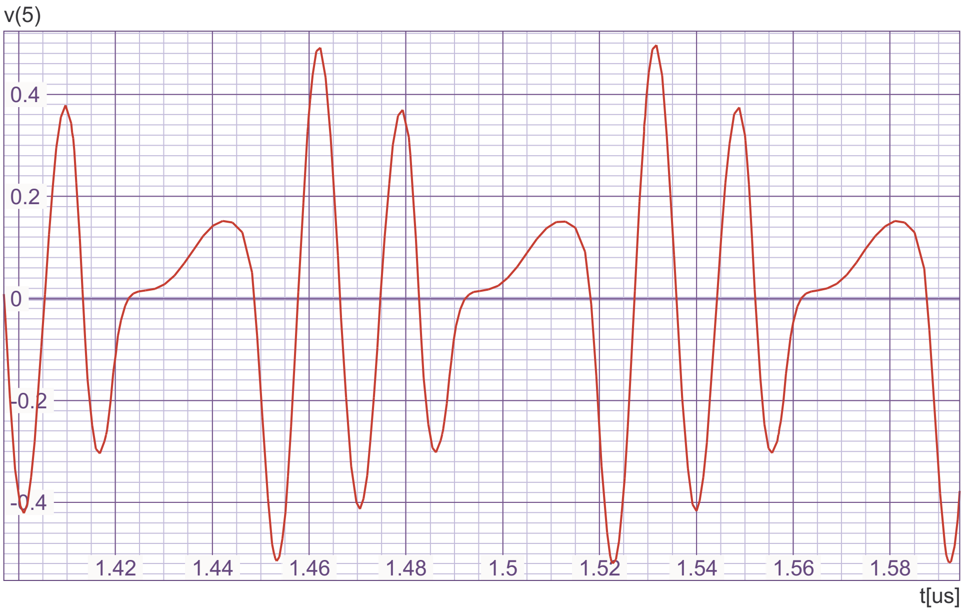
Za izračun stacionarnega stanja z uporabo ekstrapolacijske metode je bila izbrana enaka natančnost kot pri direktni tranzientni analizi. S tem dosežemo primerljivost časov, ki so potrebni, da dosežemo stacionarno stanje pri obeh metodah. V naslednjem poglavju je tako prikazana primerjava računske zahtevnosti za izbrana testna vezja pri obeh metodah.

Stacionarno stanje, ki je bilo izračunano z direktno tranzientno analizo, je bilo izvedeno na računalniku AMD Athlon XP 2500+ (1,83 GHz) s 512 MB pomnilnika. Pri vsakem testnem vezju je naveden čas izračuna stacionarnega stanja.

Frekvenčni množilnik

Opazovani izhodni signal: $v(5)$
Osnovna frekvenca: 14,4 MHz
Izbrana natančnost: $\delta_r = 5 \cdot 10^{-3}$, $\delta_a = 5 \cdot 10^{-5}$

Tranzientna analiza



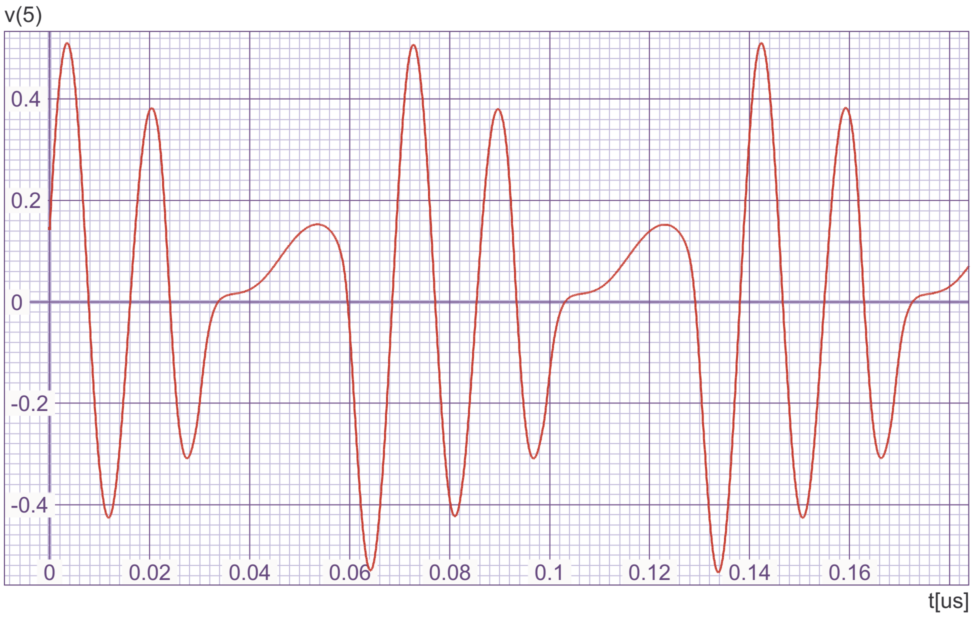
Slika 5.14. Odziv frekvenčnega množilnika pri času 5τ

Stacionarni odziv dosežen pri: $t > 10\ \mu s$
Zgornja vrednost ovojnice signala pri $t = 10\ \mu s$ je 0,4962 V.
Določitev časovne konstante na podlagi amplitude pri 5τ:
0,4962 V $\cdot(1 - e^{-5})$ =0,4929 V
Amplituda 0,4929 V je dosežena pri t=1,53 μs (Slika 5.14)
5τ =1,53 μs
τ =0,306 μs
$\delta_r = 5 \cdot 10^{-3}$
Čas simulacije za zahtevano relativno natančnost:
$t >$1,621 μs
Število period: 1,621 $\mu s\cdot$ 14,4 MHz $\approx$ 23 period
Čas izračuna stacionarnega stanja: 0,2 s.

Ekstrapolacijski algoritem

Vrednosti parametrov:

t_{del_1} =3/14,4 MHz	(3 periode za 1. iteracijo ekstrapolacijskega algoritma)
t_{del_2} =0	(za vse naslednje iteracije ekstrapolacijskega algoritma)
m_1 =4	(vrednost parametra m za 1. iteracijo ekstrapolacijskega algoritma)
m_2 =6	(vrednost parametra m za 2. iteracijo ekstrapolacijskega algoritma)



Slika 5.15. Odziv frekvenčnega množilnika z uporabo ekstrapolacijskega algoritma

Število iteracij ekstrapolacijskega algoritma: 2

Skupaj period pri izračunu stacionarnega stanja z ekstrapolacijsko metodo: 13 period

Nelinearno usmerniško vezje RC

Opazovani izhodni signal: $v(4)$

Osnovna frekvenca: 500 Hz

Izbrana natančnost: $\delta_r = 5 \cdot 10^{-4}$, $\quad \delta_a = 5 \cdot 10^{-6}$

Tranzientna analiza

Stacionarni odziv dosežen pri: $t > 1\ s$

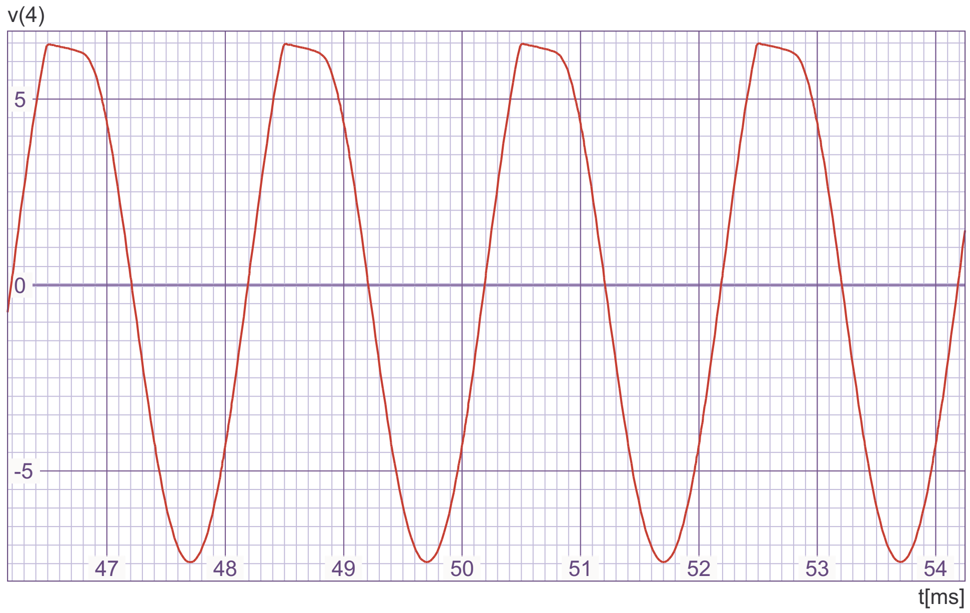
Zgornja vrednost ovojnice signala pri $t = 1\ s$ je 6,537 V.

Določitev časovne konstante na podlagi amplitude pri 5τ:

6,537 $V \cdot (1 - e^{-5})$ =6,493 V

Amplituda 6,493 V je dosežena pri t=52,5 ms (Slika 5.16)

5τ =52,5 ms



Slika 5.16. Odziv nelinearnega usmerniškega vezja RC pri času 5τ

$\underline{\underline{\tau = 10{,}5\ ms}}$

$\delta_r = 5 \cdot 10^{-4}$

Čas simulacije za zahtevano relativno natančnost:

$\underline{\underline{t > 79{,}81\ ms}}$

Število period: 79,81 $ms \cdot$ 500 $Hz \approx$ $\underline{\underline{\text{40 period}}}$

Čas izračuna stacionarnega stanja: 0,1 s.

Ekstrapolacijski algoritem

Vrednosti parametrov:

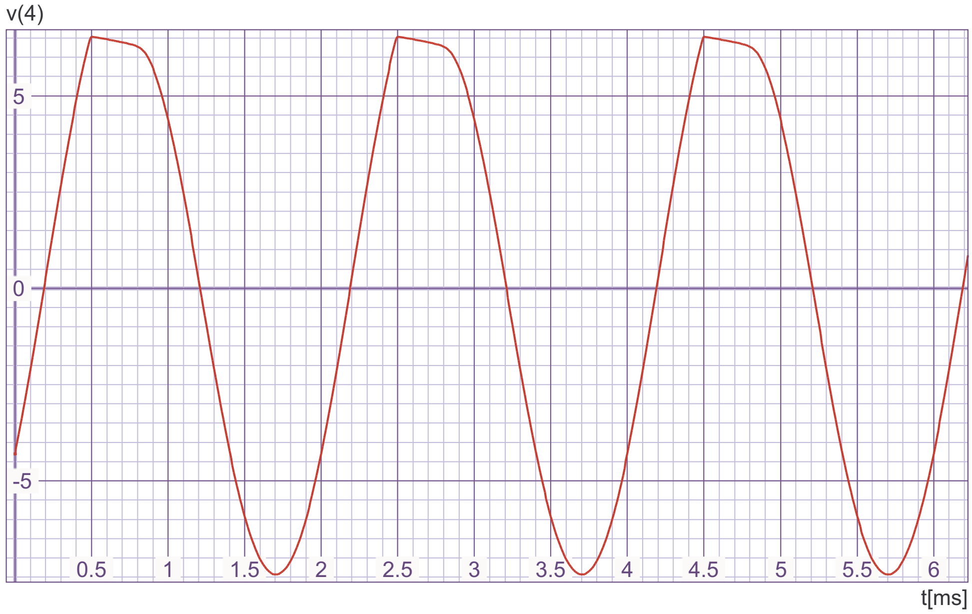
t_{del_1} =0,1/500 Hz (0,1 periode za 1. iteracijo ekstrapolacijskega algoritma)

t_{del_2} =0 (za vse naslednje iteracije ekstrapolacijskega algoritma)

m =4 (vrednost parametra m za vse iteracije ekstrapolacijskega algoritma)

Število iteracij ekstrapolacijskega algoritma: 4

Skupaj period pri izračunu stacionarnega stanja z ekstrapolacijsko metodo: $\underline{\underline{\text{16,1 periode}}}$



Slika 5.17. Odziv nelinearnega usmerniškega vezja RC z uporabo ekstrapolacijskega algoritma

Napetostni množilnik

Opazovani izhodni signal: $v(1)$

Osnovna frekvenca: 50 Hz

Izbrana natančnost: $\delta_r = 5 \cdot 10^{-6}$, $\delta_a = 5 \cdot 10^{-5}$

Tranzientna analiza

Stacionarni odziv dosežen pri: $t > 10\ s$

Zgornja vrednost ovojnice signala pri $t = 10\ s$ je 1256 V.

Določitev časovne konstante na podlagi amplitude pri 5τ:

1256 $V \cdot (1 - e^{-5})$ =1248 V

Amplituda 1248 V je dosežena pri t=734 ms (Slika 5.18)

5τ =734 ms

τ =146,8 ms

$\delta_r = 5 \cdot 10^{-6}$

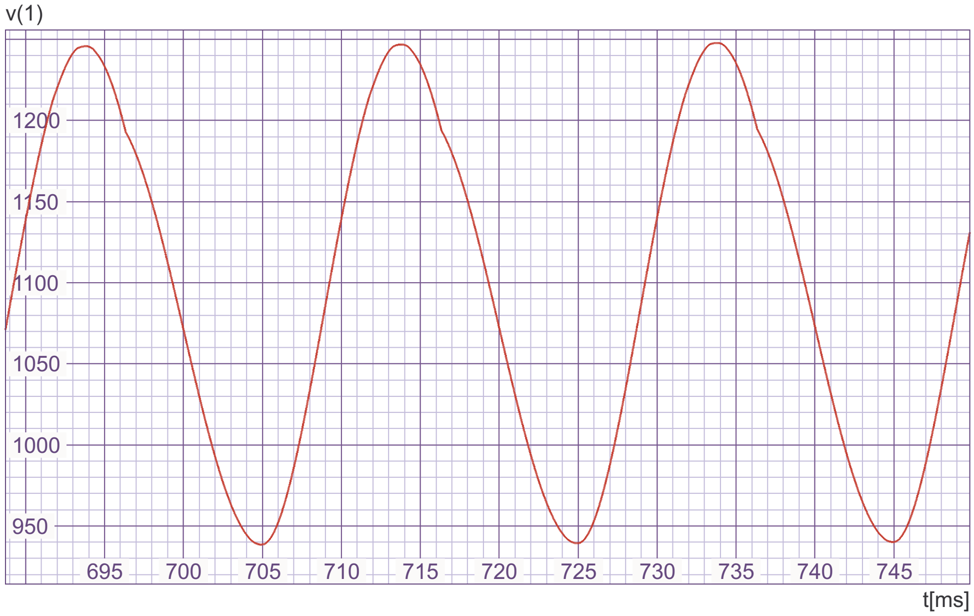
Čas simulacije za zahtevano relativno natančnost:

t >1,792 s

Število period: 1,792 $s\cdot$ 50 $Hz \approx$ 90 period

Čas izračuna stacionarnega stanja: 150 s.

Ekstrapolacijski algoritem



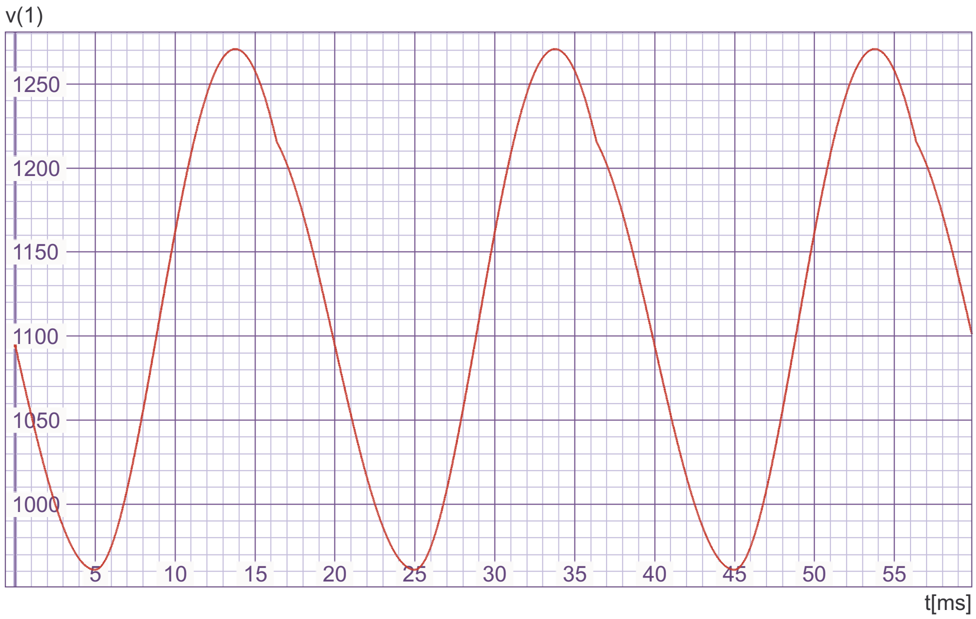
Slika 5.18. Odziv napetostnega množilnika pri času 5τ

Vrednosti parametrov:

t_{del_1} =0,1/50 Hz (0,1 periode za 1. iteracijo ekstrapolacijskega algoritma)

t_{del_2} =0 (za vse naslednje iteracije ekstrapolacijskega algoritma)

m =6 (vrednost parametra m za vse iteracije ekstrapolacijskega algoritma)



Slika 5.19. Odziv napetostnega množilnika z uporabo ekstrapolacijskega algoritma

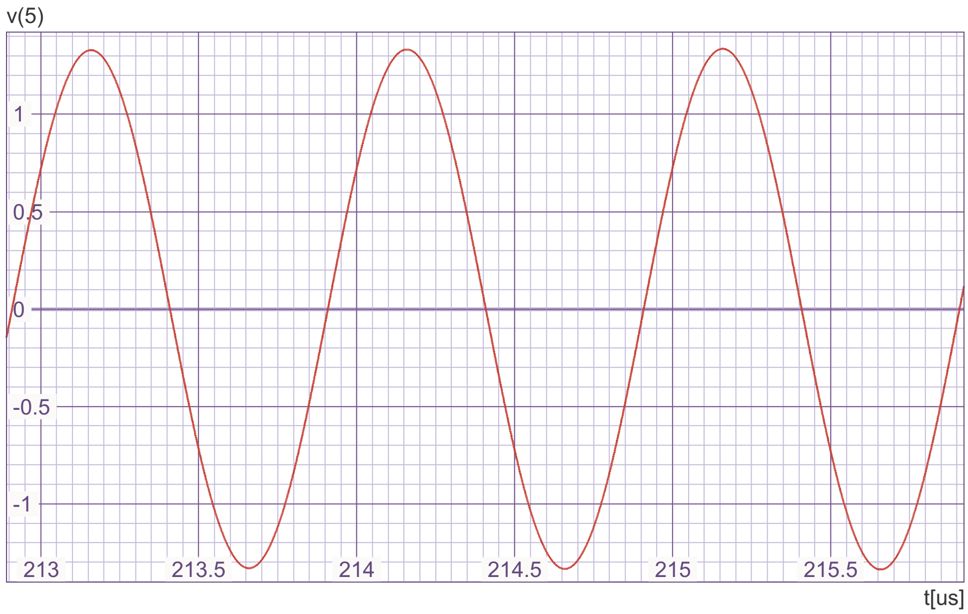
Število iteracij ekstrapolacijskega algoritma: 5

Skupaj period pri izračunu stacionarnega stanja z ekstrapolacijsko metodo: 30,1 periode

Ozkopasovni filter

Opazovani izhodni signal: $v(5)$
Osnovna frekvenca: 1 MHz
Izbrana natančnost: $\delta_r = 10^{-5}$, $\delta_a = 10^{-5}$

Tranzientna analiza



Slika 5.20. Odziv ozkopasovnega filtra pri času 5τ

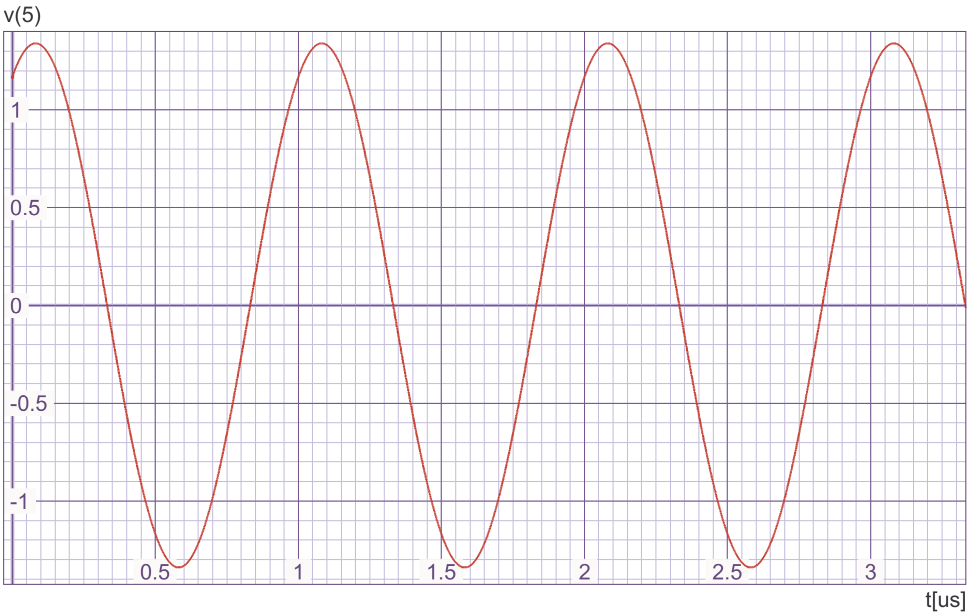
Stacionarni odziv dosežen pri: $t >$10 μs
Zgornja vrednost ovojnice signala pri $t =$10 μs je 1,344 V.
Določitev časovne konstante na podlagi amplitude pri 5τ:
1,344 $V \cdot (1 - e^{-5}) =$1,335 V
Amplituda 1,335 V je dosežena pri $t =$215,2 μs (Slika 5.20)
$5\tau =$215,2 μs
$\tau =$43,04 μs
$\delta_r = 10^{-5}$
Čas simulacije za zahtevano relativno natančnost:
$t >$496 μs
Število period: 496 $\mu s \cdot$ 1 $MHz \approx$ 496 period
Čas izračuna stacionarnega stanja: 48 s.

Ekstrapolacijski algoritem

Vrednosti parametrov:

t_{del} =0 (za vse iteracije ekstrapolacijskega algoritma)

m =6 (vrednost parametra m za vse iteracije ekstrapolacijskega algoritma)



Slika 5.21. Odziv ozkopasovnega filtra z uporabo ekstrapolacijskega algoritma

Število iteracij ekstrapolacijskega algoritma: 4

Skupaj period pri izračunu stacionarnega stanja z ekstrapolacijsko metodo: 24 period

Superozkopasovni filter s kvarcom

Opazovani izhodni signal: $v(4)$

Osnovna frekvenca: 32,76741 kHz

Izbrana natančnost: $\delta_r = 5 \cdot 10^{-4}$, $\delta_a = 5 \cdot 10^{-4}$

Tranzientna analiza

Stacionarni odziv dosežen pri: $t > 2\ s$

Zgornja vrednost ovojnice signala pri t =2 s je 3,78 V.

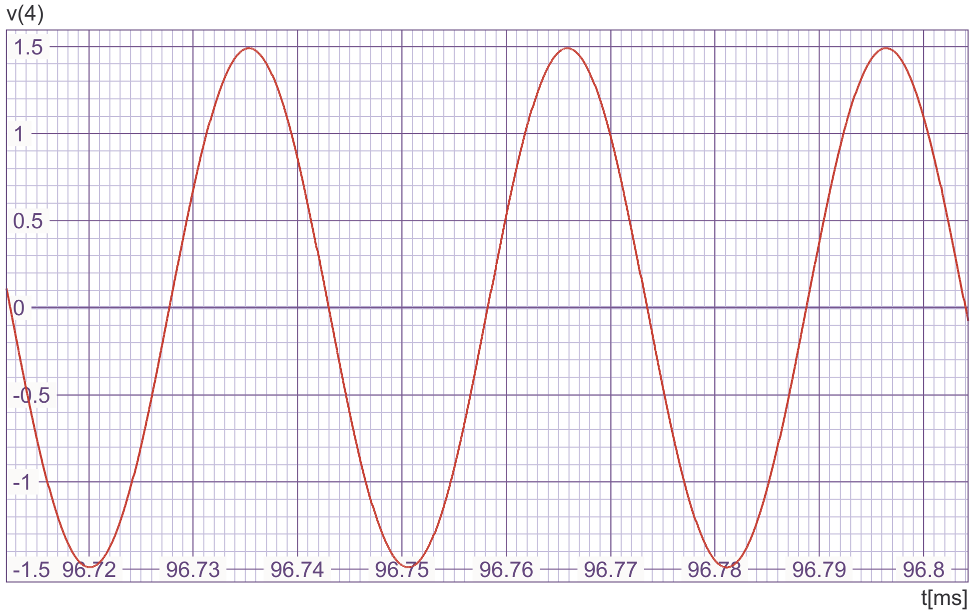
Določitev časovne konstante na podlagi amplitude pri 0,5τ:

3,78 $V \cdot (1 - e^{-0,5})$ =1,49 V

Amplituda 1,49 V je dosežena pri t =96,8 ms (Slika 5.22)

0,5τ =96,8 ms

τ =194 ms



Slika 5.22. Odziv superozkopasovnega filtra pri času 0,5τ

$\delta_r = 5 \cdot 10^{-4}$

Čas simulacije za zahtevano relativno natančnost:

$t >$1475 ms

Število period: 1475 $ms\cdot$ 32,76741 $kHz \approx$ 48.332 period

Čas izračuna stacionarnega stanja: 1110 s.

