

České vysoké učení technické v Praze
Fakulta elektrotechnická

Disertační práce

srpen, 2018

Ing. Jiří Náhlík

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra teorie obvodů

***Optimalizace reálných periodicky
spínaných obvodů***
Disertační práce

Jiří Náhlík

Praha, srpen 2018

Doktorský studijní program: Elektrotechnika a informatika
Studijní obor: *Teoretická elektrotechnika (2602V013)*

Školitel: doc. Dr. Ing. Jiří Hospodka

Školitel specialista: Ing. Ondřej Šubrt, Ph.D.

Prohlášení

Prohlašuji, že jsem předloženou disertační práci na téma “*Optimalizace reálných periodicky spínaných obvodů*” vypracoval samostatně a v předložené práci důsledně citoval použitou literaturu.

V Praze dne 31. 8. 2018

.....
Ing. Jiří Náhlík

Poděkování

Na tomto místě bych rád poděkoval svému školiteli doc. Dr. Ing. Jiřímu Hospodkovi a svému školiteli specialistovi Ing. Ondřej Šubrtovi, Ph.D. za jejich vedení, rady a připomínky jak při výzkumné činnosti, tak i při vypracovávání této práce. Dále bych chtěl poděkovat své rodině za jejich podporu při studiu a výzkumu.

Anotace

Tato práce se zabývá optimalizačními metodami pro reálné periodicky spínané obvody. Analýza a tím i optimalizace takových obvodů počítačem je velmi časově náročná, optimalizační čas závisí na efektivitě použitého optimalizačního algoritmu. Jako jeden z prostředků, které lze na tyto úlohy použít, jsou evoluční algoritmy. Ty se již v minulosti ukázaly jako univerzálně použitelné optimalizační algoritmy, které jsou robustní, ale v některých případech méně efektivní, zejména v případech, kdy je vyžadována rychlá konvergence optimalizace. Tato práce se proto zabývá návrhem hybridních algoritmů, které by pro optimalizaci reálných periodicky spínaných obvodů byly vhodné a potlačovaly nevýhody vlastních evolučních algoritmů.

První část práce (kapitoly 1 a 2) se věnuje evolučním optimalizačním algoritmům a jejich modifikacím a také i problematice analýzy periodicky spínaných obvodů a jejich optimalizaci. Na základě analýzy současného stavu problematiky jsou pak definovány cíle práce (kapitola 3). Druhá část práce obsahuje v souladu s cíli práce výsledky výzkumu v oblasti optimalizačních algoritmů pro reálné periodicky spínané obvody. Pro testovací účely je nejprve proveden návrh banky filtrů 4. řádu se spínanými kapacitami. Dále jsou na základě analýzy dosavadního stavu problematiky navrženy hybridní optimalizační algoritmy vhodné pro optimalizaci periodicky spínaných obvodů, je navržena metodika hodnocení optimalizačních algoritmů a v závěru jsou pak jednotlivé algoritmy touto metodikou zhodnoceny na optimalizaci navrženého reálného spínaného obvodu.

Klíčová slova

Optimalizace, Diferenciální evoluce, Rojová optimalizace, Simplexový algoritmus, Hybridní optimalizační algoritmy, Analogové diskrétně pracující obvody, Spínané kapacitami, Filtry, Banka filtrů

Abstract

This thesis deals with optimization methods for real periodically switched circuits. The analysis and thus the optimization of such circuits by the computer is very time-consuming and the optimization time depends on the efficiency of a used optimization algorithm. One of the means, which can be used for these tasks, is evolutionary algorithm. Evolutionary algorithms show to be universally applicable and robust but in some cases less efficient, especially when rapid optimization convergence is required. This thesis deals with the design of hybrid algorithms that would be suitable for the optimization of real periodically switched circuits and suppress the disadvantages of evolutionary algorithms.