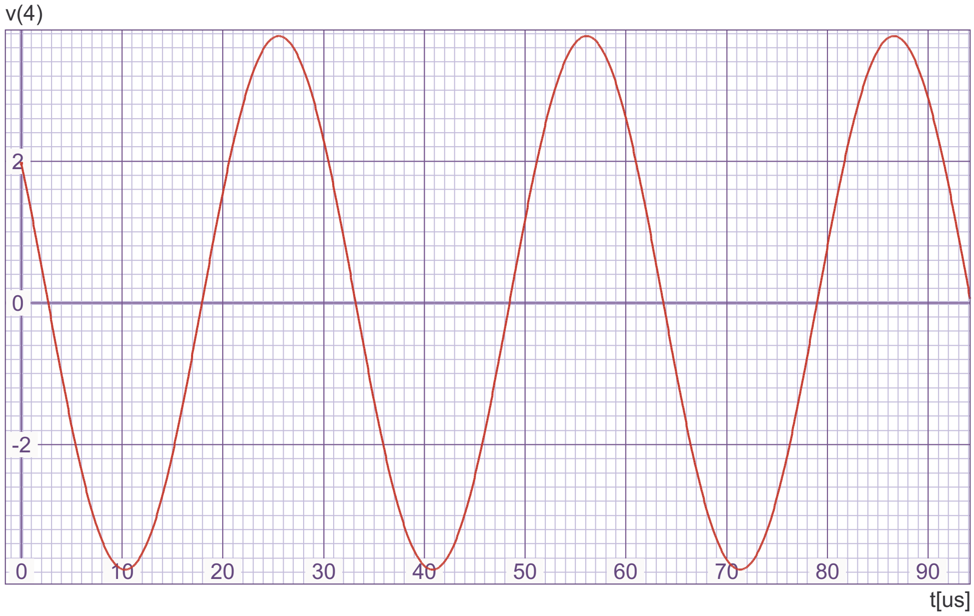
Ekstrapolacijski algoritem

Vrednosti parametrov:

t_{del} =10/32,76741 kHz	(10 period za vse iteracije ekstrapolacijskega algoritma)
m_1 =4	(vrednost parametra m za 1. iteracijo ekstrapolacijskega algoritma)
m_2 =8	(vrednost parametra m za 2. iteracijo ekstrapolacijskega algoritma)
m_3 =12	(vrednost parametra m za naslednje iteracije ekstrapolacijskega algoritma)

Število iteracij ekstrapolacijskega algoritma: 3

Skupaj period pri izračunu stacionarnega stanja z ekstrapolacijsko metodo: 54 period



Slika 5.23. Odziv superozkopasovnega filtra z uporabo ekstrapolacijskega algoritma

Preklopni napajalnik

Opazovani izhodni signal: $v(4)$

Osnovna frekvenca: 1 MHz

Izbrana natančnost: $\delta_r = 10^{-7}$, $\delta_a = 10^{-6}$

Tranzientna analiza

Stacionarni odziv dosežen pri: $t >$100 ms

Zgornja vrednost ovojnice signala pri $t =$100 ms je 6,7848 V.

Določitev časovne konstante na podlagi amplitude pri 9τ:

6,7848 $V \cdot (1 - e^{-9}) =$6,7840 V

Amplituda 6,7840 V je dosežena pri $t =$8,53 ms (Slika 5.24)

$9\tau =$8,53 ms

$\underline{\underline{\tau = 0{,}948\ ms}}$

$\delta_r = 10^{-7}$

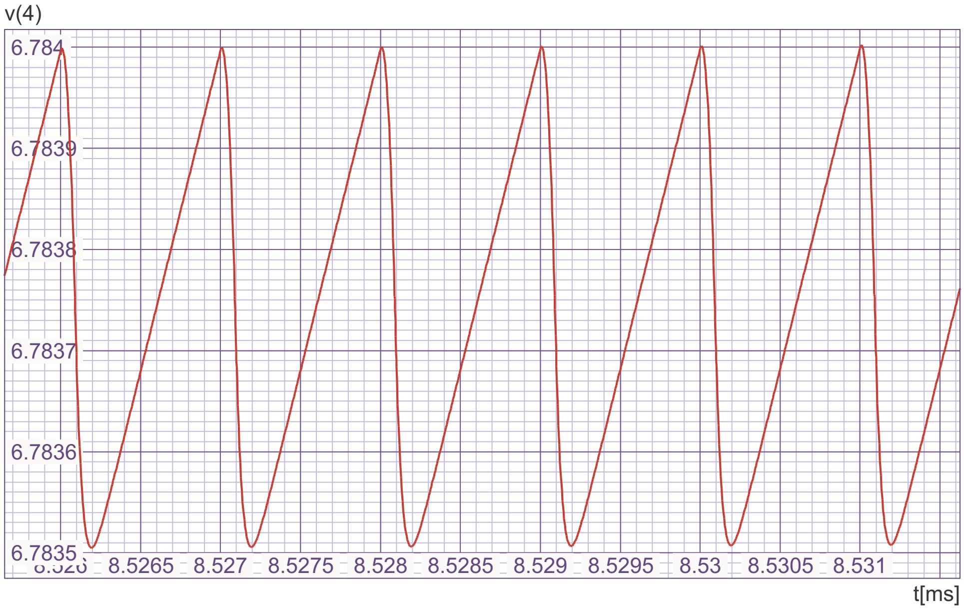
Čas simulacije za zahtevano relativno natančnost:

$\underline{\underline{t > 15{,}3\ ms}}$

Število period: 15,3 $ms\cdot$ 1 $MHz \approx$ $\underline{\underline{\text{15.300 period}}}$

Čas izračuna stacionarnega stanja: 152 s.

Ekstrapolacijski algoritem

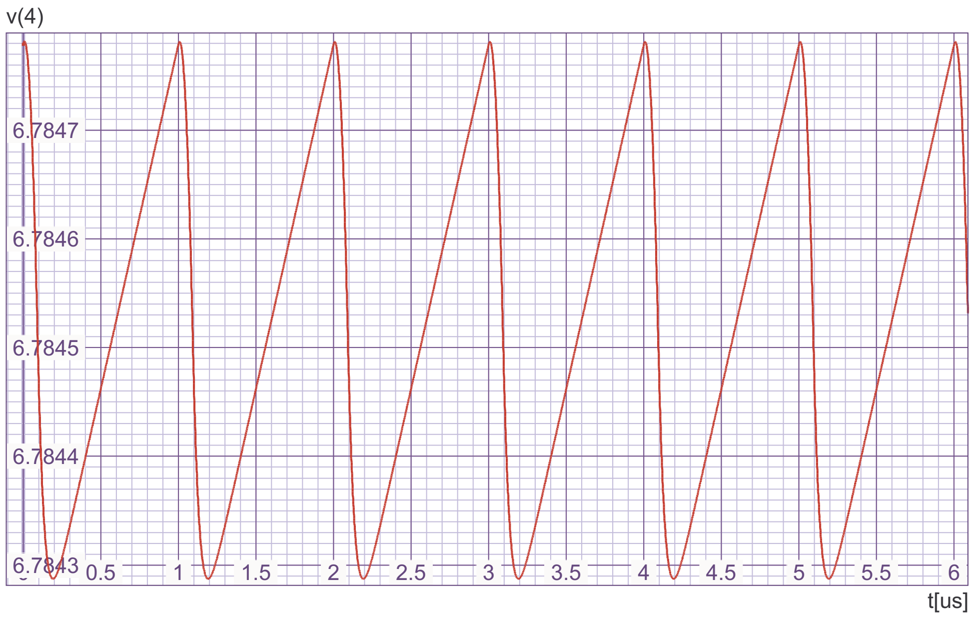


Slika 5.24. Odziv preklopnega napajalnika pri času 9τ

Vrednosti parametrov:

t_{del} =0 (za vse iteracije ekstrapolacijskega algoritma)

m =6 (vrednost parametra m za vse iteracije ekstrapolacijskega algoritma)



Slika 5.25. Odziv preklopnega napajalnika z uporabo ekstrapolacijskega algoritma

Število iteracij ekstrapolacijskega algoritma: 5

Skupaj period pri izračunu stacionarnega stanja z ekstrapolacijsko metodo: 30 period

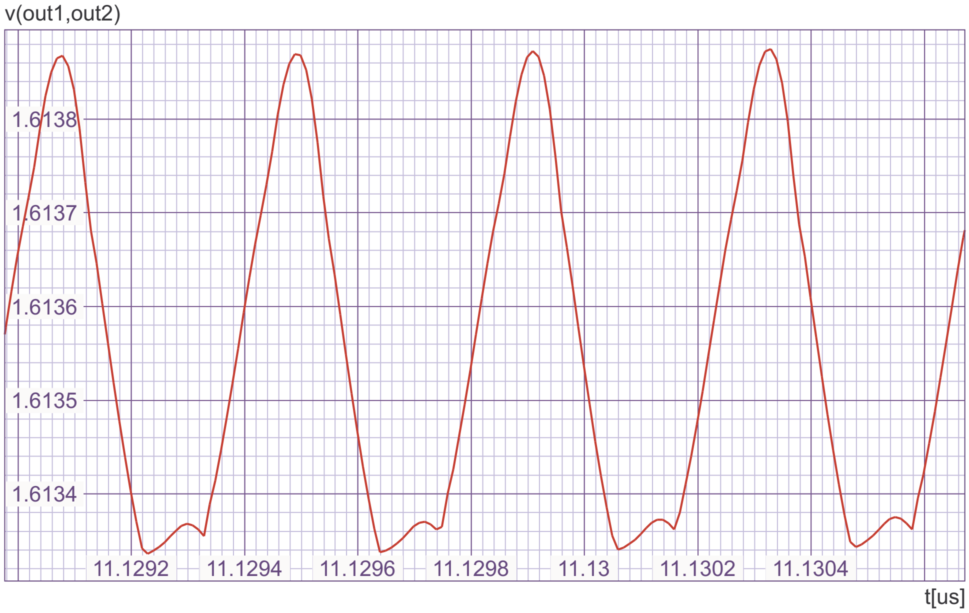
Greinacherjev usmernik

Opazovani izhodni signal: $v(out1, out2)$

Osnovna frekvenca: 2,4 GHz

Izbrana natančnost: $\delta_r = 10^{-6}$, $\delta_a = 10^{-6}$

Tranzientna analiza



Slika 5.26. Odziv Greinacherjevega usmernika pri času 5τ

Stacionarni odziv dosežen pri: $t >$100 μs

Zgornja vrednost ovojnice signala pri $t =$100 μs je 1,6248 V.

Določitev časovne konstante na podlagi amplitude pri 5τ:

1,6248 $V \cdot (1 - e^{-5}) =$1,6139 V

Amplituda 1,6139 V je dosežena pri $t =$11,13 μs (Slika 5.26)

$5\tau =$11,13 μs

$\tau =$2,226 μs

$\delta_r = 10^{-6}$

Čas simulacije za zahtevano relativno natančnost:

$t >$30,75 μs

Število period: 30,75 $\mu s \cdot$ 2,4 $GHz \approx$ 73.800 period

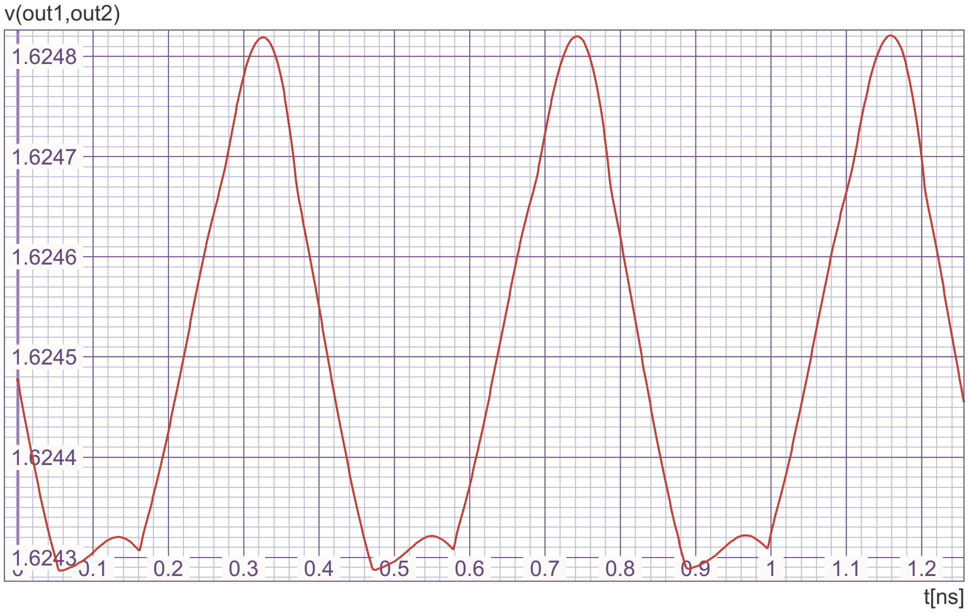
Čas izračuna stacionarnega stanja: 460 s.

Ekstrapolacijski algoritem

Vrednosti parametrov:

t_{del} =0 (za vse iteracije ekstrapolacijskega algoritma)

m =6 (vrednost parametra m za vse iteracije ekstrapolacijskega algoritma)



Slika 5.27. Odziv Greinacherjevega usmernika z uporabo ekstrapolacijskega algoritma

Število iteracij ekstrapolacijskega algoritma: 52

Skupaj period pri izračunu stacionarnega stanja z ekstrapolacijsko metodo: 312 period

Oscilator

Opazovani izhodni signal: $v(1)$

Osnovna frekvenca: 84,5 kHz (končna frekvenca pri času 2 ms)

Izbrana natančnost: $\delta_r = 10^{-4}$, $\delta_a = 10^{-4}$

Tranzientna analiza

Stacionarni odziv dosežen pri: t >2 ms

Zgornja vrednost ovojnice signala pri t =2 ms je 55,4 V.

Določitev časovne konstante na podlagi amplitude pri 3τ:

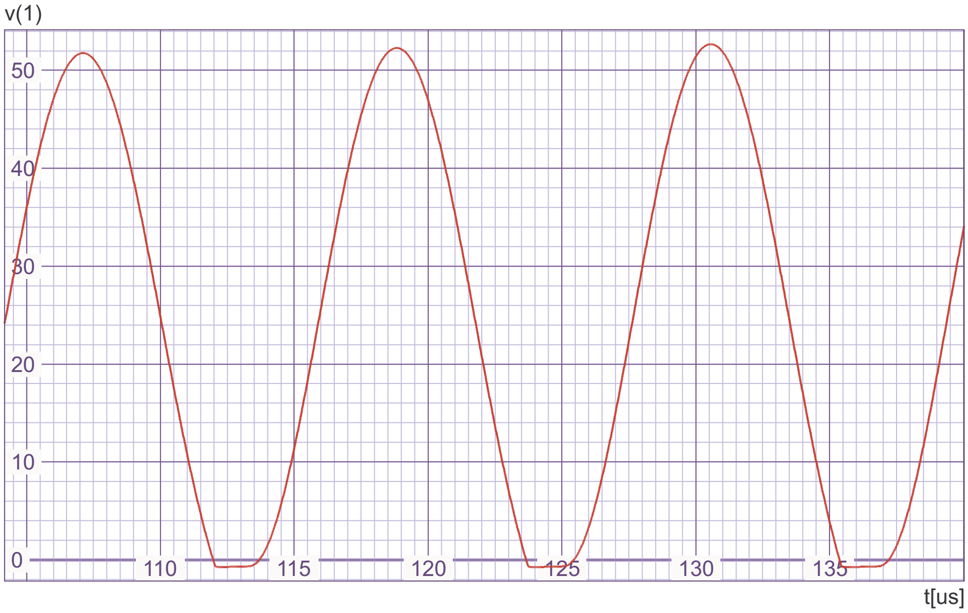
55,4 $V \cdot (1 - e^{-3})$ =52,6 V

Amplituda 52,6 V je dosežena pri t =130,5 μs (Slika 5.28)

3τ =130,5 μs

τ =43,5 μs

$\delta_r = 10^{-4}$



Slika 5.28. Odziv oscilatorja pri času 3τ

Čas simulacije za zahtevano relativno natančnost:

$t >$400 μs

Število period: 400 $\mu s\cdot$ 84,5 $kHz \approx$ 34 period

Čas izračuna stacionarnega stanja: 1,6 s.

Ekstrapolacijski algoritem

Vrednosti parametrov:

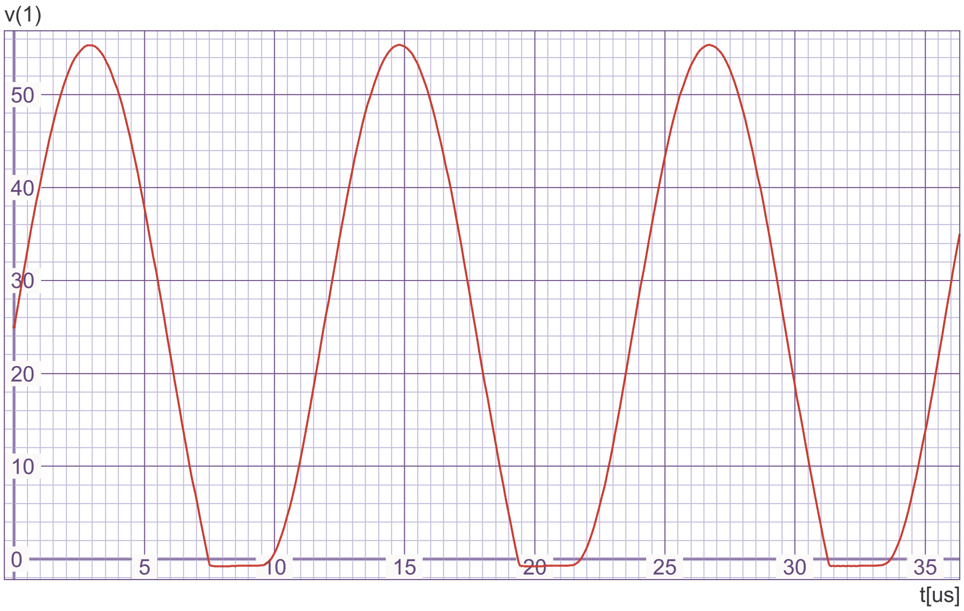
$t_{del} =$0 (za vse iteracije ekstrapolacijskega algoritma)

$m =$4 (vrednost parametra m za vse iteracije ekstrapolacijskega algoritma)

$L =$25 V (vrednost nivoja pri metodi določevanja frekvence)

Število iteracij ekstrapolacijskega algoritma: 4

Skupaj period pri izračunu stacionarnega stanja z ekstrapolacijsko metodo: 16 period



Slika 5.29. Odziv oscilatorja z uporabo ekstrapolacijskega algoritma

Nizkošumni ojačevalnik

Opazovani izhodni signal: $v(3)$

Osnovna frekvenca: 1 GHz

Izbrana natančnost: $\delta_r = 10^{-4}$, $\delta_a = 10^{-4}$

Tranzientna analiza

Stacionarni odziv dosežen pri: $t >$100 ns

Zgornja vrednost ovojnice signala pri $t =$100 ns je 1,569 V.

Določitev časovne konstante na podlagi amplitude pri 6τ:

1,569 $V \cdot (1 - e^{-6}) =$1,565 V

Amplituda 1,565 V je dosežena pri $t =$20,1 ns (Slika 5.30)

$6\tau =$20,1 ns

$\tau =$3,35 ns

$\delta_r = 10^{-4}$

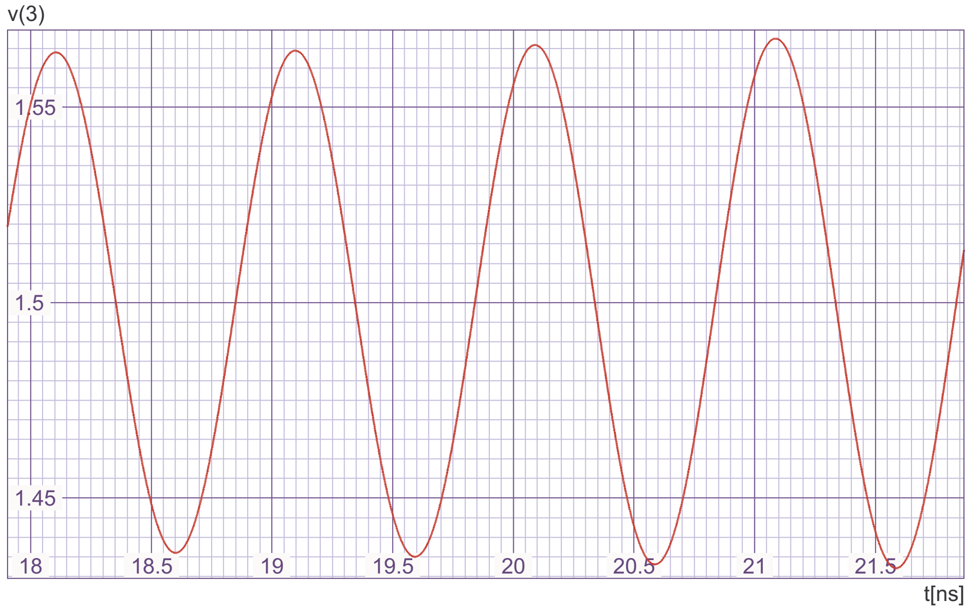
Čas simulacije za zahtevano relativno natančnost:

$t >$30,9 ns

Število period: 30,9 $ns \cdot$ 1 $GHz \approx$ 31 period

Čas izračuna stacionarnega stanja: 0,4 s.

Ekstrapolacijski algoritem

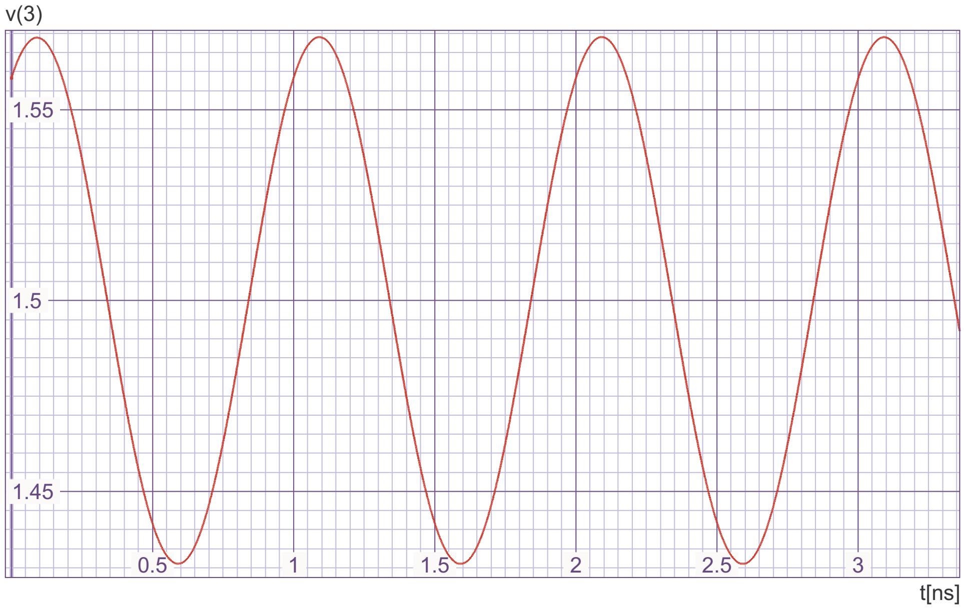


Slika 5.30. Odziv nizkošumnega ojačevalnika pri času 6τ

Vrednosti parametrov:

t_{del} =0 (za vse iteracije ekstrapolacijskega algoritma)

m =2 (vrednost parametra m za vse iteracije ekstrapolacijskega algoritma)



Slika 5.31. Odziv nizkošumnega ojačevalnika z uporabo ekstrapolacijskega algoritma

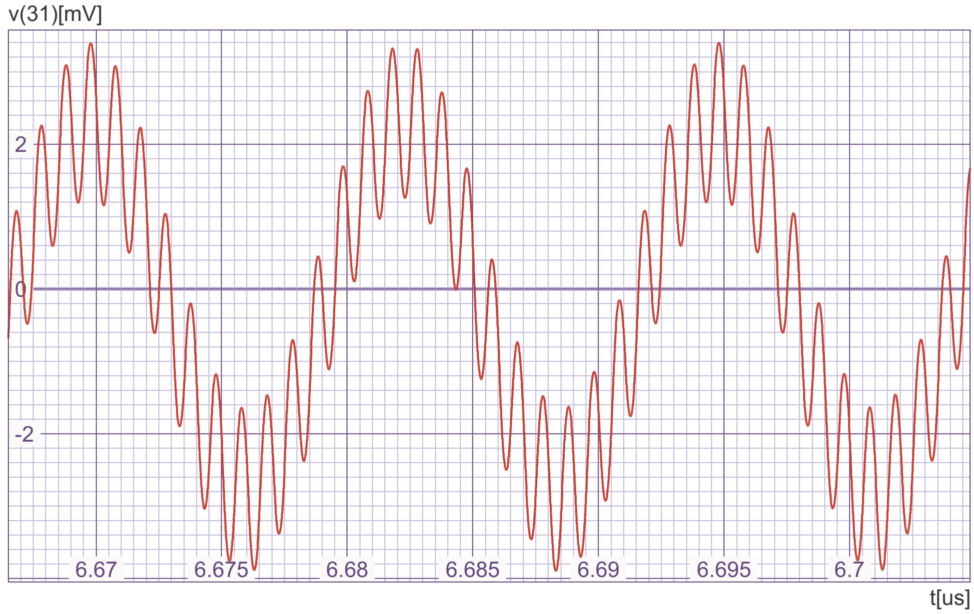
Število iteracij ekstrapolacijskega algoritma: 14

Skupaj period pri izračunu stacionarnega stanja z ekstrapolacijsko metodo: 28 period

Mešalnik

Opazovani izhodni signal: $v(31)$
Osnovna frekvenca: 80 MHz
Izbrana natančnost: $\delta_r = 10^{-6}$, δ_a =2,1 $\cdot 10^{-6}$

<u>Tranzientna analiza</u>



Slika 5.32. Odziv mešalnika pri času 3τ

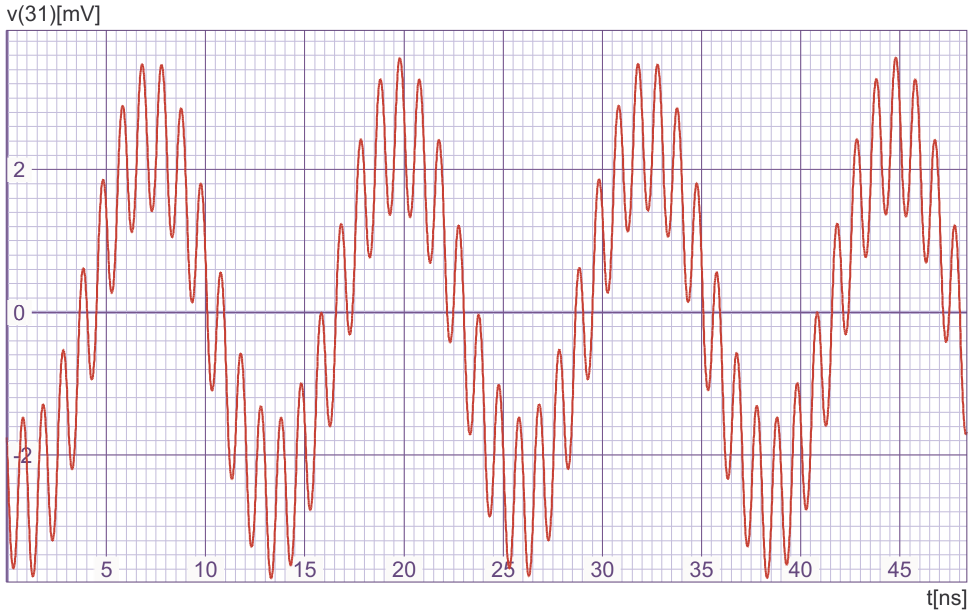
Stacionarni odziv dosežen pri: t >20 μs
Zgornja vrednost ovojnice signala pri t =20 μs je 3,58 mV.
Določitev časovne konstante na podlagi amplitude pri 3τ:
3,58 $mV \cdot (1 - e^{-3})$ =3,40 mV
Amplituda 3,40 mV je dosežena pri t =6,695 μs (Slika 5.32)
3τ =6,695 μs
τ =2,232 μs
$\delta_r = 10^{-6}$
Čas simulacije za zahtevano relativno natančnost:
t >30,84 μs
Število period: 30,84 $\mu s \cdot$ 80 $MHz \approx$ 2.467 period
Čas izračuna stacionarnega stanja: 500 s.