The first part of the thesis (chapters 1 and 2) deals with evolution optimization algorithms and their modifications as well as with the problems of analysis of periodically switched circuits and their optimization. Based on an analysis of the current state of the art, the objectives of the work are defined (chapter 3). The second part of the thesis contains results of research in the field of optimization algorithms for real periodically switched circuits. For testing purposes, a 4th order filter bank with switched capacitors is designed. Then the hybrid optimization algorithms for the optimization of periodically switched circuits are proposed as well as the methodology of evaluation of optimization algorithms. Based on this methodology the proposed algorithms are evaluated on optimization of the proposed real switching circuit.

Key words

Optimization, Differential Evolution, Particle swarm optimization, Simplex algorithms, Hybrid optimization algorithms, Analogue discrete time circuits, Switched Capacitor, Filter, Filter bank

Obsah

| | | |
|-------|--|----|
| 1 | Úvod..... | 1 |
| 2 | Současný stav problematiky | 3 |
| 2.1 | Možnosti AC analýzy periodicky spínaných obvodů | 3 |
| 2.2 | Optimalizační algoritmy..... | 4 |
| 2.2.1 | Evoluční algoritmy..... | 6 |
| 2.2.2 | Obecné principy evolučních algoritmů..... | 7 |
| 2.2.3 | Účelová funkce | 8 |
| 2.2.4 | Omezení rozsahu proměnných..... | 9 |
| 2.2.5 | Genetické algoritmy - GA..... | 11 |
| 2.2.6 | Diferenciální evoluce - DE | 12 |
| 2.2.7 | Rojení částic - PSO | 17 |
| 2.2.8 | Nelder – Mead simplexový algoritmus..... | 21 |
| 2.2.9 | Hybridní algoritmy DE, PSO, NM simplexu..... | 24 |
| 2.3 | Zhodnocení současného stavu problematiky..... | 25 |
| 3 | Cíle disertační práce..... | 27 |
| 4 | Výsledky disertační práce | 29 |
| 4.1 | Implementace banky filtrů třetího řádu | 30 |
| 4.2 | Praktické aspekty optimalizací reálných struktur..... | 33 |
| 4.2.1 | Vliv kvality generování náhodných čísel..... | 33 |
| 4.2.2 | Operační zesilovače využívané v SC obvodech | 34 |
| 4.3 | Návrh SC banky filtrů čtvrtého řádu | 35 |
| 4.3.1 | Implementace bikvadratických sekcí | 37 |
| 4.3.2 | Návrh operačního zesilovače pro banku filtrů | 39 |
| 4.3.3 | Optimalizace navržené banky filtrů | 41 |
| 4.4 | Navržené hybridní algoritmy pro periodicky spínané obvody..... | 42 |

| | | |
|-------|---|----|
| 4.4.1 | Algoritmus DE-PSO1 | 42 |
| 4.4.2 | Algoritmus DE-PSO2 | 43 |
| 4.4.3 | Algoritmus PSO-DEm | 45 |
| 4.4.4 | Algoritmus DESX..... | 46 |
| 4.5 | Hodnocení navržených hybridních algoritmů | 49 |
| 4.5.1 | Použitá účelová funkce | 49 |
| 4.5.2 | Metodika hodnocení optimalizačních algoritmů | 50 |
| 4.5.3 | Nastavení algoritmů a testu..... | 51 |
| 4.5.4 | Výsledky porovnání algoritmů | 52 |
| 5 | Závěr | 59 |
| 5.1 | Přínosy práce | 59 |
| 5.2 | Další možný směr výzkumu..... | 61 |
| 6 | Literatura..... | 63 |
| 7 | Seznam autorových publikací | 73 |
| 7.1 | Publikace vztahující se k tématu práce | 73 |
| 7.2 | Ostatní publikace a výsledky nevztahující se k tématu práce | 75 |

1 Úvod

Periodicky spínané elektronické obvody jsou pro své vlastnosti součástí integrovaných řešení na čipu již několik desetiletí a můžeme se s nimi setkat v různých oblastech zpracování analogového signálu od elektrických filtrů až po Analogově-Digitální a Digitálně-Analogové převodníky. Mezi periodicky spínané analogové obvody můžeme zařadit dva principy:

1. techniku spínaných kapacitorů a
2. techniku spínaných proudů.