<u>Ekstrapolacijski algoritem</u>

Vrednosti parametrov:

t_{del} =0 (za vse iteracije ekstrapolacijskega algoritma)

m =14 (vrednost parametra m za vse iteracije ekstrapolacijskega algoritma)



Slika 5.33. Odziv mešalnika z uporabo ekstrapolacijskega algoritma

Število iteracij ekstrapolacijskega algoritma: 20

Skupaj period pri izračunu stacionarnega stanja z ekstrapolacijsko metodo: 280 period

5.3 Primerjava računske zahtevnosti

Primerjava računske zahtevnosti za izračun stacionarnega stanja z direktno tranzientno analizo in z uporabo ekstrapolacijskega algoritma je bila podana že v prejšnih poglavjih (Poglavje 5.2). Na tem mestu bomo zbrali podatke o številu period, ki jih je treba simulirati, da vezje doseže stacionarno stanje. Pri obeh postopkih je bila uporabljana ista natančnost izračuna (δ_r), da lahko rezultate med seboj primerjamo.

Poudariti je treba, da v tej primerjavi računske zahtevnosti ni upoštevan čas izračuna samega ekstrapolacijskega algoritma, temveč samo čas izračuna tranzientne analize oz. števila period. Čas izračuna ekstrapolacijskega algoritma je zanemarljiv v primerjavi s časom, ki je potreben za izračun tranzientne analize. Zaradi tega so faktorji, ki predstavljajo pospešitev direktne tranzientne analize, nekoliko večji od dejanskih, kjer bi upoštevali še čas izračuna ekstrapolacijskega algoritma.

V Tabeli 5.1 so zbrani podatki o številu period za deset testnih vezij. Prikazana je primerjava med direktno tranzientno analizo in metodo izračuna stacionarnega stanja z uporabo ekstrapolacijskega algoritma. Razmerje števila period je enako faktorju pospešitve.

Vezje	δ_r	Število period		faktor pospešitve
		tranzientna analiza	ekstrapolacijski algoritem	
Vezje 1: Frekvenčni množilnik	$5 \cdot 10^{-3}$	23	13	1,77
Vezje 2: Nelinearno usmerniško vezje RC	$5 \cdot 10^{-4}$	40	16,1	2,48
Vezje 3: Napetostni množilnik	$5 \cdot 10^{-6}$	90	30,1	2,99
Vezje 4: Ozkopasovni filter	10^{-5}	496	24	20,7
Vezje 5: Superozkopasovni filter s kvarcom	$5 \cdot 10^{-4}$	48.332	54	895
Vezje 6: Preklopni napajalnik	10^{-7}	15.300	30	510
Vezje 7: Greinacherjev usmernik	10^{-6}	73.800	312	237
Vezje 8: Oscilator	10^{-4}	34	16	2,13
Vezje 9: Nizkošumni ojačevalnik	10^{-4}	31	28	1,11
Vezje 10: Mešalnik	10^{-6}	2.467	280	8,81

Tabela 5.1. Primerjava računske zahtevnosti za testna vezja

Iz Tabele 5.1 je razvidno, da je število period, ki jih vezja potrebujejo za dosego stacionarnega stanja pri direktni tranzientni analizi, razporejeno v več velikostnih razredih (od 23 do 73.800 period). Število period pri ekstrapolacijskem algoritmu je razporejeno v območju nekaj 10 do nekaj 100 period. Posledica tega je, da je ekstrapolacijski algoritem najbolj učinkovit za vezja, pri katerih je število period pri direktni tranzientni analizi veliko (vezje 7) in najmanj učinkovito pri vezjih, pri katerih izračun stacionarnega stanja s tranzientno analizo ne traja dolgo (vezje 9 in vezje 1).

Pri vseh testnih vezjih je z uporabo ekstrapolacijske metode dosežena pospešitev izračuna stacionarnega stanja vezja, pri nekaterih vezjih celo večkratna. Ti rezultati potrjujejo upravičenost izbire ekstrapolacijske metode pri izračunu stacionarnega stanja kot tudi njeno implementacijo s programski paket SPICE OPUS. Metoda se je izkazala za uspešno pri vseh testnih vezjih, zato z veliko verjetnostjo pričakujemo uporabnost metode tudi na mnogih drugih vezjih, pri katerih izračun stacionarnega stanja traja dolgo.

Iz opisanih rezultatov je razvidno, da ni dosežena samo pospešitev izračuna stacionarnega stanja električnih vezij, temveč je dosežen tudi glavni namen doktorske disertacije – pospešitev postopka optimizacije električnih vezij, kjer je potreben izračun stacionarnega stanja. Če je znotraj optimizacijskega postopka zahtevan izračun samo stacionarnega stanja, smo z uporabo ekstrapolacijskega postopka pri izračunu stacionarnega stanja pospešili izračun optimizacije za isti faktor kot je pospešen sam izračun stacionarnega stanja. Primer optimizacije testnega vezja bo opisan na Poglavju 5.5.

5.4 Točnost izračuna

Iz Slik 5.14 do 5.33 in iz podatkov o stacionarnem stanju pri dovolj velikih časih, kjer vsi prehodni pojavi izzvenijo, je razvidno, da sta izračuna stacionarnega stanja po obeh metodah (direktna tranzientna analiza in izračun stacionarnega stanja z uporabo ekstrapolacijske metode) enaka v okviru natančnosti simulatorja oz. modelov elementov.

Razlike se gibljejo v območju nekaj desetink ali stotink odstotka, kar je že v mejah natančnosti simulatorja. Zato bi bilo pri obeh metodah nesmiselno pričakovati popolnoma enake rezultate.

Same razlike v rezultatih izvirajo tudi iz narave izračuna stacionarnega stanja vezja pri ekstrapolacijski metodi. V primerjavi z direktno tranzientno analizo, kjer izračun poteka tako kot zagon vezja v realnosti, pa pri ekstrapolacijski metodi vezje analiziramo samo delček časa, nato pa ustrezno nastavimo začetno stanje vezja in ponovno poženemo tranzientno analizo. Pri tem postopku, kjer vezje postavimo v neko nenaravno stanje, lahko vnesemo določene spremembe oz. napake, ki pa se skozi ekstapolacijski postopek ohranijo oz. akumulirajo in iz katerih tudi izvirajo odstopanja v primerjavi z direktno tranzientno analizo.

V Tabeli 5.2 so zbrani podatki iz Poglavja 5.2. Podatki prikazujejo primerjavo zgornje vrednosti ovojnice opazovanega signala za testna vezja pri izračunu stacionarnega stanja z direktno tranzientno analizo in pri uporabi ekstapolacijskega algoritma. Izračunana je tudi relativna razlika med tema dvema vrednostima, ki je izražena v odstotkih.

Točnost izračuna stacionarnega stanja z ekstrapolacijskim algoritmom v primerjavi z direktno tranzientno analizo, oz. relativna razlika se za izbrana testna vezja giblje od 0 % (izračun z uporabo ekstrapolacijskega algoritma je enak izračunu z direktno tranzientno analizo) do 0,97 %. Razlike so tako pozitivne kot negativne, povprečna relativna razlika preko vseh vezij pa je 0,17 %.

Vezje	Zgornja vrednost ovojnice opazovanega signala		relativna razlika
	tranzientna analiza	ekstrapolacijski algoritem	
Vezje 1: Frekvenčni množilnik	0,4962 V ($@t = 10\ \mu s$)	0,501 V	+0,97 %
Vezje 2: Nelinearno usmerniško vezje RC	6,537 V ($@t = 1\ s$)	6,528 V	-0,14 %
Vezje 3: Napetostni množilnik	1256 V ($@t = 10\ s$)	1268 V	+0,96 %
Vezje 4: Ozkopasovni filter	1,344 V ($@t = 10\ \mu s$)	1,341 V	-0,22 %
Vezje 5: Superozkopasovni filter s kvarcom	3,78 V ($@t = 2s$)	3,79 V	+0,26 %
Vezje 6: Preklopni napajalnik	6,7848 V ($@t = 100\ ms$)	6,7848 V	0,00 %
Vezje 7: Greinacherjev usmernik	1,6248 V ($@t = 100\ \mu s$)	1,6248 V	0,00 %
Vezje 8: Oscilator	55,4 V ($@t = 2\ ms$)	55,5 V	+0,18 %
Vezje 9: Nizkošumni ojačevalnik	1,569 V ($@t = 100\ ns$)	1,568 V	-0,06 %
Vezje 10: Mešalnik	3,58 mV ($@t = 20\ \mu s$)	3,57 mV	-0,28 %

Tabela 5.2. Primerjava točnosti izračuna stacionarnega stanja za testna vezja

5.5 Optimizacija

Za dodatno potrditev uporabnosti metode izračuna stacionarnega stanja z ekstrapolacijskim postopkom smo izvedli optimizacijo testnega vezja 7 (dvostopenjski Greinacherjev usmernik [72], ki se uporablja pri sistemih za prenos energije z mikrovalovi [73, 74, 75]) na Sliki 5.10. Razlog za izbiro tega vezja je bil, da vezje potrebuje največje število period za izračun stacionarnega stanja z direktno tranzientno analizo (73.800 period). Ocenjen čas optimizacije za nekaj tisoč iteracij optimizacijskega postopka z direktno tranzientno analizo se giblje (za povprečno zmogljiv računalnik) med nekaj deset in nekaj sto dnevi, kar je za sorazmerno enostavno vezje nesprejemljivo dolgo. Vezje je sestavljeno iz diod, ki so modelirane z MOS tranzistorji.

Pri optimizaciji izbranega testnega vezja bomo spreminjali dimenzije integriranih MOS tranzistorjev in kapacitivnosti kondenzatorjev. Optimizacijski parametri so podani v Tabeli 5.3

Optimizacijski parametri so omejeni z eksplicitnimi omejitvami, ki so podane v Tabeli 5.4. Začetne vrednosti parametrov so določene glede na tehnolgijo, ki jo uporabljamo (0,18 μm TSMC) in glede na zahteve, ki jih mora imeti vezje (velika enosmerna komponenta, majhna valovitost izhodnega signala). S temi vrednostmi, ki so rezultat ročne optimizacije načrtovalca,

Parameter 0 (p_0)	Širina tranzistorjev, modeliranih z diodama D11 in D12 (stopnja 1)
Parameter 1 (p_1)	Širina tranzistorjev, modeliranih z diodama D21 in D22 (stopnja 1)
Parameter 2 (p_2)	Širina tranzistorjev, modeliranih z diodama D11 in D12 (stopnja 2)
Parameter 3 (p_3)	Širina tranzistorjev, modeliranih z diodama D21 in D22 (stopnja 2)
Parameter 4 (p_4)	Kapacitivnosti vseh kondenzatorjev (stopnja 1 in 2)

Tabela 5.3. Optimizacijski parametri za testno vezje 7 (dvostopenjski Greinacherjev usmernik)

vezje karseda najbolje izpolnjuje zgoraj omenjene zahteve.

Optimizacijski parameter	Začetna vrednost	Minimalna vrednost	Maksimalna vrednost	korak parametra
p_0	10 μm	0,22 μm	100 μm	0,18 μm
p_1	10 μm	0,22 μm	100 μm	0,18 μm
p_2	8 μm	0,22 μm	100 μm	0,18 μm
p_3	8 μm	0,22 μm	100 μm	0,18 μm
p_4	100 pF	10 pF	300 pF	10 pF

Tabela 5.4. Eksplicitne omejitve optimizacijskih parametrov (začetne, minimalne, maksimalne vrednosti in korak)

Avtomatska optimizacija je potekala na podlagi optimizacijskih zahtev, ki so bile definirane preko meritev vezja [77, 78, 79, 80]. Zahtevana je bila določena enosmerna izhodna napetost in omejenost valovitosti pri bremenu na izhodu in brez njega. Na osnovi meritev vezja, zahtev za posamezno meritev, uteži in norme meritve je bila na podlagi enačb (2.14) in (2.15) tvorjena kriterijska funkcija, za katero postopek optimizacije poišče minimalno vrednost na določenem območju. Naklon kompromisnega področja $k_{t,i}$ je bil enak 0 za vse meritve ($i = 1, 2, 3, 4$). Naklon kazenskega območja $k_{p,i}$ je pa določen kot razmerje med utežjo in normo posamezne meritve. V Tabeli 5.5 so zbrane meritve vezja, zahteve, uteži, norme za meritve in nakloni kazenskih področij.

Meritvi 1 in 2 sta bili izvedeni pri bremenu (kot je prikazano na Sliki 5.10). Pri meritvah 3 in 4 je bilo breme (tokovni vir I_{load}) odstranjeno ($I_{load} = 0$). Meritve so definirane z enačbami (5.1)-(5.4).

$$V_{DC\_load} = \left[\frac{1}{T} \int_T (v_{OUT2}(t) - v_{OUT1}(t))dt \right]_{I_{load}=10\mu A} \tag{5.1}$$

Številka meritve (i)	meritev (m_i)	zahteva ($m_{G,i}$)	utež	norma	naklon kazenskega področja ($k_{p,i}$)
1	V_{DC_load}	$> 2\ V$	3	$1\ V$	$3\ V^{-1}$
2	V_{ripple_load}	$< 250\ \mu V$	1	$100\ \mu V$	$10^{-4} V^{-1}$
3	V_{DC_noload}	$> 3\ V$	1	$1\ V$	$1\ V^{-1}$
4	V_{ripple_noload}	$< 140\ \mu V$	1	$100\ \mu V$	$10^{-4} V^{-1}$

Tabela 5.5. Meritve, ki so bile izvedene na testnemu vezju

$$V_{ripple\_load} = \tag{5.2}$$
$$= \max_T \left[(v_{OUT2}(t) - v_{OUT1}(t))\right]_{I_{load}=10\mu A} - \min_T \left[(v_{OUT2}(t) - v_{OUT1}(t))\right]_{I_{load}=10\mu A}$$

$$V_{DC\_noload} = \left[\frac{1}{T}\int_T (v_{OUT2}(t) - v_{OUT1}(t))dt\right]_{I_{load}=0} \tag{5.3}$$

$$V_{ripple\_noload} = \tag{5.4}$$
$$= \max_T \left[(v_{OUT2}(t) - v_{OUT1}(t))\right]_{I_{load}=0} - \min_T \left[(v_{OUT2}(t) - v_{OUT1}(t))\right]_{I_{load}=0}$$

V meritvah uporabljeni integrali, maksimalne vrednosti in minimalne vrednosti so izračunane na intervalu dolžine $T = 417\ ps$, kolikor znaša perioda vhodnega signala oz. napetostnega vira V_s.

Z opisano postavitvijo optimizacije smo v 2964 iteracijah optimizacijske zanke dosegli povprečno 16% izboljšanje meritev glede na začetne vrednosti. Tabela 5.6 prikazuje vrednosti meritev pred in po optimizaciji, Tabela 5.7 pa vrednosti parametrov po optimizaciji.

Meritev	pred optimizacijo	po optimizaciji	izboljšanje
m_1	1,624 V	1,794 V	10,5 %
m_2	326,0 μV	249,9 μV	23,3 %
m_3	2,670 V	2,911 V	9,0 %
m_4	176,5 V	138,8 μV	21,3 %

Tabela 5.6. Vrednosti meritev pred in po optimizaciji

Iz Tabel 5.5 in 5.6 je razvidno, da zahtevi za meritvi 1 in 3 sicer nista bili izpolnjeni, kljub temu pa sta se vrednosti teh dveh meritev izboljšali za 10,5 % in 9,0 %. Zaradi tega vrednost kriterijske funkcije na koncu ni bila 0 (to bi se zgodilo v primeru, če bi bile izpolnjene zahteve vseh meritev), ampak je imela vrednost 0,6792.

Parameter	vrednost
p_0	16,60 μm
p_1	21,10 μm
p_2	27,76 μm
p_3	26,68 μm
p_4	300 pF

Tabela 5.7. Vrednosti parametrov po optimizaciji

Opisana optimizacija je na računalniku AMD Athlon XP 2500+ (1,83 GHz) s 512 MB RAM-a trajala 3 ure in 10 minut. V Poglavju 5.3 je bilo pokazano, da smo z uporabo ekstrapolacijskega algoritma za izbrano testno vezje dosegli 237-kratno pospešitev izračuna stacionarnega stanja. Iz teh podatkov in podatkov o času optimizacije sledi, da bi optimizacija, kjer bi izvajali direktno tranzientno analizo, trajala približno 31 dni oz. 1 mesec, kar je nesprejemljivo dolgo. Z uporabo hitrejšega račnalnika oz. več računalnikov hkrati bi sicer lahko ta čas skrajšali, vendar lahko izračun stacionarnega stanja in s tem tudi optimizacijskega postopka pospešimo že z uporabo ekstrapolacijskih metod, ki so bile predstavljene v prejšnjih poglavjih.

6

Zaključek

V doktorski disertaciji smo obravnavali izračun stacionarnega stanja električnih vezij z uporabo programskega paketa SPICE OPUS. Predstavljena je bila direktna metoda izračuna stacionarnega stanja, kjer izvajamo običajno tranzientno analizo toliko časa, dokler vsi prehodni pojavi ne izzvenijo. Prikazani so bili problemi, ki pri takem izračunu določenih vezij nastanejo in zaradi katerih izračun stacionarnega stanja traja nesprejemljivo dolgo.

V nadaljevanju so bile opisane metode, s pomočjo katerih lahko izračunamo stacionarno stanje električnih vezij veliko hitreje, ne da bi bilo potrebno vezje analizirati preko celotnega časovnega intervala. Izmed možnih rešitev je bila izbrana metoda, ki uporablja ekstrapolacijske postopke. Za potrditev primernosti smo metodo najprej simulirali s pomočjo klicev programskega orodja SPICE OPUS iz zunanjega programa in obdelave tako dobljenih rezultatov.

Kot najbolj primeren se je izkazal algoritem epsilon, ki je bil nato implementiran v programski paket SPICE OPUS kot nova analiza, imenovana *ssse* (Steady-State Shooting by Extrapolation). Učinkovitost analize izračuna stacionarnega stanja je bila potrjena na desetih realnih primerih vezij. Pri vseh testnih vezjih se je izračun stacionarnega stanja pospešil, pri nekaterih tudi do več stokrat. Rezultati, ki so bili dobljeni po obeh postopkih (algoritem epsilon in tranzientna analiza), so v okviru natančnosti modelov elementov ekvivalentno enaki. Dodatno smo učinkovitost metode pokazali še pri optimizaciji testnega vezja, ki bi z uporabo direktne tranzientne analize trajala en mesec, z uporabo analize *ssse* pa dobimo rezultat optimizacije v nekaj več kot treh urah, kar predstavlja 237-kratno pospešitev.

Dodatna prednost analize *ssse* pred izračunom stacionarnega stanja z direktno tranzientno analizo je tudi v tem, da pri slednji vnaprej ne vemo, kdaj naj zaključimo izračun oz. kako dolg naj bo časovni interval, preko katerega naj tranzientna analiza poteka. Pri izračunu si želimo le določene absolutne in relativne natančnosti. Če smo ti dve natančnosti dosegli, moramo preveriti po zaključku izvajanja tranzientne analize in v nasprotnem primeru podaljšati čas tranzientne analize. Prednost analite *ssse* pa je v tem, da vnaprej nastavimo željeno relativno in absolutno natančnost izračuna stacionarnega stanja, nakar sam algoritem analize poskrbi, da se ustavi, ko je zahtevana natančnost dosežena.

6.1 Izvirni prispevki k znanosti

Doktorska disertacija vsebuje naslednje samostojne in izvirne prispevke k znanosti:

1. Analiza zahtevnosti izračuna stacionarnega stanja električnih vezij z obstoječim programskim paketom SPICE.
2. Izbor primernih metod za hiter izračun stacionarnega stanja električnih vezij in simulacija njihove primernosti za optimizacijske algoritme.
3. Implementacija ustreznega algoritma za izračun stacionarnega stanja električnega vezja v obstoječ programski paket SPICE in potrditev učinkovitosti na realnih primerih.

V Poglavju 2.3 je predstavljena zahtevnost izračuna stacionarnega stanja električnih vezij z uporabo direktne tranzientne analize programskega paketa SPICE. Predstavljeni so tudi primeri, pri katerih je izračun stacionarnega stanja dolgotrajen. Poglavje 4.3 opisuje metode, s pomočjo katerih lahko stacionarno stanje izračunamo mnogo hitreje kot z direktno tranzientno analizo. Izbrane metode so v Poglavju 4.2 simulirane, s čemer potrdimo njihovo primernost za uporabo pri optimizacijskih algoritmih. V Poglavju 4.3 je na podlagi treh kriterijev izbrana najprimernejša metoda, ki je bila implementirana v programski paket SPICE (Poglavje 4.4). Učinkovitost implementirane metode je v Poglavju 5 potrjena na več realnih primerih vezij.

Spodaj je navedeno objavljeno delo, s katerim sem si pridobil pravico za neposredni prehod na doktorski študij:

WAGNER BORUT, BÜRMEN ARPAD, PUHAN JANEZ, FAJFAR IZTOK, TUMA TADEJ, *Computing the steady-state response of nonlinear circuits by means of the ϵ-algorithm*, Elektroteh. vestn., 2005, letn. 72, št. 5, str. 297-302, [COBISS.SI-ID 5120596]

Spodnji članek je bil objavljen v reviji s faktorjem vpliva po SCI in vsebuje del rezultatov raziskav, ki so obravnavane v tej doktorski disertaciji.

WAGNER BORUT, BÜRMEN ARPAD, PUHAN JANEZ, TOMAŽIČ SAŠO, TUMA, TADEJ, *Application of extrapolation algorithms in nonlinear circuit simulation and optimization*

with SPICE OPUS, Informacije MIDEM, 2006, letn. 36, št. 3, str. 140-147, [COBISS.SI-ID 5674580]

7

Priloge

Članek ERK 2004: [45]

B. WAGNER, J. PUHAN, Á. BŰRMEN, I. FAJFAR, T. TUMA, *Simulacija in optimizacija analognih električnih vezij s pomočjo ravnotežne harmonske analize*, Zbornik trinajste mednarodne Elektrotehniške in računalniške konference ERK 2004, 27. – 29. september 2004, Portorož, Slovenija. Ljubljana, zv. A, str. 29–32.

Članek ERK 2005: [81]

B. WAGNER, Á. BŰRMEN, J. PUHAN, I. FAJFAR, T. TUMA, *Uporaba ekstrapolacijskih metod pri računanju stacionarnega stanja električnih vezij*, Zbornik štirinajste mednarodne Elektrotehniške in računalniške konference ERK 2005, 26. – 28. september 2005, Portorož, Slovenija. Ljubljana, zv. A, str. 82–85.

Članek EV 2005: [82]

B. WAGNER, Á. BŰRMEN, J. PUHAN, I. FAJFAR, T. TUMA, *Computing the steady-state response of nonlinear circuits by means of the ϵ-algorithm*, Elektroteh. vestn., 2005, vol. 72, no. 5, str. 297–302.

Članek Midem 2006: [83]

B. WAGNER, Á. BŰRMEN, J. PUHAN, S. TOMAŽIČ, T. TUMA, *Application of extrapolation algorithms in nonlinear circuit simulation and optimization with SPICE OPUS*, Informacije MIDEM, 2006, letn. 36, št. 3, str. 140–147.

Članek ERK 2006: [84]

B. WAGNER, Á. BŰRMEN, J. PUHAN, S. TOMAŽIČ, T. TUMA, *Optimizacija dvostopenjskega*

Greinacherjevega polnovalnega usmernika, Zbornik petnajste mednarodne Elektrotehniške in računalniške konference ERK 2006, 25. – 27. september 2006, Portorož, Slovenija. Ljubljana, zv. A, str. 96–99.

7.1 Članek ERK 2004

Simulacija in optimizacija analognih električnih vezij s pomočjo ravnotežne harmonske analize

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

Simulation and Optimization of Analog Circuits Using Harmonic Balance Analysis

This paper presents Harmonic Balance analysis simulation technique of nonlinear analog circuits [1, 2, 3]. It is used for circuits where steady-state response is required. The method has many advantages for circuits which require considerable time to reach the steady-state (resonant circuits, oscillators…). Algorithm is coded in Nutmeg (interpreter designed as a user interface for SPICE [4], but it has all properties of a programming language). Transient analysis is performed in SPICE OPUS [5], nonlinear analog circuit simulator. Harmonic balance analysis is demonstrated on a simple circuit. Results are compared with transient analysis. This method can be used for optimization of nonlinear analog circuits [6, 7].

1. Uvod

Simulacija analognih elektronskih vezij je računsko dokaj zahteven postopek. Pri določenih vezjih nas pogosto zanima, kakšen je odziv vezja v stacionarnem stanju, ko prehodni pojavi izzvenijo. Zato moramo simulirati delovanje vezja včasih tudi preko nekaj 100 ali celo 1000 period, da prehodni pojav popolnoma izzveni. Če želimo tako vezje optimizirati, t.j. poiskati vrednosti določenih parametrov tako, da bo najbolje izpolnjevalo načrtovalske zahteve, moramo vezje simulirati velikokrat z različnimi vrednostmi optimizacijskih parametrov, kar občutno upočasni celoten računski postopek. Da dobimo rezultate v doglednem času, bi morali uporabiti zmogljiv računalnik ali uporabiti paralelno procesiranje [8].

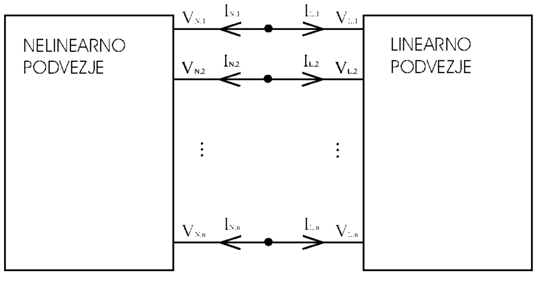
Za nelinearna elektronska vezja, za katera nas ne zanima prehodni pojav, ki jih vzbujamo s periodičnim signalom in od katerih pričakujemo periodični odziv, lahko za izračun odziva v stacionarnem stanju uporabimo metodo ravnotežne harmonske analize [1]. Metoda izkazuje prednosti pred tranzientno analizo predvsem za vezja, ki imajo veliko kvaliteto Q. Da bi izračunali stacionarni odziv teh vezij, bi morali simulirati vezje preko 100 ali celo nekaj 1000 period, da bi prehodni pojav izzvenel, za kar bi porabili dosti časa. Pri ravnotežni harmonski analizi dobimo odziv vezja v stacionarnem stanju dosti prej, kar pomeni, da bo tudi morebitni postopek optimizacije, ki ga bomo izvajali na tem vezju, precej hitrejši.

Pri metodi ravnotežne harmonske analize vezje razdelimo na nelinearen in linearen del. Linearni del vezja rešujemo v frekvenčnem prostoru, nelinearni pa v časovnem, tako da simuliramo nekaj osnovnih period signala odziva. S pomočjo Fourierjeve transformacije dobimo frekvenčne komponente napetosti in tokov na vozliščih med nelinearnim in linearnim delom vezja. Na teh vozliščih uporabimo napetostni in tokovni Kirchhoffov zakon in s pomočjo iteracijske metode dosežemo, da so tokovi posameznih frekvenčnih komponent iz enega dela vezja v drug del enaki (od tod ime ravnotežna harmonska analiza).

2. Ravnotežna harmonska analiza

Pogosto nas za določena vezja zanima stacionarni odziv, ko prehodni pojavi popolnoma izzvenijo. Za vezja, ki jih vzbujamo s harmoničnimi signali, lahko s pomočjo ravnotežne harmonske analize do tega rezultata pridemo hitreje, kot če bi vezje simulirali s tranzientno analizo. Da bi prehodni pojav izzvenel, bi morali vezje simulirati preko nekaj 100 ali 1000 period, za kar bi porabili precej časa.

Nelinearno analogno vezje razdelimo na nelinearno in linearno podvezje, ki sta med seboj spojeni v *n* vozliščih (slika 1).



Slika 1: Razdelitev vezja na nelinearni in linearni del.

Signale, ki so prisotni v vezju, predstavimo v obliki Fourierjeve vrste (1)

$$x(t) = \sum_{\omega_k \in \Omega} A_k \cos(\omega_k t) + B_k \sin(\omega_k t), \qquad (1)$$

pri čemer so koeficienti A_k in B_k Fourierjevi koeficienti, ki pripadajo frekvencam ω_k. Ω je množica frekvenc, ki jih vsebujejo signali v vezju.

Vezje vzbujamo s harmoničnimi viri. Zaradi nelinearnosti je odziv vezja v splošnem linearna kombinacija vseh večkratnikov vseh harmoničnih nihanj, s katerimi vezje vzbujamo. Tako dobimo množico frekvenc **Ω**, ki nastopajo v odzivu vezja. Pri določanju odziva na podlagi ravnotežne harmonske analize je v množici **Ω** končno število frekvenc. Ker bi morala ta množica vsebovati neskončno mnogo frekvenc, naredimo pri izračunu napako, kar pomeni, da enačba (5) v nadaljevanju ni popolnoma izpolnjena.
Za izbiro množice Ω na podlagi signala vzbujanja lahko uporabimo različne načine izbire frekvenc, ki jih upoštevamo pri določevanju odziva vezja [1].

Linearni del vezja analiziramo v frekvenčnem prostoru. Iz vozliščnih napetosti $\mathbf{V_L}$ (slika 1) s pomočjo topologije vezja in lastnosti elementov izračunamo tokove $\mathbf{I_L}$, ki tečejo v linearni del vezja [9].

Napetosti na vozliščih nelinearnega podvezja $\mathbf{V_N}$ najprej s pomočjo inverzne Fourierjeve transformacije pretvorimo v časovni prostor

$$\mathbf{v}_N(t) = F^{-1}\{\mathbf{V}_N\} \; . \tag{2}$$

Potem s tranzientno analizo na podlagi vzbujanja $\mathbf{v_N}(t)$ izračunamo tokove v nelinearni del vezja $\mathbf{i_N}(t)$, ki jih potem s Fourierjevo transformacijo pretvorimo nazaj v frekvenčni prostor

$$\mathbf{I}_N = F\{\mathbf{i}_N(t)\} \; . \tag{3}$$

Na vozliščih med nelinearnim in linearnim delom najprej uporabimo napetostni Kirchhoffov zakon

$$\mathbf{V}_L = \mathbf{V}_N \; . \tag{4}$$

Cilj metode harmoničnega ravnovesja je poiskati napetosti med nelinearnim in linearnim delom vezja, pri katerih bodo tokovi, ki tečejo in enega dela v drug del vezja, enaki (5) (tokovni Kirchhoffov zakon). Glej sliko 1.

$$\mathbf{I}_L = -\mathbf{I}_N \tag{5}$$

Definirajmo funkcijo napake, ki je enaka vsoti tokov, ki za posamezno vozlišče tečejo v nelinearni in linearni del vezja:

$$\mathbf{E} = \mathbf{I}_L + \mathbf{I}_N \; . \tag{6}$$

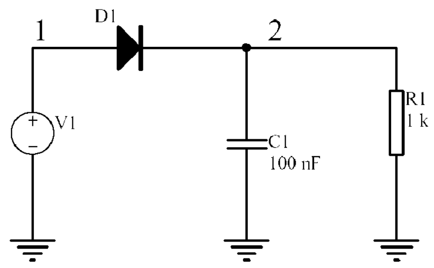
Funkcijo napake **E** je treba z iteracijskim algoritmom minimizirati, t.j. doseči, da bo **E**=**0**. Tedaj bo zaradi (5) izpolnjen še tokovni Kirchhoffov zakon in vezje bo v ravnovesju.
Funkcijo napake lahko minimiziramo z Newtonovo iteracijsko metodo

$$\mathbf{V}^{(i+1)} = \mathbf{V}^{(i)} - \left[\mathbf{J}^{(i)}\right]^{-1}\mathbf{E}^{(i)} \; , \tag{7}$$

pri čemer predstavlja indeks i zaporedno iteracijo. Matrika **J** je Jacobijeva matrika $\mathbf{J} = \partial\mathbf{E}/\partial\mathbf{U}$.

3. Testno vezje

Za demonstracijo delovanja ravnotežne harmonske analize smo izbrali enostavno vezje (slika 2).

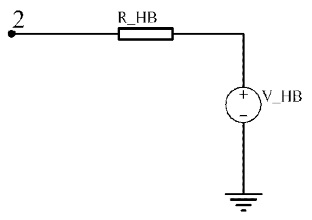


Slika 2: Testno vezje.

Vezje smo simulirali v časovnem prostoru s programskim orodjem za simulacijo analognih elektronskih vezij SPICE OPUS [5].
Metodo harmoničnega ravnovesja smo implementirali v programskem jeziku Nutmeg, ki je del programa SPICE OPUS. Na sliki 6 je prikazan odziv vezja na podlagi tranzientne analize in s pomočjo ravnotežne harmonske analize.

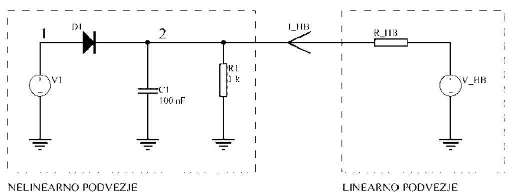
Vezje na sliki 2 vzbujamo v vozlišču 1 z napetostnim virom $v_1(t) = 1\text{V} + 1\text{V}\sin(2\pi \cdot 1\text{MHz} \cdot \text{t})$. Za vezje želimo izračunati odziv v vozlišču 2, torej $v_2(t)$. Odločimo se, da bomo opazovali enosmerno komponento, osnovno frekvenčno komponento in osem višje harmonskih komponent, torej bo množica Ω v enačbi (1) podana v obliki $\Omega = \{2\pi k \cdot 1\text{MHz}; 0 \le k \le 9\}$. Ker smo vzeli končno število frekvenčnih komponent, funkcija napake **E** (6) nikoli ne more biti enaka **0**. Zato smo z iteracijo (7) končali, ko je bila izpolnjena neenačba $\|\mathbf{E}\| < 10^{-4}$.

V večini vezij, ki jih simuliramo, so skoraj vsi elementi nelinearni. Zato smo tudi za to testno vezje modificirali metodo ravnotežne harmonske analize. Celotno vezje, ki ga simuliramo (slika 2), smo proglasili za nelinearno podvezje, v vozlišče 2, v katerem želimo izračunati odziv, pa dodamo linearno podvezje, ki je prikazano na sliki 3. Če bi imelo vezje na sliki 2 več vozlišč, v katerih bi želeli izračunati odziv, bi morali v vsako vozlišče dodati tako linearno vezje.



Slika 3: Linearno podvezje.

Celotno vezje, ki ga uporabimo za simulacijo po metodi ravnotežne harmonske analize, je prikazano na sliki 4.



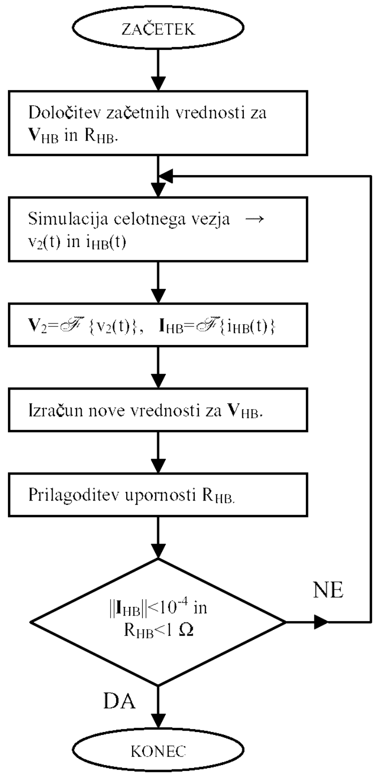
Slika 4: Vezje za simulacijo po metodi ravnotežne harmonske analize.

Ker smo metodo ravnotežne harmonske analize modificirali na zgoraj opisan način, je sedaj funkcija napake (6) enaka $\mathbf{E}=\mathbf{I}_{HB}$ na sliki 4. Ko bo tok $\mathbf{I}_{HB}$, ki je posledica dodajanja vezja iz slike 3, enak **0**, bo vezje v ravnovesju. Takrat bo napetost v vozlišču 2 $v_2(t)$ enaka napetosti generatorja $v_{HB}(t)$.

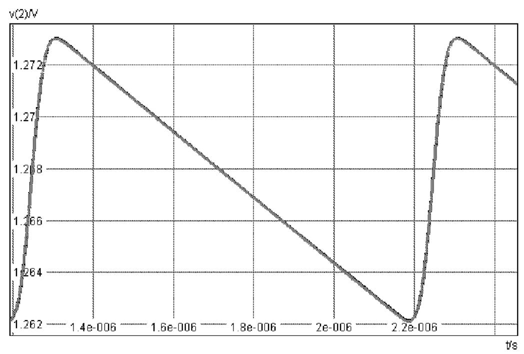
Na začetku nastavimo vrednost napetostnega vira $\mathbf{V}_{HB}$ na **0** in upornost R_{HB}=1 kΩ. Celotno vezje s slike 4 simuliramo v časovnem prostoru. S tem dobimo odziv vezja ($v_2(t)$ in $i_{HB}(t)$).
V vsaki iteraciji na podlagi toka i_{HB} z enačbo (3) izračunamo $\mathbf{I}_{HB}$ in z uporabo (7) ter znanega upora R_{HB} izračunamo novo vrednost napetostnih virov $\mathbf{V}_{HB}$. Če je tok $\mathbf{I}_{HB}$ že dovolj majhen, lahko ustrezno spremenimo vrednost upornosti R_{HB}.
Iteracijski postopek določevanja $v_2(t)$ ponavljamo toliko časa, dokler ni izpolnjena neenačba $\|\mathbf{E}\| < 10^{-4}$ oz. $\|\mathbf{I}_{HB}\| < 10^{-4}$ in da ima upornost R_{HB} dovolj majhno vrednost. Celoten iteracijski postopek prikazuje slika 5.

Odziv vezja na sliki 2 $v_2(t)$ na podlagi metode ravnotežne harmonske analize ter s pomočjo tranzientne analize (po daljšem času, ko prehodni podaj izzveni) je prikazan na sliki 6. Razlika med obema odzivoma je neopazna (10^{-5}), zato je na sliki 7 prikazana še razlika obeh odzivov.

Na sliki 7 opazimo, da v napaki, ki je enaka razliki odzivov po metodi ravnotežne harmonske analize in tranzientni analizi, prevladuje frekvenčna komponenta 10 MHz. Ta ugotovitev se ujema z izbiro množice **Ω**, saj smo predpostavili, da odziv vezja vsebuje enosmerno, osnovno in 8 višje harmonskih komponent (frekvence do 9 MHz).

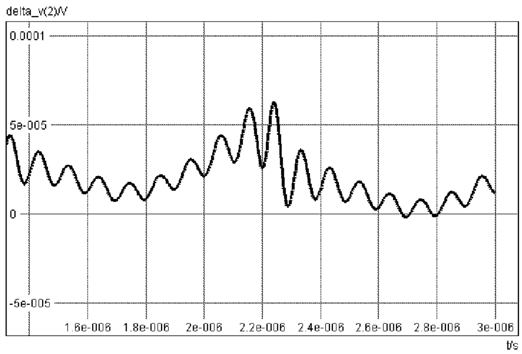


Slika 5: Iteracijski postopek za določevanje odziva vezja po metodi ravnotežne harmonske analize.



Slika 6: Odziv testnega vezja, dobljen z ravnotežno harmonsko analizo in s tranzientno analizo (razlika je neopazna).

Za analizo testnega vezja je bilo potrebno 36 iteracij (slika 5). V vsaki iteraciji je potrebno vezje simulirati časovni interval vsaj ene periode osnovne frekvence 1 MHz. Dosti časa porabimo za računanje Fourierjeve transformacije (SPICE računa s pomočjo DFT in ne FFT), tako da ima opisana metoda prednost pred tranzientno analizo, če bi bilo treba vezje simulirati preko vsaj 100 period, da bi dobili stacionarni odziv. Vezje, ki je bilo predstavljeno v tem članku, je namenjeno zgolj za prikaz delovanja metode ravnotežne harmonske analize, saj dobimo stacionarni odziv že po nekaj 10 periodah.



Slika 7: Razlika odzivov vezja, dobljenih z ravnotežno harmonsko analizo in s tranzientno analizo.

4. Optimizacija analognih elektronskih vezij

Opisano metodo lahko uporabimo pri optimizaciji analognih elektronskih vezij [7, 8], kjer moramo vezje, ki ga želimo optimizirati, velikokrat simulirati. Tako lahko celoten postopek optimizacije traja tudi več dni. V vsaki simulaciji spreminjamo določene parametre vezja (vrednosti elementov vezja…), dokler vezje ne izpolnjuje zahtev načrtovalca. Z uporabo metode ravnotežne harmonske analize se lahko optimizacija vezja občutno skrajša.

5. Izbira inačice ravnotežne harmonske analize

Zaradi enostavne implementacije v Nutmegu je bila uporabljena osnovna metoda ravnotežne harmonske analize [1, 2, 3, 11]. Iteracijski postopek (7) je bil izveden s pomočjo blokovne Newtonove metode [12] (iteracija za vsako frekvenčno komponento posebej).
Druge različice metode ravnotežne harmonske analize se razlikujejo po tem, na kakšen način izvedemo pretvorbo iz časovnega v frekvenčni prostor [13, 14], na kakšen način predstavimo nelinearnost nelinearnih elementov [15, 16].
Izbira ustrezne različice je odvisna od tega, ali želimo metodo ravnotežne harmonske analize vgraditi v obstoječi simulator. Pri tem moramo upoštevati njegove lastnosti in način delovanja, saj vse različice niso enako ugodne za implementacijo. Drugače je, če načrtujemo nov simulator vezij, v katerega bomo vgradili metodo ravnotežne harmonske analize. V tem primeru bomo simulator v osnovi načrtovali tako, da bomo lahko implementirali različico, ki je najbolj učinkovita.
Za implementacijo v SPICE OPUS je ugodna tudi uporaba t.i. shooting metode v kombinaciji ekstrapolacijsko metodo [1, 10].

6. Zaključek

V članku smo predstavili ravnotežno harmonsko analizo, ki smo jo uporabili kot metodo za analizo analognih električnih vezij. Metoda je uporabna, če nas zanima odziv vezja v stacionarnem stanju. Ravnotežna harmonska analiza ima prednosti pred tranzientno analizo za vezja, katera bi morali simulirati dolgo časa, da bi prišla v stacionarno stanje (resonančna vezja, oscilatorji…). Prikazali smo delovanje metode na enostavnem vezju, primerjali odziv s tranzientno analizo in preučili možnost uporabe te metode pri optimizaciji.

7. Viri

[1] Kenneth S. Kundert, Jacob K. White and Alberto Sangiovanni-Vincentelli, *Steady-state methods for simulating analog and microwave circuits*, Kluwer Academic Publishers, 1990

[2] Michael S. Nakhla and Jiri Vlach, *A Piecewise Harmonic Balance Technique for Determination of Periodic Response of Nonlinear Systems*, IEEE Transactions on Circuit and Systems, Vol. CAS-23, No. 2, February 1976, pp. 85-91

[3] Vittorio Rizzoli and Andrea Neri, *State of the Art and Present Trends in Nonlinear Microwave CAD Techniques*, IEE Transactions on Microwave Theory and Techniques, Vol. 36, No. 2, February 1988, pp. 343-365

[4] T. Quarles, A. R. Newton, D. O. Pederson, A. Sangiovanni-Vincentelli, *SPICE3 Version 3f4 User's Manual*, University of California, Berkeley, California, 1989

[5] http://www.fe.uni-lj.si/spice/

[6] J. Puhan, T. Tuma, *Optimization of analog circuits with SPICE 3f4*, Proceedings of the ECCTD'97, vol. 1, pp. 177-180, 1997

[7] J. Puhan, T.Tuma, I. Fajfar, *Optimisation Methods in SPICE, a Comparison*, *Proceedings* of the ECCTD'99, vol. 1, pp. 1279-1282, 1999

[8] Arpad Bűrmen, Iztok Fajfar, Janez Puhan, Andrej Nussdorfer, Tadej Tuma, *Optimizacija elektronskih vezij z vzporednim omejenim simpleksnim postopkom*, Elektrotehniški vestnik 69(1): 7–12, Ljubljana, 2002

[9] T. Tuma and F. Bratkovič, *A General Approach to Circuit Equations*, The International Journal of Circuit Theory and Applications, Vol. 22, pp. 431–445, 1994

[10] Stig Skelboe, *Computation of the Periodic Steady-State Response of Nonlinear Network by Extrapolation Methods*, IEEE Transactions on Circuits and Systems, Vol. CAS-27, No. 3, March 1980, pp. 161-175

[11] Vittorio Rizzoli, Claudio Cechetti, Alessandro Lipparini and Franco Mastri, *General-Purpose Harmonic Balance Analysis of Nonlinear Microwave Circuits Under Multitone Excitation*, IEEE Transactions on Microwave Theory and Techniques, Vol. 36, No. 12, December 1988, pp. 1650-1660

[12] Chao-Ren Chang, Patrick L. Heron and Michael B. Steer, *Harmonic Balance and Frequency-Domain Simulation of Nonlinear Microwave Circuits Using the Block Newton Method*, IEEE Transactions on Microwave Theory and Techniques, Vol. 38, No. 4, April 1990, pp. 431-434

[13] Patrick L. Heron and Michael B. Steer, *Jacobian Calculation Using the Multidimensional Fast Fourier Transformation in the Harmonic Balance Analysis of Nonlinear Circuits*, IEEE Transcations on Microwave Theory and Techniques, Vol. 38, No. 4, April 1990, pp. 429-431

[14] Akio Ushida and Leon O. Chua, *Frequency-Domain Analysis on Nonlinear Circuits Driven by Multi-Tone Signals*, IEEE Transactions on Circuits and Systems, Vol. CAS-31, No. 9, September 1984

[15] George W. Rhyne, Michael B. Steer and Bevan D. Bates, *Frequency-Domain Nonlinear Circuit Analysis Using Generalized Power Series*, IEEE Transactions on Microwave Theory and Techniques, Vol. 36, No. 2, February 1988, pp. 379-387

[16] Janne Roos, *Frequency-domain analysis of nonlinear circuits using Chebyshev polynomials*, Master's Thesis, Helsinki University of Technology, February 22, 1994

8. Zahvala

Raziskave je sofinanciralo Ministrstvo za šolstvo znanost in šport (MŠZŠ) Republike Slovenije v okviru programa P2-0246 - Algoritmi in optimizacijski postopki v telekomunikacijah.

7.2 Članek ERK 2005

Uporaba ekstrapolacijskih metod pri računanju stacionarnega stanja električnih vezij

Borut Wagner, Árpád Bűrmen, 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 sitih in ojačevalnikih, preklopnih napajalnikih itd. Električno vezje je treba simulirati tudi več 1000 period, da se prehodni pojavi iznihajo 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

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

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

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

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

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

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

$$\mathbf{x}^{-1} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2^2}. \qquad (4)$$

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

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

pri čemer vzamemo za $\mathbf{x_0}(0) = \rho_{2k}^{(0)}$.
Posebna inačica rho algoritma predpostavlja, da je zaporedje $\left\{\|\mathbf{x}^{(n)}\|\right\}_{n=0,1,2,\ldots}$ 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.

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

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

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

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

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

Pri sodih indeksih k računamo inverz vektorja glede na $\mathbf{\Delta}\epsilon_{2k}^{(n)}$, pri lihih 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

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

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

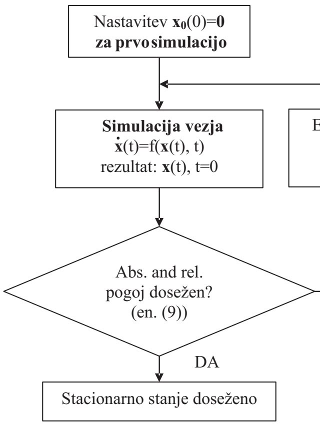
Iteracijski postopek je prikazan na sliki 1.

4 Primerjava metod

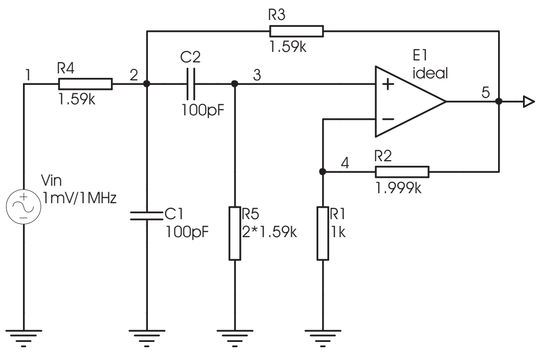
Opisane ekstrapolacijske algoritme smo preizkusili na dveh vzorčnih vezjih: ozkopasovnem situ in preklopnem napajalniku. Električni shemi vezij vidimo na slikah 2 in 4.

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

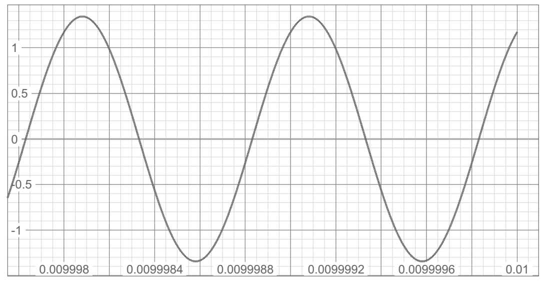


Slika 2: Ozkopasovno sito

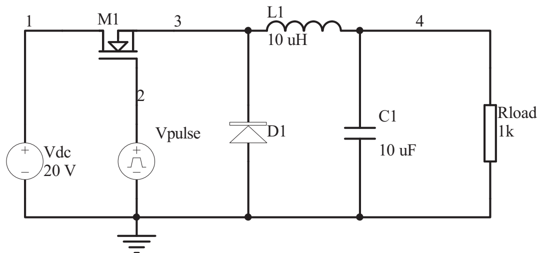
$1k\Omega$.
Odziv preklopnega napajalnika v stacionarnem stanju je prikazan na sliki 5.

Vezji smo najprej simulirali z običajno tranzientno analizo. Za dosego 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 dosego stacionarnega stanja, je označeno z N_ϵ, N_ρ, N_ϑ 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.



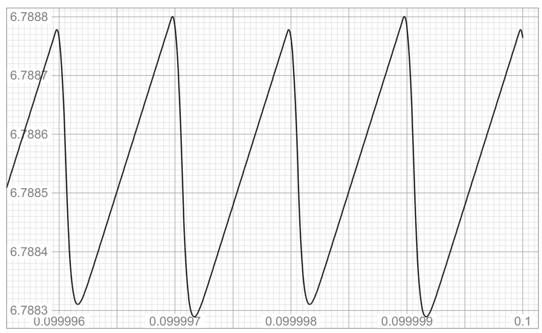
Slika 3: Stacionarni odziv ozkopasovnega sita (vozlišče $v_5(t)$)



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 (vozlišče $v_4(t)$)

	Ozkopasovno sito	Preklopni napajalnik
N_{TRAN}	10.000	10.000
N_{ϵ}	12	24
N_{ρ}	94	30
N_{ϑ}	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 algotitem-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 Convergenc and the Summantion 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. Puhan, 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. Bűrmen, J. Puhan, I. Fajfar, T. Tuma, “Computing the steady-state response of nonlinear circuits by means of ϵ-algorithm” *Elektrotehniški vestnik*, vol. 72 (2):??–??, Ljubljana, Slovenija, 2005.

[9] J. Puhan, T. Tuma, “Optimization of analog circuits with SPICE 3f4”, *Proceedings of the ECCTD ’97*, vol. 1, pp. 177-180, 1997.

7.3 Članek EV 2005

Elektrotehniški vestnik XX(Y): 1–6, YEAR
Electrotechnical Review, Ljubljana, Slovenija

Computing the steady-state response of nonlinear circuits by means of the ϵ-algorithm

Borut Wagner, Árpád Bűrmen, Janez Puhan, Iztok Fajfar and Tadej Tuma

University of Ljubljana, Faculty of Electrical Engineering,
Tržaška cesta 25, SI-1000 Ljubljana, Slovenia
E-mail: borut.wagner@fe.uni-lj.si

Abstract. The paper presents the problem of computing the steady-state response of electronic circuits. The direct approach to obtaining the steady-state response (transient analysis) necessitates a considerable amount of time.
The process of obtaining the steady-state response through transient analysis can be accelerated by using extrapolation methods. The ϵ-algorithm is the most appropriate extrapolation algorithm for implementation in the SPICE OPUS circuit simulator developed at the Faculty of Electrical Engineering of the University of Ljubljana.
A short description of the algorithm is given upon which implementation details are discussed. The algorithm is tested on two circuits. The computation time needed by the direct approach is compared to the time needed by the ϵ-algorithm.
The results show that the ϵ-algorithm is appropriate for the rapid evaluation of the steady-state response of circuits excited by a single periodic signal. Finaly, directions for future research are given.

Key words: steady-state response, nonlinear circuits, epsilon algorithm, circuit simulation, SPICE

Računanje stacionarnega odziva nelinearnih vezij z ϵ-algoritmom

Povzetek. V članku je predstavljen problem računanja stacionarnega odziva nelinearnih elektronskih vezij s pomočjo časovne (tranzientne) analize, ki lahko za določena vezja traja zelo dolgo. Tranzientno analizo se da pospešiti s pomočjo ekstrapolacijskih metod. ϵ-algoritem je najprimernejši za vgradnjo v simulator vezij SPICE OPUS, ki ga razvijamo na Fakulteti za elektrotehniko v Ljubljani.
Opisana sta princip delovanja ϵ-algoritma in izvedba analize za določanje stacionarnega stanja na podlagi analize v časovnem prostoru (tranzientne analize).
Algoritem je bil testiran na dveh testnih vezjih. Vezji sta bili simulirani s tranzientno analizo in s pospešeno tranzientno anlizo. Primerjava časov, ki so bili potrebni za izračun stacionarnega odziva, kaže, da je ϵ-algoritem zelo primeren za vezja, ki so vzbujana z enim signalnim virom.
Na koncu članka so podane smernice za nadaljnje raziskave na tem področju.

Ključne besede: stacionarni odziv, nelinearna vezja, algoritem epsilon, simulacija vezij, SPICE

1 Introduction

Computing the steady-state response of a nonlinear electrical circuit is usually much time and computer power consuming, especially for circuits with time constants much larger than the period of the steady-state response. A direct approach to computing the steady-state response necessitates running a transient analysis until all initial transients die off.

A problem occurs when the period of the steady-state response is much smaller than the largest time constant of the circuit. Such circuits must be simulated for hundreds or even thousands of periods before they reach the steady-state. To obtain sufficient accuracy, a hundred or more time points must be evaluated per period of the response. Therefore, the simulation often takes several million time points before steady-state is reached.

Several techniques exist that speed up the computation of the steady-state response [1, 2, 3, 4, 5]. The main difference between them is the domain in which the analysis is performed (frequency domain, time domain, mixed time-frequency domain).

Due to its simplicity and the fact that it doesn't require any major changes in the simulator, the most appropriate algorithm for implementation in SPICE OPUS [6, 7, 8] is the ϵ-algorithm [3, 9, 10, 11]. Originally the ϵ-algorithm was developed for accelerating the convergence of series.

In the sections that follow, a quick overwiev of

Received 2. February, 2005
Accepted 20. October, 2005

the problem of obtaining the steady-state response is given followed by an introduction to the ϵ-algorithm. The implementation details of the ϵ-algorithm are discussed with emphasis on the parameters that determine its performance. The algorithm is tested on two circuits and its performance is compared to the performance of the direct approach. In the end, conclusions are given and directions for the future research are discussed.

2 Circuit simulation with SPICE

Nonlinear dynamical electrical circuits can be described by a system of ordinary differential equations

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

where $\mathbf{x}(t)$ represents branch voltages, node voltages and branch currents. In transient analysis, Eq. (1) is solved starting from the initial value $\mathbf{x}_0(t_0)$. Resulting waveforms are represented by $\mathbf{x}(t)$ for $t_0 \leq t \leq t_n$. For further reference, $\mathbf{x}(t)$ is a column matrix with m components:

$$\mathbf{x}(t) = \left[\, x_1(t), \quad x_2(t), \quad \ldots \quad x_m(t) \,\right]^T. \tag{2}$$

Eq. (1) is numerically integrated and solved using the Newton-Raphson iterative method [6, 12] for times $t_1 \leq t_2 \leq \ldots \leq t_n$. Time step $t_{\Delta i} = t_{i+1} - t_i$ should be taken small enough to avoid numerical errors. Decreasing the time step increases the computation time, especially when the steady-state response of a circuit is sought with the direct approach.

Circuit simulation is a part of every circuit optimization [13] where parameters of the circuit are looked for subject to designer's requirements. During optimization, a circuit is simulated many times, so individual simulations should be as short as possible.

3 Steady-state response

The steady-state response of an electronic circuit is obtained after all initial transients disappear. The time required to reach the steady state depends on characteristics of the circuit and the excitation frequency. The simulation time is particularly large when the largest time constant of the circuit is much greater than the period of the highest frequency exciting the circuit.

In practice, the circuit should be simulated more than ten times the largest time constant of the circuit in order to attain a sufficient accuracy of the steady-state response. After the simulation is finished, the accuracy of the computed steady-state response can be checked. The circuit is in the steady state if

$$\begin{gathered}
|x_i(t_n) - x_i(t_{n-1})| = 0 \\
t_n = t_0 + t + nT \\
t_{n-1} = t_0 + t + (n-1)T \\
i = 1, 2, \ldots, m
\end{gathered} \tag{3}$$

for all $0 \leq t \leq T$, n is large enough ($n \to \infty$) and the period of the steady-state response is T. A circuit is simulated for a finite number of periods T, so Eq. (3) is not exactly satisfied. We can assume that the steady state has been reached when relative and absoulte tolerance criteria

$$|x_i(t_n) - x_i(t_{n-1})| \leq \delta_a + \max\left[|x_i(t_n)|, |x_i(t_{n-1})|\right]\delta_r \tag{4}$$

for all $i = 1, 2, \ldots, m$, and n large enough are satisfied.

In practice, it is sufficient if δ_r and δ_a are less than 10^{-5} and 10^{-4}, respectively, but greater than precision of data representation (e.g. relative and absolute precision of *double* is 10^{-14} and 10^{-320}, respectively).

Relative and absolute criteria (Eq. (4)) can also be used when the steady-state response of a circuit is sought by means of the ϵ-algorithm.

The steady-state response is important in the analysis of power conversion circuits and in evaluation of nonlinear properties of narrow-band circuits excited by a single frequency.

4 ϵ-algorithm

Suppose that the circuit has a steady-state response with period T. So the sequence

$$\mathbf{x}^{(i)} = \mathbf{x}(t_0 + t + iT), \qquad i = 0, 1, \ldots \tag{5}$$

converges for all $0 \leq t \leq T$.

Now form a two-dimensional array depicted in Fig. 1 where the following equations represent the relationships of individual array entries for some value of t.

$$\begin{aligned}
\epsilon_{-1}^{(i)} &= \mathbf{0}, & i &= 1, 2, 3, \ldots \\
\epsilon_{0}^{(i)} &= \mathbf{x}^{(i)}, & i &= 0, 1, 2, \ldots \\
\epsilon_{j+1}^{(i)} &= \epsilon_{j-1}^{(i+1)} + (\epsilon_{j}^{(i+1)} - \epsilon_{j}^{(i)})^{-1}, & i &= 0, 1, 2, \ldots \\
& & j &= 0, 1, 2, \ldots
\end{aligned} \tag{6}$$

$$
\begin{array}{cccccc}
 & \epsilon_0^{(0)} & & & & \\
\epsilon_{-1}^{(1)} & & \epsilon_1^{(0)} & & & \\
 & \epsilon_0^{(1)} & & \epsilon_2^{(0)} & & \\
\epsilon_{-1}^{(2)} & & \epsilon_1^{(1)} & & \epsilon_3^{(0)} & \\
 & \epsilon_0^{(2)} & & \epsilon_2^{(1)} & & \epsilon_4^{(0)} \\
\epsilon_{-1}^{(3)} & & \epsilon_1^{(2)} & & \epsilon_3^{(1)} & \vdots \\
 & \epsilon_0^{(3)} & & \epsilon_2^{(2)} & \vdots & \\
\epsilon_{-1}^{(4)} & & \epsilon_1^{(3)} & \vdots & & \\
\vdots & \epsilon_0^{(4)} & \vdots & & & \\
 & \vdots & & & &
\end{array}
$$

Figure 1. Two-dimension array for the ϵ-algorithm

The procedure described by Eq. (6), is the so-called ϵ-algorithm [3, 9, 10, 11]. If the inverse of the vector in Eq. (6) is taken component-wise

$$\mathbf{v}^{-1} = (v_1^{-1}, v_2^{-1}, \ldots, v_m^{-1}), \tag{7}$$

the scalar ϵ-algorithm is obtained. If however

$$\mathbf{v}^{-1} = \mathbf{v}/\left\|\mathbf{v}\right\|_2^2, \tag{8}$$

(6) represents the vector ϵ-algorithm. Here only the vector ϵ-algorithm will be discussed.

By means of the ϵ-algorithm the convergence of the sequence in Eq. (5) can be accelerated [3, 9, 10, 11]. Consecutive sequences $\{\epsilon_j^{(i)}\}_{i=0}^{\infty}$, $j = 2, 4, 6, \ldots$ converge faster than the original sequence $\{\epsilon_0^{(i)}\}_{i=0}^{\infty}$. The greater the value of j the faster the sequence converges.

In one iteration of the ϵ-algorithm a circuit is simulated for $i = 2k$ periods. Then $\epsilon_{2k}^{(0)}$ is obtained using Eq. (6) and represents the initial condition for the next iteration of the ϵ-algorithm.

5 SPICE OPUS and the ϵ-algorithm

The ϵ-algorithm was tested in combination with the SPICE OPUS circuit simulator [7]. The circuit is first simulated for several periods. Results of the simulation are then interpreted by an external program which extracts the state of the circuit $\mathbf{x}^{(i)}(t_i)$ at $t_i = t_0 + t + iT$, $i = 0, 1, 2, \ldots, n$ for selected t. These values represent $\{\epsilon_0^{(i)}\}_{i=0}^{n}$. Next $\{\epsilon_j^{(i)}\}_{i=0}^{n-j}$, $j = 2, 4, 6, \ldots$ using Eq. (6) is obainted. The last calculated ϵ-value $\epsilon_n^{(0)}$ represents new initial conditions $\mathbf{x}_0(t_0)$ for simulating the circuit in the next iteration of the ϵ-algorithm. The simulation is started at time $t = 0$, so t_0 is always 0. The algorithm is depicted in Fig. 2.

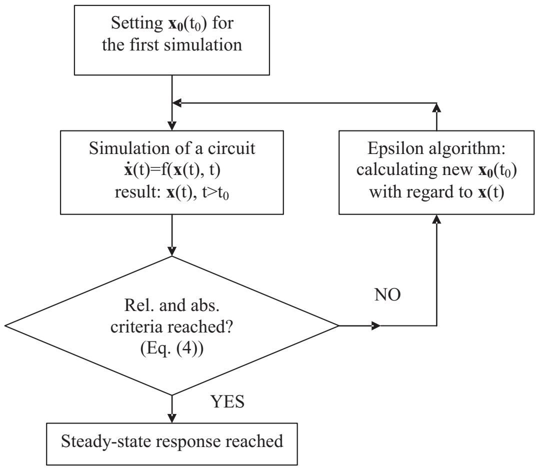


Figure 2. ϵ-algorithm for determing the steady-state response

The speed of convergence depends on the following parameters.

Parameter *frequency*
The parameter *frequency* represents the base frequency of the response of the circuit ($T = 1/frequency$). The *frequency* depends on the frequency of the input signal. In case of autonomous circuits (e.g. oscillators), the *frequency* should be determined from the response of the circuit. If the circuit is not in its steady state yet, the *frequency* can change in every iteration of the ϵ-algorithm.

Parameters *tdel* **and** *tdelSec*
Parameters *tdel* and *tdelSec* represent t, $t \geq 0$. $t = tdel$ in the first iteration of the ϵ-algorithm and $t = tdelSec$ for the remaining iterations. Normally, *tdel* (and *tdelSec*) should be 0. For cases, where the response of the circuit includes some initial rapidly decreasing transients, *tdel* (or/and *tdelSec*) can be set to a non zero value to speed up the convergence.

Parameters *NumPointsMin*, *NumPointsMax* **and** *PointsStep*
Values of $\mathbf{x}^{(i)}$ are extracted from the response of the circuit. The number of $\mathbf{x}^{(i)}$ ($i = 0, 1, 2, \ldots, i_{max}$) that enter into the ϵ-algorithm can be set with parameters *NumPointsMin* and *NumPointsMax*. For the first iteration of the ϵ-algorithm, $i_{max} = NumPointsMin$ is used. In next iterations, i_{max} is increased by *PointsStep* until $i_{max} = NumPointsMax$ is reached.

Parameter *PeriodsPerPoint*

If differences between $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(i+1)}$ are very small, performing the ϵ-algorithm with Eq. (6) could result in a large numerical error. In such cases, the time between $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(i+1)}$ can be multiplied by $PeriodsPerPoint = 1, 2, 3, 4, \ldots$. With this multiplication we practically increased the period T in Eq. (5) so the differences between $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(i+1)}$ become larger.

Parameter $FreqMax$
When a circuit is simulated with SPICE OPUS, an appropriate initial time step t_Δ has to be chosen to get sufficiently accurate results for the ϵ-algorithm. At least 100 points must be evaluated within one period T, so $t_\Delta \leq T/100$. t_Δ is calculated from $FreqMax$: $t_\Delta = 1/(2 \cdot FreqMax)$.

Parameters $EpsStopAbsTol$ and $EpsStopRelTol$
Iterations of ϵ-algorithm are stopped when the stopping criterion (Eq. (4)) is satisfied ($\delta_a = EpsStopAbsTol$ and $\delta_r = EpsStopRelTol$).

Parameter $MaxIters$
If the stopping criterion (Eq. (4)) can't be satisfied (bad choice of parameters), the ϵ-algorithm is stopped when the number of iterations exceeds $MaxIters$.

Determing the above described parameters is very important for fast obtaining the steady-state response. If the choice of parameters is inappropriate, the steady state can't be reached by the algorithm and one ends up with wrong results.

In practice, parameters should be modified until the steady-state response is reached in a few iterations of the ϵ-algorithm. If the circuit is being optimized, parameters need to be determined only once and can remain unchanged for the rest of the optimization. The optimal parameter values are similar for similar circuits. Therefore, the parameter values can be set depending on the type of the circuit and its characteristics.

6 Test circuits

Efficiency of the ϵ-algorithm was tested with two test circuits. For both circuits, the time required for calculating the steady-state response by means of ϵ-algorithm is substantiality smaller than the time required by the direct approach.

Voltage multiplier
The first circuit is a voltage multiplier, Fig. 3.
At the input of the circuit (between nodes 9 and 10), a sine voltage source with amplitude 311 V and frequency 50 Hz is connected. Output of the circuit is

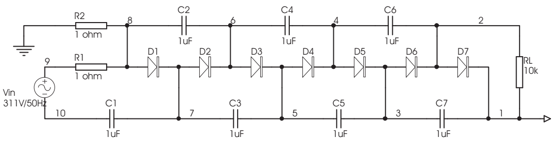


Figure 3. Test circuit 1 (voltage multiplier)

at node 1. If the circuit is simulated for a sufficient amount of time, the response in Fig. 4 is obtained.

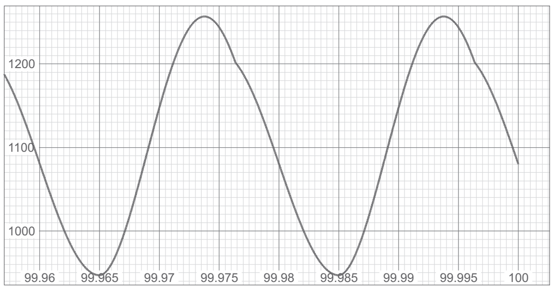


Figure 4. Steady-state response ($v_1(t)$) of the voltage multiplier

Narrow-band filter
Circuit in Fig. 5 is a narrow-band filter with quality $Q = 100$ and resonant frequency 1 MHz.

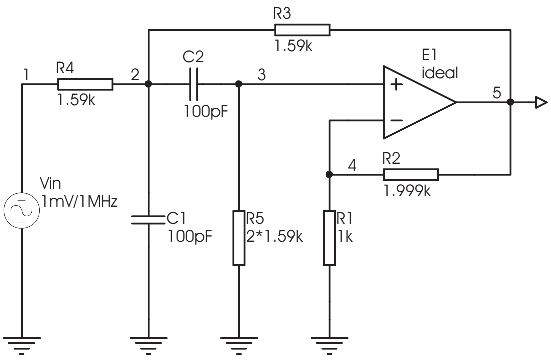


Figure 5. Test circuit 2 (narrow-band filter)

The input of the circuit is at node 1 and the output at node 5. If a sine voltage source with the amplitude 1 mV and frequency 1 MHz is connected to the input, the steady-state response in Fig. 6 is obtained.

7 Comparison of the ϵ-algorithm with the direct approach

In this section, computation times needed for obtaining the steady-state response for the direct approach

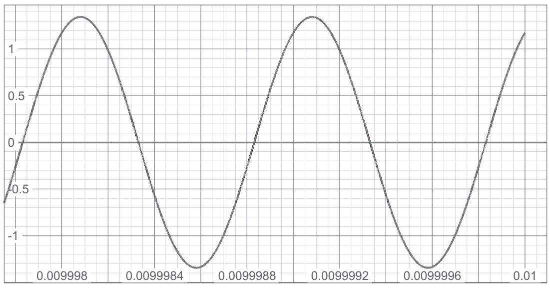


Figure 6. Steady-state response ($v_5(t)$) of the narrow-band filter

and the ϵ-algorithm are compared. Results are shown in Table 1.

Transient analysis (direct approach)
The circuits were simulated from time $t = 0$ to $t = t_{SS}$ with initial time step t_Δ. The number of periods simulated was N_{TRAN}.

ϵ-algorithm (accelerated transient analysis)
Values for the parameters are listed in Table 1. Using these parameter values, the steady-state response was reached after I_{ITER} iterations of the ϵ-algorithm. The total number of periods simulated in the ϵ-algorithm was N_{EPS}.

Parameter value	Voltage multiplier	Narrow-band filter
t_Δ	10 μs	0.5 ns
t_{SS}	10 s	10 ms
N_{TRAN}	5,000	10,000
tdel/tdelSec	2 ms / 0 ms	1 μs / 1 μs
MaxFreq	50 kHz	1 GHz
NumPointMin	4	4
NumPointMax	4	16
PointStep	2	2
PeriodsPerPoint	1	1
EpsStopAbsTol	$5 \cdot 10^{-5}$	10^{-7}
EpsStopRelTol	$5 \cdot 10^{-6}$	10^{-7}
I_{ITER}	9	2
N_{EPS}	36.1	12
C_{acc}	139	833

Table 1. Comparison of the ϵ-algorithm and the direct approach

By computing the steady-state response, the direct approach (transient analysis) was accelerated by factor $C_{acc} = N_{TRAN}/N_{EPS}$. The computation time required by the ϵ-algorithm itself was not taken into consideration for being neglectable.

8 Application of the ϵ-algorithm

The ϵ-algorithm can be used for RF circuits when the steady-state response is needed. If the circuit is being optimized (many simulations of the same circuit in an optimization loop), the faster computation of the steady-state response considerably shortens the optimization process.
The ϵ-algorithm can also be applied to other areas of electrical engineering, e.g. telecommunications [14].

9 Conclusion

The ϵ-algorithm described in this paper proved successful in the computing of the steady-state response of electronic circuits. For the two test circuits, the steady-state response was obtained in more than 100 times shorter time than with the direct approach. This achievement is very important when a circuit has to be simulated many times, e.g. in parametric optimization of a circuit. The main disadvantage of using the ϵ-algorithm is that some properties of the circuit must be known in advance in order to set the parameters of the ϵ-algorithm. When the parameters are set, they can remain unchanged for the remaining steady-state simulations of the circuit.

10 Future work

Future research will focus on testing the ϵ-algorithm with more circuits.
Automatic parameter tuning in the ϵ-algorithm would help the user when characteristics of a circuit are unknown. Automatic parameter tuning will help determining parameters through several runs of the ϵ-algorithm with different values of parameters.
The ϵ-algorithm will be implemented into the SPICE OPUS circuit simulator [7] as a new analysis for computing the steady-state response of circuits.

Other steady-state evaluation techniques are also being considered [1, 2, 4, 5]. Currently, they are not quite appropriate for implementation in SPICE OPUS since some radical changes and additions are needed.
Once different methods for computing the steady-state response are implementated, a global steady-state analysis will be included in SPICE OPUS that will switch between them automatically.

11 References

[1] K. S. Kundert, J. K. White, A. Sangiovanni-Vincentelli *"Steady-state methods for simulation analog and microwave circuits"*, Kluwer Academic Publishers, 1990.

[2] K. S. Kundert, G. B. Sorkin, A. Sangiovanni- Vincentelli, "Applying Harmonic Balance to Almost-Periodic Circuits", *IEEE Transactios on Microwave Theory and Techniques*, vol. 36, no. 2, pp. 366-378, February 1988.

[3] 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, March 1980.

[4] N. Soveiko, M. S. Nakhla, "Steady-State Analysis of Multitone Nonlinear Circuits in Wavelet Domain", *"IEEE Transactios on Microwave Theory and Techniques"*, vol. 52, no. 3, pp. 785-797, March 2004.

[5] Janne Roos, *"Frequency-domain analysis of nonlinear circuits using Chebyshev polynomials"*, Master's Thesis, Helsinki University of Technology, February 22, 1994.

[6] T. Quarles, A. R. Newton, D. O. Pederson, A. Sangiovanni-Vincentelli, *"SPICE3 Version 3f4 User's Manual"*, University of California, Berkeley, California, 1989.

[7] 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/ .

[8] J. Puhan, T. Tuma, I. Fajfar, "SPICE for Windows 95/98/NT", *Electrotechnical Review, vol. 65, no. 5, pp. 267-271, Ljubljana, Slovenia, 1998.*

[9] J. A. Steele, A. T. Dolovich, "Toward the kernel of the vector epsilon algorithm", *International Journal for Numerical Methods in Engineering*, vol. 48, no. 5, pp. 721-730, June 2000.

[10] X. Gourdon, P. Sebah, *Convergence acceleration of series*, January 10, 2002,
URL: http://numbers.computation.free.fr/
Constants/Miscellaneous/seriesacceleration.html .

[11] 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, February 1986.

[12] G. A Korn, T. M. Korn, *"Mathematical handbook for scientists and engineers"*, McGraw-Hill Book Company, 1961.

[13] J. Puhan, T. Tuma, "Optimization of analog circuits with SPICE 3f4", *Proceedings of the ECCTD'97*, vol. 1, pp. 177-180, 1997.

[14] S. Tomažič, "On optimality of quadrature amplitude modulation", *Proceedings of the International Conference on Advances in the Internet, Processing, Systems, and Interdisciplinary Research*, Sveti Stefan, Montenegro, IPSI, 2004, October 2-9, 2004.

Borut Wagner received his B. Sc. degree in Electrical Engineering in 2003 from the Faculty of Electrical Engineering of the University of Ljubljana, Slovenia, where he is employed as a member of the national young researcher scheme. He is currently working towards a Ph. D. degree at the Electronic Design Automation Laboratory at the same faculty. His research interests include steady-state analyses of nonlinear electrical circuits, circuit simulation and circuit optimization.

Árpád Bűrmen received his B. Sc. and Ph. D. degrees in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana, Slovenia, in 1999 and 2003, respectively. From 1999-2002 he worked as a junior researcher at the Faculty of Electrical Engineering. Since 2002 he has been a Teaching Assistant at the same faculty. His research interests include analog and mixed-mode simulation of circuits and systems, automated design of analog circuits, optimization methods and their convergence and applications, and parallel distributed algorithms.

Janez Puhan received his B. Sc., M. Sc. and Ph. D. degrees in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana, Slovenia, in 1993, 1998 and 2000, respectively. Since 1996 he has been with the same faculty where he is an Assistant Professor. His research interests include computer-aided design of analog circuits, optimisation methods, and computer-aided circuit analysis.

Iztok Fajfar received his B. Sc., M. Sc. and Ph. D. degrees in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana, in 1991, 1994 and 1997, respectively. Since 1992 he has been with the same faculty where he is currently an Associate Professor. He teaches several introductory and advanced courses in computer programming. His research interests include design and optimisation of electronic circuits.

Tadej Tuma received his B. Sc, M. Sc. and Ph. D. degrees from University of Ljubljana, Faculty of Electrical Engineering, in 1988, 1991, and 1995, respectively. He is an Associate Professor at the same faculty where he teaches four undergraduate and three postgraduate courses. His research interests are mainly in the field of computer-aided circuit analysis and design.

12 Acknowledgment

The research has been supported by the Ministry of Higher Education, Science and Technology of Republic of the Slovenia within programme P2-0246 - Algorithms and optimization methods in telecommunications.

7.4 Članek Midem 2006

Application of extrapolation algorithms in nonlinear circuit simulation and optimization with SPICE OPUS

Borut Wagner, Árpád Bűrmen, Janez Puhan, Sašo Tomažič, Tadej Tuma

University of Ljubljana, Faculty of Electrical Engineering, Tržaška cesta 25, Ljubljana, Slovenia

Key words: extrapolation algorithms, circuit simulation, circuit optimization, integrated circuit design, SPICE.

Abstract

In this paper the extrapolation algorithms for vector sequence acceleration are presented. Four extrapolation algorithms are described and their application to circuit simulation is discussed. Steady state evaluation times are compared for the presented extrapolation algorithms and the direct method on real-world test circuits. Results show that the most appropriate extrapolation algorithm for evaluating the steady state of a circuit is the epsilon algorithm. The epsilon algorithm was implemented in SPICE OPUS circuit simulator. The implemented epsilon algorithm was used for optimizing the steady-state response of a test circuit. The accelerated evaluation makes the optimization of steady-state response possible.

Uporaba ekstrapolacijskih postopkov pri simulaciji in optimizaciji nelinearnih vezij s programskim paketom SPICE OPUS

Ključne besede: ekstrapolacijski postopki, simulacija električnih vezij, optimizacija električnih vezij, načrtovanje integriranih vezij, SPICE.

Povzetek

V članku so predstavljeni ekstrapolacijski postopki, ki se uporabljajo za pospeševanje konvergence zaporedij. Opisani so štirje postopki in njihov način uporabe pri simulaciji električnih vezij. Podana je primerjava časov za računanje stacionarnega stanja testnih električnih vezij brez in z uporabo ekstrapolacijskih postopkov. Za implementacijo v programski paket SPICE OPUS je bil izbran epsilon algoritem, ki se je izkazal za najbolj primernega. Postopek je bil uporabljen tudi pri optimizaciji testnega vezja. Zaradi hitrejšega izračuna samega stacionarnega stanja se je pohitril tudi celoten postopek optimizacije stacionarnega stanja in postal primeren za praktično uporabo.

1 Introduction

Extrapolation algorithms for vector sequences /1,2,3,4/ are used for accelerating sequences that converge slowly. The limit of a sequence can be calculated efficiently by evaluating only a few terms of the sequence without the explicit knowledge of the sequence generator. Using an extrapolation algorithm and only few terms of the sequence, a new initial term of the sequence can be calculated. With the new initial term, further terms of the sequence are evaluated by the sequence generator. The procedure is iterated until the differences between consequent terms of the sequence are small enough to assume that we are close to the limit of the sequence.

When computing the steady-state response /5/ of a nonlinear circuit all signals in the circuit are assumed to have the same fundamental frequency f and the period $T = 1/f$. The circuit is simulated starting at some initial condition until steady state is reached. Simulation results $\mathbf{x}(t)$, $t \geq 0$ represent node voltages and branch currents of the circuit at time t. The sequence $\{\mathbf{x}_0^{(i)} = \mathbf{x}(iT)\}$, $i = 0,1,2,3...$ is convergent if the circuit has a steady-state response with period T.
To accelerate the computation of steady state, the circuit is simulated for n periods and the extrapolation method is used on vectors $\mathbf{x}_0^{(0)}, \mathbf{x}_0^{(1)}, \mathbf{x}_0^{(2)}, ...\mathbf{x}_0^{(n)}$ to compute the new initial vector $\mathbf{x}_1^{(0)}$. Then the circuit is simulated starting with initial conditions $\mathbf{x}_1^{(0)}$ and the new sequence $\{\mathbf{x}_1^{(i)} = \mathbf{x}(iT)\}$, $i = 0,1,2,3...$ is extracted from the response of the circuit. This is repeated k-times until $\left\|\mathbf{x}_k^{(n)} - \mathbf{x}_k^{(0)}\right\|$ is small enough to assume the circuit is in steady state.

The paper is organized as follows. In Section 2 a brief description of four extrapolation algorithms (epsilon algorithm, rho algorithm, theta algorithm and topological epsilon algorithm) is given. In Section 3 the simulation of electrical circuits with SPICE OPUS is presented. Problems that occur when steady-state response of circuits has to be computed are presented. The acceleration of the steady-state response computation by means of extrapolation algorithms is described. In Section 4 simulation times for the test circuits (Greinacher rectifier, narrow-band filter, switching power supply) are compared for the direct approach (transient analysis until all initial transients die off) and for the accelerated steady-state computation by means of extrapolation algorithms. Section 5 describes the implementation details of the selected extrapolation algorithm in SPICE OPUS. In Section 6, the implemented algorithm is used in the optimization of a test circuit. Section 7 concludes the paper.

2 Extrapolation algorithms

Vector sequence

$$\{\mathbf{x}_0^{(i)}\}, \quad i = 0,1,2,3... \tag{1}$$

is generated from the initial vector $\mathbf{x}_0^{(0)}$ using the sequence generator $\mathbf{x}_0^{(i+1)} = \mathbf{F}(\mathbf{x}_0^{(i)})$, $i = 0,1,2,3...$. If the sequence $\{\mathbf{x}_0^{(i)}\}$ is convergent, the limit of the sequence is denoted by $\mathbf{x}$.
An extrapolation algorithm $\mathbf{E}$ generates a new vector sequence $\{\mathbf{x}_{k+1}^{(0)}\}$, $k = 0,1,2,3...$ with the following two steps

$$\mathbf{x}_{k+1}^{(0)} = \mathbf{E}(\mathbf{x}_k^{(0)}, \mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)} ... \mathbf{x}_k^{(n_k)}), \quad k = 0,1,2,3... \tag{2}$$

and

$$\mathbf{x}_k^{(i)} = \mathbf{F}(\mathbf{x}_k^{(i-1)}), \quad i = 1,2,3...n_k \text{ for each } k = 0,1,2,3... \,. \tag{3}$$

The extrapolation algorithm generates the first term $\mathbf{x}_{k+1}^{(0)}$ of a new sequence $\{\mathbf{x}_{k+1}^{(i)}\}, i = 0,1,2...$ from the $n_k + 1$ terms of sequence $\{\mathbf{x}_k^{(i)}\}, i = 0,1,2...n_k$.
For each k, $n_k + 1 \leq d$, $d = \dim(\mathbf{x}_k^{(0)})$ has to be satisfied. If $n_k + 1 > d$ the extrapolation algorithm is overdetermined.

The sequence $\{\mathbf{x}_{k+1}^{(0)}\}, k=0,1,2,3...$ generated by (2) and (3) converges faster than sequence (1) to the same limit $\mathbf{x}$.

In the following subsections four extrapolation algorithms are described: epsilon algorithm, rho algorithm, theta algorithm and topological epsilon algorithm. The algorithms differs only in the definition of $\mathbf{E}$.

2.1 Epsilon algorithm

For each k in (2) a two-dimension array depicted in Fig. 1 is formed by

$$\boldsymbol{\varepsilon}_{-1}^{(i)}=\mathbf{0}\ ,\ i=1,2,3..n_k \tag{4a}$$

$$\boldsymbol{\varepsilon}_0^{i}=\mathbf{x}_k^{(i)}\ ,\ i=0,1,2..n_k \tag{4b}$$

$$\boldsymbol{\varepsilon}_{j+1}^{(i)}=\boldsymbol{\varepsilon}_{j-1}^{(i+1)}+(\boldsymbol{\varepsilon}_j^{(i-1)}-\boldsymbol{\varepsilon}_j^{(i)})^{-1}\ ,\ i=0,1,2..n_k\ ,\ j=0,1,2..n_k-1 \text{ where } i+j+1\le n_k \tag{4c}$$

$$\begin{array}{ccccccc}
 & & \boldsymbol{\varepsilon}_0^{(0)} & & & & \\
\boldsymbol{\varepsilon}_{-1}^{(1)} & & & & \boldsymbol{\varepsilon}_1^{(0)} & & \\
 & & \boldsymbol{\varepsilon}_0^{(1)} & & & \ddots & \\
\boldsymbol{\varepsilon}_{-1}^{(2)} & & & & \boldsymbol{\varepsilon}_1^{(1)} & & \\
 & & \boldsymbol{\varepsilon}_0^{(2)} & & & \cdots & \boldsymbol{\varepsilon}_{n_k}^{(0)} \\
 & & & & \boldsymbol{\varepsilon}_1^{(2)} & & \\
\vdots & & \vdots & & \vdots & & \\
 & & \boldsymbol{\varepsilon}_0^{(n_k-1)} & & & \cdot^{\cdot^{\cdot}} & \\
\boldsymbol{\varepsilon}_{-1}^{(n_k)} & & & & \boldsymbol{\varepsilon}_1^{(n_k-1)} & & \\
 & & \boldsymbol{\varepsilon}_0^{(n_k)} & & & &
\end{array}$$

Figure 1: Two-dimension array for the epsilon algorithm

The procedure described by (4), is the so-called epsilon algorithm /1/. The inverse of the vector in (4c) is computed as

$$\mathbf{x}^{-1}=\frac{\mathbf{x}}{\|\mathbf{x}\|_2^2}\ . \tag{5}$$

If n_k is an even number, $\boldsymbol{\varepsilon}_{n_k}^{(0)}$ is the result of the epsilon algorithm, so the extrapolation function (2) is defined as

$$\mathbf{E}(\mathbf{x}_k^{(0)},\mathbf{x}_k^{(1)},\mathbf{x}_k^{(2)}...\mathbf{x}_k^{(n_k)})=\boldsymbol{\varepsilon}_{n_k}^{(0)}\ . \tag{6}$$

2.2 Rho algorithm

The rho algorithm /1/ is similar to the epsilon algorithm. For each k in (2) n_k has to be even. The result is $\boldsymbol{\rho}_{n_k}^{(0)}$ obtained from

$$\boldsymbol{\rho}_{-1}^{(i)} = \mathbf{0} \ , \ i = 1,2,3..n_k \tag{7a}$$
$$\boldsymbol{\rho}_{0}^{i} = \mathbf{x}_k^{(i)} \ , \ i = 0,1,2..n_k \tag{7b}$$
$$\boldsymbol{\rho}_{j+1}^{(i)} = \boldsymbol{\rho}_{j-1}^{(i+1)} + (j+1)\left[\boldsymbol{\rho}_{j}^{(i-1)} - \boldsymbol{\rho}_{j}^{(i)}\right]^{-1} \ , \ i = 0,1,2..n_k \ , \ j = 0,1,2..n_k - 1 \text{ where } i + j + 1 \le n_k \,. \tag{7c}$$

The inverse of the vector in (7c) is calculated in the same manner as in (5).

2.3 Theta algorithm

Extrapolation by theta algorithm /1/ (for an even n_k) is computed using the following equations

$$\theta_{-1}^{(i)} = \mathbf{0} \ , \ i = 1,2,3..n_k \tag{8a}$$
$$\theta_{0}^{i} = \mathbf{x}_k^{(i)} \ , \ i = 0,1,2..n_k \tag{8b}$$
$$\theta_{2j+1}^{(i)} = \theta_{2j-1}^{(i+1)} + \left[\Delta\theta_{2j}^{(i)}\right]^{-1} \ , \qquad i = 0,1,2..n_k \ , \ j = 0,1,2...\frac{n_k}{2} - 1 \text{ where } i + 2j + 1 \le n_k \tag{8c}$$
$$\theta_{2j+2}^{(i)} = \theta_{2j}^{(i+1)} + \left(\left[\Delta\theta_{2j}^{(i+1)}\right], \left[\Delta\theta_{2j+1}^{(i+1)}\right]\right) \times \left(\Delta^2\theta_{2j+1}^{(i+1)}\right)^{-1}, \tag{8d}$$
$$i = 0,1,2..n_k \ , \ j = 0,1,2...\frac{n_k}{2} - 1 \text{ where } i + 2j + 2 \le n_k$$

The abbreviations in (8c) and (8d) are

$$\Delta\theta_{j}^{(i)} = \theta_{j}^{(i+1)} - \theta_{j}^{(i)} \tag{8e}$$
$$\Delta^2\theta_{j}^{(i)} = \Delta\theta_{j}^{(i+1)} - \Delta\theta_{j}^{(i)} \tag{8f}$$

and the inverses in (8c) and (8d) are calculated by (5).

For each k and even n_k the theta algorithm extrapolation function $\mathbf{E}$ is defined as

$$\mathbf{E}(\mathbf{x}_k^{(0)}, \mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)} ... \mathbf{x}_k^{(n_k)}) = \theta_{n_k}^{(0)} \tag{9}$$

2.4 Topological epsilon algorithm

If the inverse of a vector $\mathbf{x}$ is defined as

$$\mathbf{x}^{-1} = \frac{\mathbf{y}}{(\mathbf{y}, \mathbf{x})} \ , \tag{10}$$

the inverse is the so-called inverse of vector $\mathbf{x}$ with respect to $\mathbf{y}$.

In the topological epsilon algorithm the inverses in odd and even terms are computed with respect to different vectors.

For each interation k of the extrapolation algorithm, the topological epsilon algorithm /1/ is defined by

$$\boldsymbol{\varepsilon}_{-1}^{(i)} = \mathbf{0} \ , \ i = 1,2,3..n_k \tag{11a}$$
$$\boldsymbol{\varepsilon}_{0}^{i} = \mathbf{x}_k^{(i)} \ , \ i = 0,1,2..n_k \tag{11b}$$
$$\boldsymbol{\varepsilon}_{2j+1}^{(i)} = \boldsymbol{\varepsilon}_{2j-1}^{(i+1)} + \frac{\mathbf{y}}{(\mathbf{y}\, , \Delta\boldsymbol{\varepsilon}_{2j}^{(i)})} \ , \ i = 0,1,2..n_k \ , \ j = 0,1,2...\frac{n_k}{2} - 1 \text{ where } i + 2j + 1 \le n_k \tag{11c}$$
$$\boldsymbol{\varepsilon}_{2j+2}^{(i)} = \boldsymbol{\varepsilon}_{2j}^{(i+1)} + \frac{\Delta\boldsymbol{\varepsilon}_{2j}^{(i)}}{(\Delta\boldsymbol{\varepsilon}_{2j}^{(i)}\, , \Delta\boldsymbol{\varepsilon}_{2j+1}^{(i)})} \ , \ i = 0,1,2..n_k \ , \ j = 0,1,2...\frac{n_k}{2} - 1 \text{ where } i + 2j + 2 \le n_k \tag{11d}$$

Inverses in the odd terms (11c) are computed with respect to an arbitrary $\mathbf{y}$ such that all terms $\boldsymbol{\varepsilon}_{2j+1}^{(i)}$ exist. Inverses in (11d) are computed with respect to $\Delta\boldsymbol{\varepsilon}_{2j}^{(i)}$.

The result of the topological epsilon algorithm is $\boldsymbol{\varepsilon}_{n_k}^{(0)}$, so the extrapolation function $\mathbf{E}$ is defined as

$$\mathbf{E}(\mathbf{x}_k^{(0)}, \mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)} ... \mathbf{x}_k^{(n_k)}) = \boldsymbol{\varepsilon}_{n_k}^{(0)} .$$

3 Simulation of electrical circuits

Nonlinear dynamical electrical circuits can be described by a system of ordinary differential equations

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathrm{t}) \qquad (12)$$

where $\mathbf{x}(t)$ represents the node voltages and the branch currents of the circuit. In transient analysis, (12) is solved starting from the initial value $\mathbf{x}(0)$. The resulting waveforms are represented by vector functions

$$\mathbf{x}(t) \;,\; 0 \le t \le t_n . \qquad (13)$$

For further reference, $\mathbf{x}(t)$ is a column vector with d components:

$$\mathbf{x}(t) = \left[{}_1x(t), \;\; {}_2x(t), \; \cdots \;\; {}_\mathrm{d}x(t)\right]^\mathrm{T} . \qquad (14)$$

(12) is numerically integrated and solved using the Newton-Raphson iterative metod for times $t = 0 \le t_1 \le t_2 \le \cdots \le t_n$. Time step $\Delta t_i = t_{i+1} - t_i$ should be small enough to avoid numerical errors. Decreasing the time step increases the computation time as more points have to be evaluated.

3.1 SPICE OPUS circuit simulator and optimizer

The simulation of electrical circuits can be done with the SPICE OPUS circuit simulator and optimizer /6/. Circuit simulation is a part of every circuit optimization /7/ where parameters of the circuit are sought, subject to design requirements. During optimization, a circuit is simulated many times, so individual simulations should be as short as possible.

When computing the steady-state response of a circuit, a problem occurs when the period of the steady-state response is much smaller than the largest time constant of the circuit. Such circuits must be simulated for hundreds or even thousands of periods before they reach steady state. To obtain sufficient accuracy, a hundred or more time points must be evaluated per period of the response. Therefore, several million time points have to be evaluated before steady state is reached.

Using the extrapolation algorithms described in Section 2 computing steady-state response of such electrical circuits can be significantly accelerated.

3.2 Application of extrapolation algorithms in circuit simulation

To compute the steady-state response of a circuit by means of an extrapolation algorithms, the circuit is simulated (solving (12)) starting from the initial value $\mathbf{x}(0) = \mathbf{0}$ for times $0 \le t \le t_n + t_{del}$. Terms of sequence (1) are extracted from the resulting waveforms $\mathbf{x}(t)$:

$$\mathbf{x}_0^{(i)} = \mathbf{x}(t_{del} + iT) \;,\; i = 0,1,2..n \;,\; n = \lfloor \mathrm{t_n}/\mathrm{T} \rfloor \qquad (15)$$

where T is the period of the steady-state response of the circuit and t_{del} is the time where the first term $\mathbf{x}_0^{(0)}$ is sampled. Then the extrapolation algorithm is used on sequence $\{\mathbf{x}_0^{(i)}\}$ to generate the extrapolated vector $\mathbf{x}_1^{(0)}$ which represents the initial value $\mathbf{x}(0)$ for a new simulation.

This iterative process is repeated k-times until $\left\| \mathbf{x}_k^{(n)} - \mathbf{x}_k^{(0)} \right\|$ in the last (k-th) iteration is small enough to assume that the circuit is in steady state.

We can assume that steady state has been reached when the relative and the absolute tolerance criteria

$$\left|{}_l x_k^{(n)} - {}_l x_k^{(0)}\right| \le \delta_a + \max\left\{\left|{}_l x_k^{(n)}\right|, \left|{}_l x_k^{(0)}\right|\right\}\delta_r \qquad (16)$$

for all $l = 1,2,\cdots d$ are satisfied.

In practice, it is sufficient if δ_a and δ_r are less than 10^{-5} and 10^{-4}, respectively and greater than the precision of the data representation (i.e. the relative and absolute precision of *double* is 10^{-14} and 10^{-320}, respectively).

A flow chart of the extrapolation algorithm is depicted in Fig. 2.

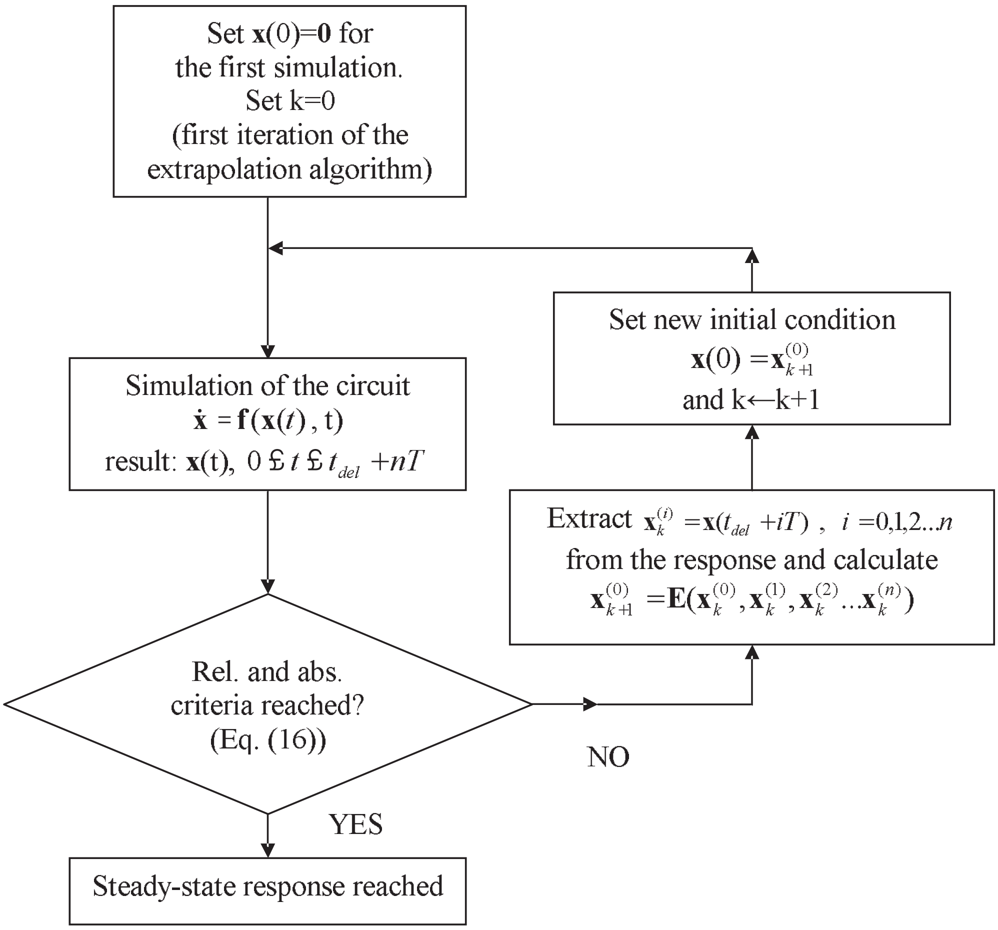


Figure 2: Flow chart of the extrapolation algorithm for determing steady-state response of a circuit.

3.3 Accelerating the computation of steady state of an electrical circuit

The steady-state response is important in the analysis of power conversion circuits, in the evaluation of nonlinear properties of narrow-band circuits, etc. For such circuits the evaluation of long transients can be accelerated using extrapolation algorithms.

In the following section the above mentioned extrapolation algorithms are tested with the Greinacher rectifier, narrow-band filter and switching rectifier test circuits.

4 Comparison

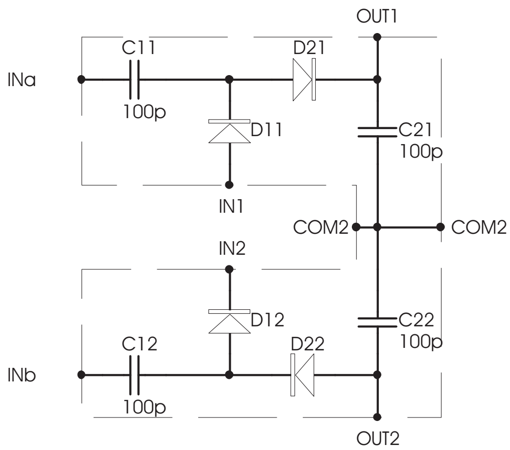
The test circuits were simulated using the direct approach (running a transient analysis until all initial transients died off). It took several thousand periods for the simulator to reach the steady state.

The steady-state responses of the circuits were also computed using the transient analysis accelerated by an extrapolation algorithm (epsilon algorithm, rho algorithm, theta algorithm and topological epsilon algorithm). The total number of periods required for computing the steady state has been greatly reduced.

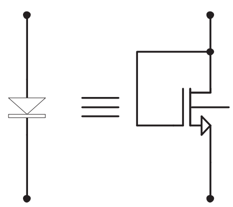
In the following subsections the test circuits and the detailed results of the steady-state simulation are presented.

4.1 Test circuits

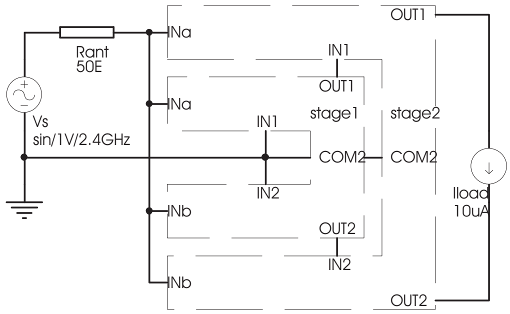
The two-stage Greinacher rectifier test circuit is depicted in Fig. 3.



(a)



(b)



(c)

Figure 3: (a) Single stage of a Greinacher rectifier.
(b) Implementation of a diode with an n-channel MOS.
(c) Test circuit: two-stage Greinacher rectifier.

Fig. 3a represents one stage of the Greinacher rectifier. Diodes in Fig. 3a are implemented as n-channel MOS transistors in 0.18μm TSMC technology (Fig. 3b). Two stages are connected together as depicted in Fig. 3c. At the input of the circuit a sinusoidal voltage source V_s with amplitude 1 V and frequency 2.4 GHz is connected in series with resistor $R_{ant} = 50\mathrm{W}$. The output of the circuit is loaded with $I_{load} = 10\mu A$.

In Fig. 4 a narrow-band filter is depicted. At the input of the circuit (node 1) a sinusoidal voltage source V_{in} with amplitude 1 V and frequency 32767.41 Hz is connected. The output of the circuit is at node 4. The quartz crystal X_1 is modelled with resistor R_s, inductor L_s and capacitors C_s and C_p.

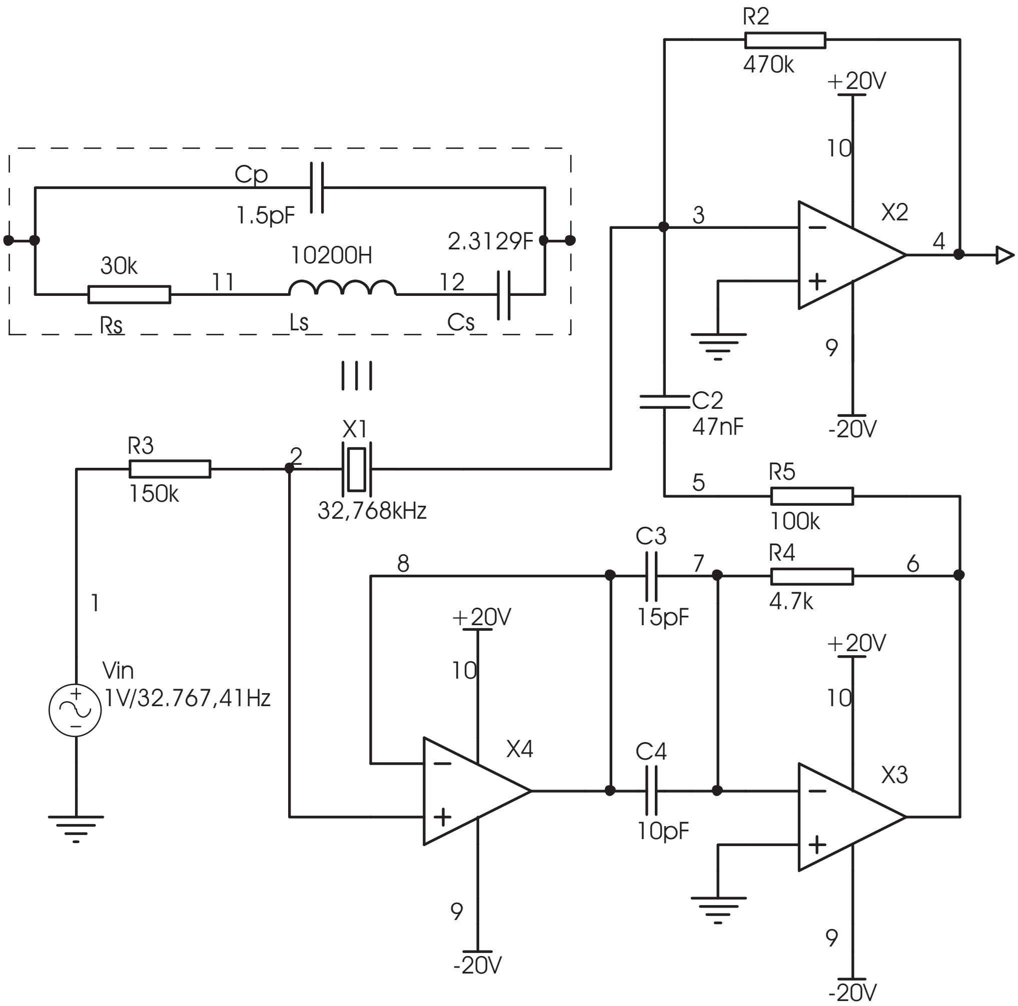


Figure 4: Narrow-band filter.

The circuit in Fig. 5 is a switching power supply. The power source (DC voltage 20 V) is connected to node 1. DC voltage of the V_{pulse} is 10 V, amplitude is 20 V, rise and fall times are $10ns$, and the duty-cycle is 9%. At the output (node 4) $R_{load} = 1k\Omega$ is connected.

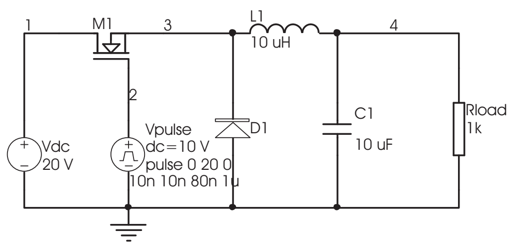


Figure 5: Switching power supply.

4.2 Results

The steady-state response of the Greinacher rectifier is shown in Fig. 6.

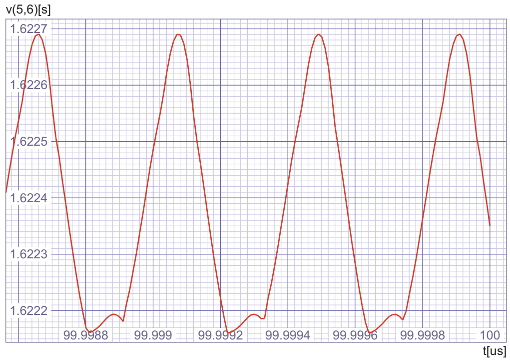


Figure 6: The steady-state response of the two-stage Greinacher rectifer (loaded with $I_{load} = 10\,mA$).

The circuit reaches the steady state in $100\,ms$ or in 240000 periods of the signal V_s. The steady state of the circuit was also computed with extrapolation algorithm. Parameter n_k (equations (4), (7), (8) and (11)) was set to 6. The extrapolation algorithm was stopped when the relative and the absolute criteria (16) were satisfied ($\delta_a = 10^{-6}$ and $\delta_r = 10^{-6}$). The number of iterations of the extrapolation algorithm (k) and the number of periods the circuit was simulated before it reached the steady state ($\sum_k (n_k + t_{del} / T)$) are listed in Table 1.

	Greinacher rectifier		Narrow-band filter		Switching power supply	
	Iterations	**Periods**	**Iterations**	**Periods**	**Iterations**	**Periods**
Epsilon algorithm	52	330	3	54	4	24
Rho algorithm	37	222	3	87	5	30
Theta algorithm	99	594	9	144	12	72
Topological epsilon algorithm	274	1644	3	54	4	24
Direct transient analysis	/	240000	/	65535	/	100000

Table 1: The number of iterations of the extrapolation algorithm and the number of simulated periods in the computation of the steady-state response.

Using an extrapolation algorithm, the steady-state response of the test circuit was computed more than 140 times faster than with the direct approach (transient analysis until all initial transients die off).

The narrow-band filter took 2 s (65535 periods) of the signal V_{in} before it reached steady state (Fig. 7). Computing the steady state with the epsilon, rho, theta and topological epsilon algorithm took 54, 87, 144 and 54 periods, respectively (Table 1) with $d_a = 5\times10^{-6}$ and $d_r = 5\times10^{-6}$.

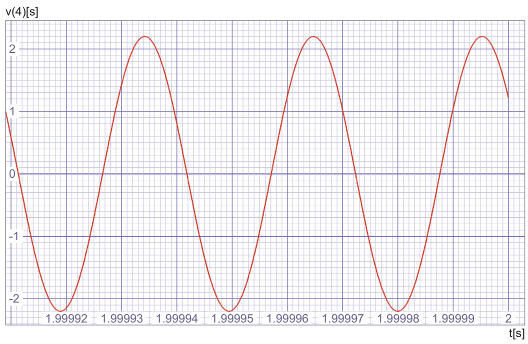


Figure 7: The steady-state response of the narrow-band filter.

The steady-state response of the switching power supply is depicted in Fig. 8. Steady state is reached in 100 ms or in 100000 periods of the signal V_{pulse}. With the epsilon, rho, theta and topological epsilon algorithm computing steady state took 24, 30, 72 and 24 periods, respectively (Table 1) with $d_a = 10^{-6}$ and $d_r = 10^{-7}$.

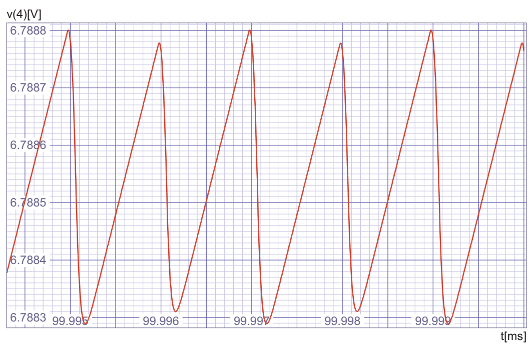


Figure 8: The steady-state response of the switching power supply.

5 Implementation of an extrapolation algorithm in SPICE OPUS

5.1 Choice of the extrapolation algorithm

The results listed in Table 1 show that for the selected test circuits the best of the extrapolation algorithms is the epsilon algorithm. Therefore we chose it for the implementation in SPICE OPUS.

The epsilon algorithm was also chosen due to it's simplicity and good results obtained with other circuits the extrapolation algorithm was tested on /8,9/.

5.2 SSSE analysis

The new *SSSE* (Steady-State Shooting with Extrapolation) analysis was implemented in the SPICE OPUS circuit simulator /6/. The syntax of the *ssse* analysis is as follows:

ssse <freq> [step [skip [periods]]] [history]

< > ... required parameter

[ ] ... optional parameter

freq represents the fundamental frequency (f) of the steady-state response, *step* represents the basis for the variable time step (same as the first argument for the transient analysis, default value: 10/ *freq*), *skip* represents the time skipped (t_{del} in (15)) before the response is sampled (default value: 0) and *periods* gives the number of periods (n_k) that are taken into account for sampling (default value: 2).
The iteration limit of the epsilon extrapolation algorithm is given by the *itl2* option. Convergence (see (16)) is defined with the absolute and the relative tolerances given by products of *sssetol*vntol*, *sssetol*abstol*, and *sssetol*reltol* options. Steady state is reached when the difference (16) is within tolerances.
By default only the last transient run results (steady-state response) are saved in a plot as final results of the ssse analysis. If the *history* flag is set then all transient iterations performed during the steady-state analysis are saved in plots.
The steady-state response of the Greinacher rectifier in Fig. 6 was obtained with the following SPICE OPUS commands.

```
.options itl2=1000

.options sssetol=1e-2

.options abstol=1e-4

.options reltol=1e-4

.control

set xmumult=1

ssse 2.4g 2p 0 6 history

.endc
```

The *xmumult* option is set to 1 to turn off numerical oscillation detection, which can interfere with the extrapolation algorithm.

6 Optimization of electrical circuits
Circuit simulation is a part of every circuit optimization /7/ where parameters of the circuit are sought subject to design requirements. During optimization, a circuit is simulated many times, so individual simulations should be as short as possible.
If the circuit is being optimized and the steady-state response is computed in every iteration of the optimization, fast computation of the steady-state response is a prerequisite for realistic optimization times.

6.1 Optimization of the Greinacher rectifier test circuit
To demonstrate the application of the *ssse* command, an optimization proccess was run on the test circuit in Fig. 3.
Table 2 lists the optimization parameters.

Parameter 0	Width of transistors modelling diodes D11 and D12 (Stage 1)
Parameter 1	Width of transistors modelling diodes D21 and D22 (Stage 1)
Parameter 2	Width of transistors modelling diodes D11 and D12 (Stage 2)
Parameter 3	Width of transistors modelling diodes D21 and D22 (Stage 2)
Parameter 4	Capacitance of all capacitors of Stage 1 and Stage 2

Table 2: The optimization parameters.

The optimization was started at *Initial value* of the parameters in Table 3. The explicit constrains are defined by the columns *Minimum value* and *Maximum value*. The parameters were allowed to change by the value of the *Increment* column in Table 3.

	Initial value	**Minimum value**	**Maximum value**	**Increment**
Parameter 0	10 µm	0.22 µm	100 µm	0.18 µm
Parameter 1	10 µm	0.22 µm	100 µm	0.18 µm
Parameter 2	8 µm	0.22 µm	100 µm	0.18 µm
Parameter 3	8 µm	0.22 µm	100 µm	0.18 µm
Parameter 4	100 pF	10 pF	300 pF	10 pF

Table 3: Initial, minimum, maximum values, and increment of optimization parameters.

The circuit behavior was described with the cost function which consists of penalty functions of measurement values multiplied by weights /10/.
Table 4 shows the measurements, measurements goals, norms and weights.
If a *measurement value* violates a *measurement goal* by *norm*, the contribution to the cost function equals *measurement weight*.

		goal	**weight**	**norm**
Measurement 1	V_{DC_load}	> 2 V	3	1 V
Measurement 2	V_{ripple_load}	< 250 µV	1	100 µV
Measurement 3	V_{DC_noload}	> 3 V	1	1 V
Measurement 4	V_{ripple_noload}	< 140 µV	1	100 µV

Table 4: Measurements performed on the test circuit.

Measurements are defined by

$$V_{DC\_load} = \left\{ \frac{1}{T} \int_{(n_k-1)T}^{n_kT} (v_{OUT2}(t) - v_{OUT1}(t))dt \right\} \Bigg|_{I_{load}=10\,mA} \tag{17a}$$

$$V_{ripple\_load} = \left\{ \max_{t \in [(n_k-1)T, n_kT]} (v_{OUT2}(t) - v_{OUT1}(t)) - \min_{t \in [(n_k-1)T, n_kT]} (v_{OUT2}(t) - v_{OUT1}(t)) \right\} \Bigg|_{I_{load}=10\,mA} \tag{17b}$$

$$V_{DC\_noload} = \left\{ \frac{1}{T} \int_{(n_k-1)T}^{n_kT} (v_{OUT2}(t) - v_{OUT1}(t))dt \right\} \Bigg|_{I_{load}=0} \tag{17c}$$

$$V_{ripple\_noload} = \left\{ \max_{t \in [(n_k-1)T, n_kT]} (v_{OUT2}(t) - v_{OUT1}(t)) - \min_{t \in [(n_k-1)T, n_kT]} (v_{OUT2}(t) - v_{OUT1}(t)) \right\} \Bigg|_{I_{load}=0} . \tag{17d}$$

Measurements 1 and 2 were made with $I_{load} = 10mA$ (Fig. 3c). Measurements 3 and 4 were made with no load ($I_{load} = 0$).

The optimization took 2964 iterations. Every iteration of the optimization included two SSSE analyses.
On an AMD Athlon XP 2500+ (1.83 GHz clock) computer with 512 MB of RAM, the optimization took 3 hours and 10 minutes. Optimum parameters are shown in Table 5 and the final measurement values in Table 6.

	Value
Parameter 0	16.60 µm
Parameter 1	21.10 µm
Parameter 2	27.76 µm
Parameter 3	26.68 µm
Parameter 4	300 pF

Table 5: Optimum parameter values after the optimization.

	Before optimization	**After optimization**	**Improvement**
Measurement 1	1.624 V	1.794 V	10.5 %
Measurement 2	326.0 µV	249.9 µV	23.3 %
Measurement 3	2.670 V	2.911 V	9.0 %
Measurement 4	176.5 µV	138.9 µV	21.3 %

Table 6: Measurement values before and after the optimization.

For Measurements 2 and 4 the goals were achieved, while Measurements 1 and 3 violated the goals. The final cost function value was 0.6792.
Table 6 also lists the measurement values before optimization (at initial values of the parameters in Table 3). Average improvement of the measurements is 16 %.
In Sections 4.2 and 5.1 we have demonstrated that the acceleration ratio for the Greinacher rectifier test circuit (in case of the epsilon algorithm) is 727 ($240000/330 \approx 727$, see Table 1). If ordinary transient analysis was used, the optimization would take more than 95 days. This clearly demonstrates that if the steady-state response of a circuit has to be computed in an optimization loop, an efficient extrapolation algorithm is necessary.

7 Conclusions

Four extrapolation algorithms (epsilon, rho, theta and topological epsilon algorithm) were described in this paper. Using these algorithms the steady-state computation has been accelerated several hundred times.
The extrapolation algorithms were tested on three test circuits (Greinacher rectifier, narrow-band filter and switching power supply). The epsilon algorithm was shown to be the most appropriate for the implementation in the SPICE OPUS circuit simulator.
The steady-state responses of the test circuits were computed several hundred times

faster than with the direct approach (transient analysis until all initial transients die off). A new SPICE OPUS analysis utilizing the epsilon algorithm was implemented.
The Greinacher rectifier test circuit was optimized using the SSSE analysis. The purpose of the optimization was to show that the optimization of the steady-state response can be accelerated significantly by using extrapolation methods. Optimization with the SSSE analysis (accelerated transient analysis) lasted 3 hours and 10 minutes. It has been estimated that the optimization with the direct approach (direct transient analysis) would take more than 95 days.

8 Acknowledgment

The research has been supported by the Slovenian Research Agency within programme P2-0246 - Algorithms and optimization methods in telecommunications.

9 References

/1/ 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, February 1986.

/2/ P. R. Graves-Morris, Extrapolation methods for vector sequences, Numer. Math. 61, pp. 475-487, 1992.

/3/ E. J. Weniger, Nonlinear Sequence Transformations: Computational Tools for the Acceleration of Convergence and the Summantion of Divergent Series, URL:
http://arxiv.org/pdf/math.CA/0107080

/4/ X. Gourdon, P. Sebah, Convergence acceleration of series, January 10, 2002, URL:
http://numbers.computation.free.fr/Constants/Miscellaneous/seriesacceleration.html

/5/ K. S. Kundert, J. K. White, A. Sangiovanni-Vincentelli, Steady-state methods for simulation analog and microwave circuits, Kluwer Academic Publishers, 1990.

/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. Puhan, T. Tuma, Optimization of analog circuits with SPICE 3f4, Proceedings of the ECCTD '97, vol. 1, pp. 177-180, 1997.

/8/ B. Wagner, A. Bürmen, J. Puhan, I. Fajfar, T. Tuma, Computing the steady-state response of nonlinear circuits by means of the ε-algorithm, Electrotechnical review, vol. 72, no. 5, pp. 297-302, 2005.

/9/ B. Wagner, A. Bürmen, J. Puhan, I. Fajfar, T. Tuma, Uporaba ekstrapolacijskih metod pri računanju stacionarnega stanja električnih vezij (Application of extrapolation methods for steady-state computation of electrical circuits), Zbornik štirinajste mednarodne Elektrotehniške in računalniške konference ERK 2005, zv. A, str. 82-85.

/10/ BÜRMEN, Arpad, STRLE, Drago, BRATKOVIČ, Franc, PUHAN, Janez, FAJFAR, Iztok, TUMA, Tadej. Automated robust design and optimization of integrated circuits by means of penalty functions. AEÜ, Int. j. electron. commun. 2003, 57, no. 1, str. 47-56.

univ. dipl. ing. el. Borut Wagner
University of Ljubljana
Faculty of Electrical Engineering
Tržaška cesta 25, SI-1000 Ljubljana, Slovenia
E-mail: borut.wagner@fe.uni-lj.si
Tel: +386 1 4768724
Fax: +386 1 4264630

doc. dr. Árpád Bűrmen
University of Ljubljana
Faculty of Electrical Engineering
Tržaška cesta 25, SI-1000 Ljubljana, Slovenia
E-mail: arpadb@fides.fe.uni-lj.si
Tel: +386 1 4768322
Fax: +386 1 4264630

doc. dr. Janez Puhan
University of Ljubljana
Faculty of Electrical Engineering
Tržaška cesta 25, SI-1000 Ljubljana, Slovenia
E-mail: janez.puhan@fe.uni-lj.si
Tel: +386 1 4768322
Fax: +386 1 4264630

prof. dr. Sašo Tomažič
University of Ljubljana
Faculty of Electrical Engineering
Tržaška cesta 25, SI-1000 Ljubljana, Slovenia
E-mail: saso.tomazic@fe.uni-lj.si
Tel: +386 1 4768432
Fax: +386 1 4264630

prof. dr. Tadej Tuma
University of Ljubljana
Faculty of Electrical Engineering
Tržaška cesta 25, SI-1000 Ljubljana, Slovenia
E-mail: tadej.tuma@fe.uni-lj.si
Tel: +386 1 4768329
Fax: +386 1 4264630

7.5 Članek ERK 2006

Optimizacija dvostopenjskega Greinacherjevega polnovalnega usmernika

Borut Wagner, Árpád Bűrmen, Janez Puhan, Sašo Tomažič in Tadej Tuma
Fakulteta za elektrotehniko
Univerza v Ljubljani
Tržaška cesta 25, SI-1000 Ljubljana, Slovenija
E-pošta: borut.wagner@fe.uni-lj.si

Optimization of the two-stage Greinacher full-wave rectifier
In this paper we present optimization of the Greinacher rectifier with SPICE OPUS circuit simulator. The rectifier is also used on wireless power transmission in radio frequency identification systems (RFID). Four optimization goals were used in optimization: DC voltage and ripple at output with and without load. Two steady-state analyses had to be computed within each iteration of the optimization loop. Each steady-state analysis took 23 minutes. For several thousands of iterations, optimization would take between ten and hundred days. For this reason the steady-state response was computed by means the extrapolation methods for transient analysis acceleration. For the Greinacher rectifer the steady-state response was computed 700-times faster than by means of the direct approach (transient analysis until all initial transients die off). The optimization took 2964 iterations and was finished in 3 hours and 10 minutes. Average improvement of the optimization goals was 16 %.

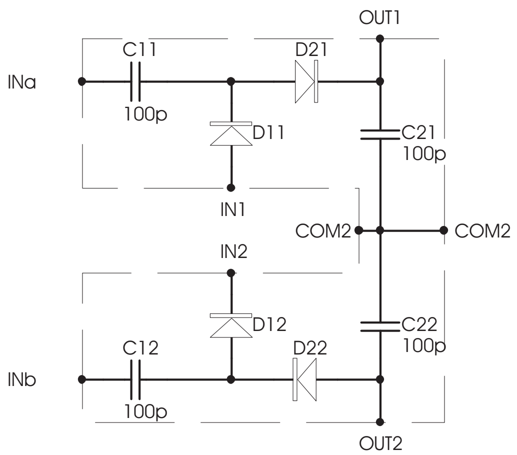
1 Uvod

V članku je predstavljena optimizacija dvostopenjskega Greinacherjevega usmernika [1], ki se med drugim uporablja pri prenosu energije npr. za napajanje sistemov za radijsko identifikacijo (RFID - Radio Frequency Identification Systems). Prikazan je primer takšnega usmerniškega vezja s privzetimi vrednostmi parametrov elementov. Na podlagi teh vrednosti je izračunan stacionarni odziv vezja pri določenem vzbujanju na vhodu in z določenim bremenom na izhodu.

V tretjem razdelku je dvostopenjsko usmerniško vezje optimizirano glede na postavljene optimizacijske zahteve. Težava, ki se pri optimizaciji vezja pojavi, je dolgotrajnost izračuna stacionarnega stanja vezja, ki nastopa znotraj vsake iteracije optimizacijske zanke. Z uporabo algoritmov za pohitritev izračuna stacionarnega stanja se da pospešiti samo analizo kot tudi celoten optimizacijski postopek. Na koncu članka je ocenjen še čas optimizacije, ki bi bil potreben, če bi za izračun stacionarnega stanja uporabili direktno tranzientno analizo namesto ekstrapolacijskih algoritmov.

2 Greinacherjev usmernik

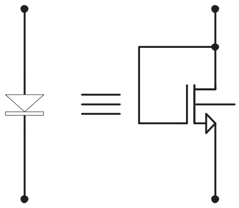
Na sliki 1 je prikazana ena stopnja Greinacherjevega usmernika. Vse diode so realizirane z uporabo n-kanalnih MOS tranzistorjev (slika 2). Uporabljena je bila 0,18 μm tehnologija TSMC.



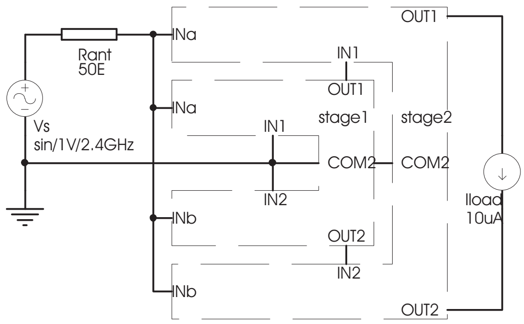
Slika 1: Ena stopnja Greinacherjevega usmernika.
Figure 1: Single stage of a Greinacher rectifier.

Analizirano je bilo dvostopenjsko Greinacherjevo usmerniško vezje (slika 3). Širine in dolžine tranzistorjev, ki modelirajo diode D_{11}, D_{12}, D_{21} in D_{22} v obeh stopnjah usmernika, so prikazane v tabeli 1.

Na vhod vezja je priključen sinusni napetostni vir frekvence 2,4 GHz in amplitude 1 V. Upor R_{ant} predstavlja impedanco antene in znaša 50 Ω. Na izhod je



Slika 2: Implementacija diode z n-kanalnim MOS tranzistorjem.
Figure 2: Implementation of a diode with an n-channel MOS.



Slika 3: Dvostopenjsko Greinacherjevo usmerniško vezje.
Figure 3: Two-stage Greinacher rectifier.

kot breme priključen enosmerni tokovni vir z vrednostjo 10 μA.

2.1 Stacionarni odziv

Vezje na sliki 3 je bilo analizirano s privzetimi vrednostmi parametrov (tabela 1). Vrednosti vseh kondenzatorjev obeh stopenj so bile 100 pF. Izračunan je bil odziv vezja v stacionarnem stanju (slika 4).

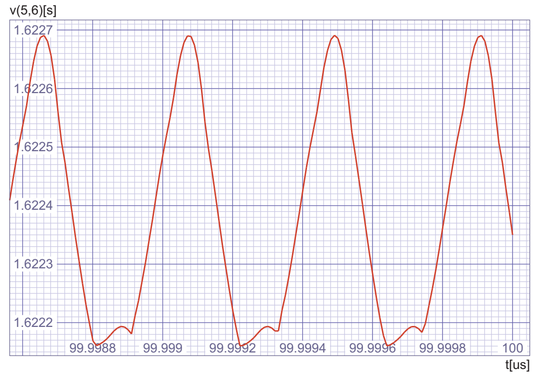
Simulacijski čas stacionarnega odziva je bil 100 μs oziroma 240.000 period vhodnega signala V_s. Na osebnem računalniku AMD Athlon XP 2500+ (frekvenca ure 1,83 GHz) s 512 MB RAMa je izračun trajal 23 minut. V nadaljevanju bodo na kratko predstavljeni algoritmi, s katerimi lahko hitreje izračunamo stacionarno stanje. Z uporabo teh algoritmov tako postane optimizacija izvedljiva v doglednem času.

2.2 Pohitritev izračuna stacionarnega odziva

Uporaba ekstrapolacijskih metod [2, 3] se je izkazala za uspešno pri pospeševanju izračuna stacionarnega stanja [4, 5]. Z uporabo algoritma epsilon lahko

	Širina	Dolžina
D_{11} in D_{12}, stopnja 1	10 μm	0,5 μm
D_{21} in D_{22}, stopnja 1	10 μm	0,5 μm
D_{11} in D_{12}, stopnja 2	8 μm	0,5 μm
D_{21} in D_{22}, stopnja 2	8 μm	0,5 μm

Tabela 1: Začetne širine in dolžine diod (modeliranih z MOS tranzistorji) dvostopenjskega Greinacherjevega usmernika.
Table 1: Initial widths and lengths of the diodes (modelled with MOS transistors) of the two-stage Greinacher rectifier.



Slika 4: Odziv vezja (slika 3) v stacionarnem stanju.
Figure 4: The steady-state response of the circuit in figure 3.

izračunamo odziv vezja na sliki 3 v stacionarnem stanju 700-krat hitreje [5].

V simulator električnih vezij SPICE OPUS [6, 8] je bila vgrajena dodatna analiza *SSSE* za izračun stacionarnega stanja z uporabo ekstrapolacijske metode [5].

3 Optimizacija

Vezje na sliki 3 je bilo optimizirano s programskim paketom SPICE OPUS [7, 8, 9] z uporabo metode simpleksov.

Znotraj optimizacijske zanke je bila uporabljena analiza *SSSE*, saj bi pri uporabi direktne tranzientne analize za izračun stacionarnega stanja optimizacija trajala od deset do sto dni (odvisno od zahtev in s tem števila iteracij optimizacijske zanke).

3.1 Optimizacijski parametri

V tabeli 2 je opisanih pet optimizacijskih parametrov.

3.1 Optimizacijski parametri

V tabeli 2 je opisanih pet optimizacijskih parametrov.

Parameter 0	Širina tranzistorjev, modeliranih z diodama D11 in D12 (stopnja 1)
Parameter 1	Širina tranzistorjev, modeliranih z diodama D21 in D22 (stopnja 1)
Parameter 2	Širina tranzistorjev, modeliranih z diodama D11 in D12 (stopnja 2)
Parameter 3	Širina tranzistorjev, modeliranih z diodama D21 in D22 (stopnja 2)
Parameter 4	Kapacitivnosti vseh kondenzatorjev (stopnja 1 in 2)

Tabela 2: Parametri optimizacije.
Table 2: The optimization parameters.

Znotraj optimizacijske zanke so se lahko optimizacijski patametri spreminjali v okviru eksplicitnih omejitev, ki so podane v tabeli 3.

Opt. par.	Začetna vrednost	Min. vrednost	Maks. vrednost	korak
0	10 μm	0,22 μm	100 μm	0,18 μm
1	10 μm	0,22 μm	100 μm	0,18 μm
2	8 μm	0,22 μm	100 μm	0,18 μm
3	8 μm	0,22 μm	100 μm	0,18 μm
4	100 pF	10 pF	300 pF	10 pF

Tabela 3: Začetne, minimalne, maksimalne vrednosti in korak optimizacijskih parametrov.
Table 3: Initial, minimum, maximum values, and increment of optimization parameters.

3.2 Optimizacijske zahteve

Optimizacija je potekala na podlagi optimizacijskih zahtev, ki so bile definirane preko meritev vezja [10]. Zahtevana je bila določena enosmerna izhodna napetost in omejenost valovitosti pri bremenu na izhodu in brez njega. Na osnovi meritev vezja, zahtev za posamezno meritev, uteži in norme meritve je tvorjena kriterijska funkcija, za katero postopek optimizacije poišče minimalno vrednost na določenem območju. V tabeli 4 so zbrane meritve vezja, zahteve, uteži in norme meritev.

Meritvi 1 in 2 sta bili izvedeni pri bremenu (kot je prikazano na sliki 3). Pri meritvah 3 in 4 je bilo breme (tokovni vir I_{load}) odstranjeno ($I_{load} = 0$). Meritve so definirane v (1)-(4).

Št. mer.	meritev	zahteva	utež	norma
1	V_{DC_load}	> 2 V	3	1 V
2	V_{ripple_load}	< 250 μV	1	100 μV
3	V_{DC_noload}	> 3 V	1	1 V
4	V_{ripple_noload}	< 140 μV	1	100 μV

Tabela 4: Meritve, ki so bile izvedene na vezju.
Table 4: Measurements performed on the circuit.

$$V_{DC\_load} = \left[\frac{1}{T} \int_T (v_{OUT2}(t) - v_{OUT1}(t)) dt \right]_{I_{load}=10\mu A} \quad (1)$$

$$V_{ripple\_load} = \\ = \max_T \left[(v_{OUT2}(t) - v_{OUT1}(t)) \right]_{I_{load}=10\mu A} - \\ - \min_T \left[(v_{OUT2}(t) - v_{OUT1}(t)) \right]_{I_{load}=10\mu A} \quad (2)$$

$$V_{DC\_noload} = \left[\frac{1}{T} \int_T (v_{OUT2}(t) - v_{OUT1}(t)) dt \right]_{I_{load}=0} \quad (3)$$

$$V_{ripple\_noload} = \\ = \max_T \left[(v_{OUT2}(t) - v_{OUT1}(t)) \right]_{I_{load}=0} - \\ - \min_T \left[(v_{OUT2}(t) - v_{OUT1}(t)) \right]_{I_{load}=0} \quad (4)$$

V meritvah uporabljena integrala, maksimalni in minimalni vrednosti so izračunani na intervalu dolžine $T = 417\ ps$, kolikor znaša perioda vhodnega signala oz. napetostnega vira V_s.

3.3 Rezultat optimizacije

Z opisano postavitvijo optimizacije smo v 2964 iteracijah optimizacijske zanke dosegli povprečno 16% izboljšanje meritev glede na začetne vrednosti. Tabela 5 prikazuje vrednosti meritev pred in po optimizaciji.

Iz tabel 4 in 5 je razvidno, da zahtevi za meritvi 1 in 3 nista bili izpolnjeni, kljub temu pa sta se vrednosti teh dveh meritev izboljšali za 10,5 % in 9,0 %. Zaradi tega vrednost kriterijske funkcije na koncu ni bila 0 (to bi se zgodilo v primeru, če bi bile izpolnjene zahteve vseh meritev), ampak je imela vrednost 0,6792.

Številka meritve	pred optimizacijo	po optimizaciji	izboljšanje
1	1,624 V	1,794 V	10,5 %
2	326,0 μV	249,9 μV	23,3 %
3	2,670 V	2,911 V	9,0 %
4	176,5 V	138,8 μV	21,3 %

Tabela 5: Vrednosti meritev pred in po optimizaciji.
Table 5: Measurement values before and after the optimization.

3.4 Primerjava z direktnim izračunom stacionarnega stanja

Izračun stacionarnega stanja dvostopenjskega Greinacherjevega usmernika na uporabljenem računalniku traja 23 minut. Ker znotraj optimizacijske zanke potrebujemo dva izračuna stacionarnega stanja (obremenjen in neobremenjen izhod), bi trajalo 2964 itreacij optimizacije približno 95 dni. Z uporabo hitrejšega račnalnika oz. več računalnikov hkrati bi sicer lahko ta čas skrajšali, vendar lahko izračun stacionarnega stanja pohitrimo že z uporabo ekstrapolacijskih metod, ki so bile predstavljene v prejšnjih razdelkih. Z uporabo le-teh smo vezje optimizirali v nekaj več kot treh urah.

4 Zaključek

V članku smo prikazali optimizacijo dvostopenjskega Greinacherjevega usmernika. Vezje smo optimizirali glede na postavljene optimizacijske zahteve. Znotraj optimizacijske zanke je bil potreben izračun stacionarnega stanja, ki je za to vezje zelo dolgotrajen. Za izračun stacionarnega stanja je bil uporabljen ekstrapolacijski algoritem, ki analizo pohitri do te mere, da je optimizacijski postopek zaključen v doglednem času. Rezultat optimizacije vezja je povprečno 16% izboljšanje meritev.

5 Zahvala

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

Literatura

[1] Jari-Pascal Curty, Norbert Joehl, François Krummenacher, Catherine Dehollain, Michel J. Declercq, "A Model for μ-Power Rectifier Analysis and Design" *IEEE Transactions on Circuits and SystemsI: Regular Papers,* vol. 52, no. 12, december 2005.

[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] K. S. Kundert, J. K. White, A. Sangiovanni-Vincentelli, *"Steady-state methods for simulation analog and microwave circuits"*, Kluwer Academic Publishers, 1990.

[4] B. Wagner, Á. Bűrmen, J. Puhan, I. Fajfar, T. Tuma, "Computing the steady-state response of nonlinear circuits by means of ϵ-algorithm" *Elektrotehniški vestnik*, vol. 72 (5):297–302, Ljubljana, Slovenija, 2005.

[5] B. Wagner, Á. Bűrmen, J. Puhan, S. Tomažič, T. Tuma, "Application of extrapolation algorithms in nonlinear circuit simulation and optimization with SPICE OPUS", *Midem informacije*, oddano 21.04.2006.

[6] T. Quarles, A. R. Newton, D. O. Pederson, A. Sangiovanni-Vincentelli, "SPICE3 Version 3f4 User's Manual", University of California, Berkeley, California, 1989.

[7] J. Puhan, T. Tuma, I. Fajfar, "SPICE for Windows 95/98/NT", *Electrotechnical Review*, vol. 65, no. 5, pp. 267-271, Ljubljana, Slovenija, 1998.

[8] SPICE OPUS simulator vezij, domača stran:
URL: http://www.fe.uni-lj.si/spice/
Fakulteta za elektrotehniko, Laboratorij za računalniško načrtovanje vezij:
URL: http://www.fe.uni-lj.si/edalab/

[9] J. Puhan, T. Tuma, "Optimization of analog circuits with SPICE 3f4", *Proceedings of the ECCTD '97*, vol. 1, pp. 177-180, 1997.

[10] BŰRMEN, Arpad, STRLE, Drago, BRATKOVIČ, Franc, PUHAN, Janez, FAJFAR, Iztok, TUMA, Tadej. "Automated robust design and optimization of integrated circuits by means of penalty functions." *AEÜ 2003, 57, no. 1, str. 47-56.*

Literatura

[1] J. Puhan, T. Tuma, I. Fajfar, *SPICE for Windows 95/98/NT*, Elektroteh. vestn., Vol. 65, No. 5, str. 267–271, 1998.

[2] T. Quarles, A. R. Newton, D. O. Pederson, A. Sangiovanni-Vincentelli, *SPICE3 version 3f4 user's manual*, University of California, Berkeley, CA, ZDA, 1989.

[3] *Simulator vezij SPICE OPUS, domača stran,*
URL: http://www.fe.uni-lj.si/spice/
Fakulteta za elektrotehniko, laboratorij za računalniške metode v elektroniki,
URL: http://www.fe.uni-lj.si/edalab/

[4] L. O. Chua, P. M. Lin, *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*, Prentice-Hall, Englewood Cliffs, New Jersey, ZDA, 1975.

[5] T. Tuma, F. Bratkovič, *A general approach to circuit equations*, Int. j. circuit theory appl., Vol. 22, No. 6, str. 431-445, 1994.

[6] G. A Korn, T. M. Korn, *Mathematical handbook for scientists and engineers*, McGraw-Hill Book Company, 1961.

[7] T. Tuma, *Analiza vezij s programom SPICE3*, Fakulteta za elektrotehniko, Ljubljana, 1997.

[8] Á. Bűrmen, *Uvod v programski paket SPICE OPUS*, Fakulteta za elektrotehniko, Ljubljana, 2006.

[9] Á. Bűrmen, *Asinhroni vzporedni direktni optimizacijski postopki za analogna elektronska vezja*, Doktorska disertacija, Fakulteta za elektrotehniko, Univerza v Ljubljani, 2003.

[10] J. Puhan, T. Tuma , *Optimization of analog circuits with SPICE 3f4*, Proceedings of the ECCTD '97, Vol. 1, str. 177–180, 1997.

[11] M. H. WRIGHT, *Direct search methods: once scorned, now respectable,* in Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis, D. F. Griffiths, G. A. Watson, eds., Addison Wesley Longman, str. 191–208, 1996.

[12] M. J. D. POWELL, *Direct search algorithms for optimization calculations*, Acta Numerica, Vol. 7, str. 287–336, 1998.

[13] R. M. LEWIS, V. J. TORCZON, M. W. TROSSET, *Direct search methods: Then and now*, Journal of Computational and Applied Mathematics, Vol. 124, No. 1-2, str. 191–207, 2000.

[14] J. PUHAN, Á. BŰRMEN, T. TUMA, *Heuristic Approach to Circuit Sizing Problem*, Informacije MIDEM - Journal of Microelectronics, Electronic Componets and Materials, Vol. 33, No. 3, str. 149–156, 2003.

[15] J. PUHAN, T. TUMA, I. FAJFAR, *Optimisation Methods in SPICE, a Comparison*, Proceedings of the ECCTD'99, Vol. 1, str. 1279-1282, 1999.

[16] R. HOOKE, T. A. JEEVES, *'direct search' solution of numerical and statistical problems*, Journal of the Association for Computing Machinery, Vol. 8, No. 2, str. 212–229, april 1961.

[17] A. R. CONN, N. I. M. GOULD, P. L. TOINT *Trust-Region Methods*, SIAM, Philadelphia, PA, ZDA, 2000.

[18] J. A. NELDER, R. MEAD, *A simplex method for function minimization*, The computer Journal, Vol. 7, str. 308–313, 1965.

[19] K. I. M. MCKINNON, *Convergence of the Nelder-Mead simplex method to a nonstationary point*, SIAM Journal on Optimization, Vol. 9, No. 1, str. 148–158, oktober/december 1998.

[20] D. BYATT, *A convergent variant of the Nelder-Mead algorithm*, magistrska naloga, Mathematics and Statistics Department, University of Canterbury, Christchurch, NZ, 2000.

[21] C. J. PRICE, I. D. COOPE, D. BYATT, *A convergent variant of the Nelder-Mead algorithm*, Journal of Optimization Theory and Applications, Vol. 113, No. 1, str. 5–19, april 2002.

[22] J. PUHAN, *Optimizacija vezij v programskem okolju SPICE*, Doktorska disertacija, Fakulteta za elektrotehniko, Univerza v Ljubljani, 2000.

[23] Á. BŰRMEN, I. FAJFAR, J. PUHAN, A. NUSSDORFER, T. TUMA, *Optimizacija elektronskih vezij z vzporednim omejenim simpleksnim postopkom*, Elektroteh. vestn., Vol. 69, No. 1, str. 7–12, 2002.

[24] K. S. KUNDERT, A. SANGIOVANNI-VINVENTELLI, *Simulation of Nonlinear Circuits in the Frequency Domain*, IEEE Transactions on Computer-Aided Design, Vol. CAD-5, No. 4, str. 521–535, oktober 1986.

[25] K. S. KUNDERT, G. B. SORKIN, A. SANGIOVANNI-VINCENTELLI, *Applying Harmonic Balance to Almost-Periodic Circuits*, IEEE Transactions on Microwave Theory and Techniques, Vol. 36, No. 2, str. 366–378, februar 1988.

[26] K. S. KUNDERT, J. K. WHITE, A. SANGIOVANNI-VINCENTELLI, *Steady-state methods for simulation analog and microwave circuits*, Kluwer Academic Publishers, Boston, Dordrecht, London, 1990.

[27] K. S. KUNDERT, *Introduction to RF Simulation and Its Application*, IEEE Journal of Solid-State Circuits, Vol. 34, No. 9, str. 1298–1319, september 1999.

[28] N. SOVEIKO, M. NAKHLA, *Wavelet Harmonic Balance*, IEEE Microwave and Wireless Components Letters, Vol. 13, No. 6, str. 232–234, junij 2003.

[29] N. SOVEIKO, M. S. NAKHLA, *Steady-State Analysis of Multitone Nonlinear Circuits in Wavelet Domain*, IEEE Transactios on Microwave Theory and Techniques, Vol. 52, No. 3, str. 785-797, marec 2004.

[30] C. R. CHANG, P. L. HERON AND M. B. STEER, *Harmonic Balance and Frequency-Domain Simulation of Nonlinear Microwave Circuits Using the Block Newton Method*, IEEE Transactions on Microwave Theory and Techniques, Vol. 38, No. 4, str. 431-434, april 1990.

[31] G. W. RHYNE, M. B. STEER AND B. D. BATES, *Frequency-Domain Nonlinear Circuit Analysis Using Generalized Power Series*, IEEE Transactions on Microwave Theory and Techniques, Vol. 36, No. 2, str. 379-387, februar 1988.

[32] K. KUNDERT, J. WHITE, A. SANGIOVANNI-VINCENTELLI, *A Mixed Frequency-Time Approach for Finding the Steady-State Solution of Clocked Analog Circuits*, Proceedings of the Custom Integrated Circuits Conference, str. 6.2.1–6.2.4, marec 1988.

[33] K. S. KUNDERT, J. WHITE, A. SANGIOVANNI-VINCENTELLI, *A Mixed Frequency-Time Approach for Distortion Analysis of Switching Filter Circuits*, IEEE Journal of Solid-State Circuits, Vol. 24, No. 2, str. 443–451, april 1989.

[34] J. ROOS, *Frequency-domain analysis of nonlinear circuits using Chebyshev polynomials*, Magistrska naloga, Helsinki University of Technology, 1994.

[35] T. J. APRILLE, JR, T. N. TRICK, *Steady-State Analysis of Nonlinear Circuits with Periodic Inputs*, Proceedings of the IEEE, Vol. 60, No. 1, str. 108–114, januar 1972.

[36] A. USHIDA, L. O. CHUA, *Frequency-Domain Analysis of Nonlinear Circuirs Driven by Multi-Tone Signals*, IEEE Transactions on Circuits and Systems, Vol. CAS-31, No. 9, str. 766–779, september 1984.

[37] P. L. HERON, M. B. STEER, *Jacobian Calculation Using the Multidimensional Fast Fourier Transformation in the Harmonic Balance Analysis on Nonlinear Circuits*, IEEE Transactions on Microwave Theory an Techniques, Vol. 38, No. 4, str. 429–431, april 1990.

[38] V. RIZZOLI, C. CECCHETTI, A. LIPPARINI, F. MASTRI, *General-Purpose Harmonic Balance Analysis of Nonlinear Microwave Circuits Under Multitone Excitation*, IEEE Transactions on Microwave Theory an Techniques, Vol. 36, No. 12, str. 1650–1660, december 1988.

[39] V. RIZZOLI, A. NERI, *State of the Art and Present trends in Nonlinear Microwave CAD Techniques*, IEEE Transactions on Microwave Theory an Techniques, Vol. 36, No. 2, str. 343–365, februar 1988.

[40] M. S. NAKHLA, J. VLACH, *A Piecewise Harmonic Balance Technique for Determination of Periodic Response of Nonlinear Systems*, IEEE Transactions on Circuis and Systems, Vol. CAS-23, No. 2, str. 85–91, februar 1976.

[41] B. TROYANOVSKY, *Frequency Domain Algorithms For Simulating Large Signal Distortion in Semiconductor Devices*, Doktorska disertacija, Stanford University, Department of Electrical Engineering, november 1997.

[42] J. C. PEDRO, N. B. CARVALHO, *A Mixed-Mode Simulation Technique for the Analysis of RF Circuits Driven By Modulated Signals*, ConfTele 2001 (III Conferência de Telecomunicaçõles), 23. april 2001.

[43] F. KRIŽANIČ, *Linearna algebra in linearna analiza*, DZS, Ljubljana, 1993.

[44] I. VIDAV, *Višja matematika*, DZS, Ljubljana, 1973.

[45] B. WAGNER, J. PUHAN, Á. BŰRMEN, I. FAJFAR, T. TUMA, *Simulacija in optimizacija analognih električnih vezij s pomočjo ravnotežne harmonske analize*, zbornik trinajste mednarodne Elektrotehniške in računalniške konference ERK 2004, zvezek A, (urednik: Baldomir Zajc), str. 29–32, 2004.

[46] D. A. SMITH, W. F. FORD, A. SIDI, *Extrapolation methods for vector sequences*, SIAM Review, Vol. 29, No. 2, str. 199–233, junij 1987.

[47] 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, str. 161–175, marec 1980.

[48] P. WYNN, *On a Device for Computing the $e_m(S_n)$ Transformation*, Mathematical tables and aids to computation (MTAC), Vol. 10, str. 91–96, 1956.

[49] A. SIDI, W. F. FORD, D. A. SMITH, *Acceleration of Convergence of Vector Sequences*, SIAM Journal on Numerical Analysis, Vol. 23, No. 1, str. 178–196, februar 1986.

[50] P. R. GRAVES-MORRIS, *Extrapolation methods for vector sequences*, Numer. Math., Vol. 61, No. 1, str. 475–487, december 1992.

[51] A. SIDI, *Practical Extrapolation Methods: theory and Applications*, Cambridge University Press, Cambridge, 2003.

[52] A. SALAM, P. R. GRAVES-MORRIS, *On the vector ϵ-algorithm for solving linear systems of equations*, Numerical Algorithms, Vol. 29, No. 1–3, str. 229–247, marec 2002.

[53] E. J. WENIGER, *Nonlinear Sequence Transformations*, Computational Tools for the Acceleration of Convergenc and the Summantion of Divergent Series,
URL: http://arxiv.org/pdf/math.CA/0107080

[54] P. SEBAH, X. GOURDON, *Convergence acceleration of series*, January 10, 2002, URL: http://numbers.computation.free.fr/Constants/Miscellaneous/seriesacceleration.html

[55] C. Brezinski, *A General Extrapolation Algorithm*, Numer. Math., Vol. 35, No. 2, str. 175-187, junij 1980.

[56] S. H. M. J. Houben, J. M. L. Maubach, R. M. M. Mattheij, *An accelerated Poincaré-map method for autonomous oscillators*, Applied Mathematics and Computation, Vol. 140, No. 2-3, str. 191–216, avgust 2003.

[57] J. P. Delahaye, *Automatic Selection of Sequence Transformations*, Mathematics of Computation, Vol. 37, No. 155, str. 197–204, julij 1981.

[58] J. P. Delahaye, *Sequence Transformations*, Springer-Verlag Berlin, 1988.

[59] K. R. Whight, *Large signal periodic time-domain circuit simulation*, IEEE Proc Circuits Devices Syst, Vol. 141, No. 4, str. 285–291, avgust 1994.

[60] K. R. Whight, P. A. Gough, P. Walker, *Large signal periodic time-domain simulation*, International Journal of Numerical Modelling: Electronic Network, Devices and Fields, Vol. 5, str. 11–21, 1992.

[61] J. A. Steele, A. T. Dolovich, *Toward the kernel of the vector epsilon algorithm*, Int. J. Numer. Meth. Engng, Vol. 48, No. 5, str. 721–730, 2000.

[62] J. Roychowdhury, *Analyzing Circuits with Widley Separated Time Scales Using Numerical PDE Methods*, IEEE Transactions on Circuits and Systems–I, Vol. 48, No. 5, str. 578–594, maj 2001.

[63] J. Roychowdhury, *A Time-domain RF Steady-State Method for Closely Spaced Tones*, Proceedings of the 39th conference on Design automation, New Orleans, Louisiana, USA, str. 510–513, 2002.

[64] L. Gyergyék, *Teorija signalov in obdelava signalov*, Fakulteta za elektrotehniko in računalništvo, Ljubljana, 1991.

[65] P. Vizmuller, *RF design guide: systems, circuits and equations*, Artech House, Boston, London, 1995.

[66] M. S. Nakhla, F. H. Branin, *Determining the periodic response of nonlinear systems by a gradient method*, Circuit Theory and Application, Vol. 5, str. 255–273, 1977.

[67] H. Jokinen, M. Valtonen, *Steady-State Time-Domain Analysis Including Frequency-Dependent Components*, CT-24, ISBN 951-22-2663-4, ISSN 0784-5979, junij 1995.

[68] R. MANCINI, *Op Amps For Everyone*, Texas Instrument Design Reference, avgust 2002.

[69] ROLF SCHAUMANN, MAC E. VAN VALKENBURG, *Design of analog filters*, Oxford University Press, New York, Oxford, 2001.

[70] E. V. IVANOV, *Bandpass crystal filter for enhanced phase-sensitive detection*, Meas. Sci. Technol., Vol. 10, No. 7, str. N77-N81, julij 1999.

[71] A. I. PRESSMAN, *Switching Power Supply Design*, Second Edition, McGraw-Hill, New York, 1998.

[72] J. P. CURTY, N. JOEHL, F. KRUMMENACHER, C. DEHOLLAIN, M. J. DECLERCQ, *A Model for μ-Power Rectifier Analysis and Design*, IEEE Transactions on Circuits and Systems—I: Regular Papers, Vol. 52, No. 12, str. 2771–2779, december 2005.

[73] J. O. MCSPADDEN, L. FAN, K. CHANG, *Design and Experiments of a High-Conversion-Efficiency 5.8-GHz Rectenna*, IEEE Transactions on Microwave Theory and Techniques, Vol. 46, No. 12, str. 2053–2060, december 1998.

[74] Y.-H. SUH, K. CHANG, *A High-Efficiency Dual-Frequency Rectenna for 2.45- and 5.8-GHz Wireless Power Transmission*, IEEE Transactions on Microwave Theory and Techniques, Vol. 50, No. 7, str. 1784–1789, julij 2002.

[75] T.-W. YOO, K. CHANG, *Theoretical and Experimental Development of 10 and 35 GHz Rectennas*, IEEE Transactions on Microwave Theory and Techniques, Vol. 40, No. 6, str. 1259–1266, junij 1992.

[76] THOMAS H. LEE, *The Design of CMOS Radio-Frequency Integrated Circuits*, Cambridge University Press, Cambridge, UK, New York, 2001.

[77] Á. BŰRMEN, D. STRLE, F. BRATKOVIČ, J. PUHAN, I. FAJFAR, T. TUMA, *Automated Robust Design and Optimization of Integrated Circuits by Means of Penalty Functions*, International Journal of Electronics and Communications, Vol. 57, No. 1, str. 47–56, januar 2003.

[78] Á. BŰRMEN, D. STRLE, F. BRATKOVIČ, J. PUHAN, I. FAJFAR, T. TUMA, *Penalty Function Approach to Robust Analog IC Design*, Informacije MIDEM - Journal of Microelectronics, Electronic Componets and Materials, Vol. 32. No. 3, str. 149–156, 2002.

[79] F. BRATKOVIČ, *Računalniško načrtovanje vezij. Občutljivost in optimizacija*, Fakulteta za elektrotehniko in računalništvo, Ljubljana, 1993.

[80] Á. BŰRMEN, F. BRATKOVIČ, J. PUHAN, I. FAJFAR, T. TUMA, *Extended Global Convergence Framework for Unconstrained Optimization*, Acta Mathematica Sinica, Vol. 20, No. 3, str. 433–440, junij 2004.

[81] B. WAGNER, Á. BŰRMEN, J. PUHAN, I. FAJFAR, T. TUMA, *Uporaba ekstrapolacijskih metod pri računanju stacionarnega stanja električnih vezij*, zbornik štirinajste mednarodne Elektrotehniške in računalniške konference ERK 2005, zvezek A, (urednika: Baldomir Zajc, Andrej Trost), str. 82–85, 2005.

[82] B. WAGNER, Á. BŰRMEN, J. PUHAN, I. FAJFAR, T. TUMA, *Computing the steady-state response of nonlinear circuits by means of the ϵ-algorithm*, Elektroteh. vestn., Vol. 72, No. 5, str. 297–302, 2005.

[83] B. WAGNER, Á. BŰRMEN, J. PUHAN, S. TOMAŽIČ, T. TUMA, *Application of extrapolation algorithms in nonlinear circuit simulation and optimization with SPICE OPUS*, Informacije MIDEM, Vol. 36, No. 3, str. 140–147, 2006.

[84] B. WAGNER, Á. BŰRMEN, J. PUHAN, S. TOMAŽIČ, T. TUMA, *Optimizacija dvostopenjskega Greinacherjevega polnovalnega usmernika*, zbornik petnajste mednarodne Elektrotehniške in računalniške konference ERK 2006, zvezek A, (urednika: Baldomir Zajc, Andrej Trost), str. 96–99, 2006.

Izjava

Izjavljam, da sem doktorsko disertacijo izdelal samostojno pod mentorstvom prof. dr. Tadeja Tume. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

Borut Wagner