První ze jmenovaných principů se začal rozvíjet v sedmdesátých letech, druhý pak v devadesátých letech dvacátého století.

Technika spínaných kapacitorů je v oblasti zpracování signálu dnes asi nejvíce užívanou technikou pro implementaci elektrických filtrů na čipu, a to především pro její výhodné vlastnosti, které nabízí oproti implementaci pomocí ARC filtrů:

- nepoužívá rezistorů, jejichž implementace je na čipu problematická,
- přenos není závislý na absolutních hodnotách hodnot kapacitorů, ale pouze na jejich vzájemném poměru,
- doladění či přeladění filtru lze realizovat změnou spínacího kmitočtu.

Těchto vlastností se s výhodou využívá právě především při návrhu filtrů na čipu. Některé návrhové postupy a metody analýzy jak idealizovaných, tak i obvodů s částečně reálnými vlastnostmi, byly v minulosti již publikovány [1–4]. Stále však nepostihují všechny reálné vlastnosti a výsledný navržený obvod musí být simulován v časové oblasti. Pro odstranění vlivu reálných vlastností je pak třeba obvod optimalizovat.

Optimalizací obvodů se spínanými kapacitami se již zabývala práce [5], kde však autor pro optimalizovaný návrh uvažuje pouze lineární modely s následujícími neidealitami: odpor spínačů v sepnutém stavu a konečný zisk a šířku pásma operačního zesilovače. Pro zrychlení výpočtů také autor využíval semi-symbolického vyjádření optimalizované funkce lineárního SC obvodu. Optimalizace tak postihovala pouze elementární, základní vlastnosti použitých součástí. Jeho práce ovšem ukazuje algoritmus Diferenciální evoluce jako vhodnou základní optimalizační metodu pro tento typ úlohy optimalizace.

Při optimalizaci spínaného obvodu na transistorové úrovni je třeba provádět analýzu obvodu v časové oblasti, což je výpočetně a časově mnohem náročnější. Je tedy

třeba volit optimalizační algoritmus a jeho nastavení, které bude dosahovat spolehlivých výsledků i při velmi malém počtu simulací obvodu.

Optimalizací periodicky spínaných obvodů se zabývá i tato práce, která navazuje na výsledky [5]. Zabývá vlastnostmi obvodů s reálnými (nelineárními) modely aktivních prvků a navrhuje hybridní optimalizační algoritmy založené především na spojení metody diferenciální evoluce s dalšími algoritmy tak, aby při použití malého počtu jedinců generace byla spolehlivost optimalizace vyšší než u jiných evolučních algoritmů.

Pro porovnání optimalizačních algoritmů pak autor využívá konkrétní návrh dvoukanálové banky filtrů čtvrtého řádu, kterou implementuje pomocí obvodů se spínanými kapacitami v technologii AMS 0.35 μm , kde vycházel i ze zkušeností s implementací SC banky filtrů třetího řádu, kterou implementoval na plošném spoji.

2 Současný stav problematiky

Vzhledem k zaměření práce na využití optimalizačních algoritmů při návrhu reálných periodicky spínaných obvodů na čipu bude tato kapitola především věnována optimalizačním algoritmům, jejich vlastnostem a jejich modifikacemi.

S rozvojem výpočetní techniky se postupně vyvíjela i kategorie tzv. evolučních optimalizačních algoritmů, které se ukázaly jako užitečný nástroj pro optimalizaci návrhu elektronických obvodů [6, 7]. Optimalizačními algoritmy aplikovanými při návrhu elektronických obvodů se již zabývalo mnoho prací, a to jak optimalizací analogových obvodů [8–10], tak i digitálních [11, 12]. Optimalizací periodicky spínaných analogových obvodů se však mnoho prací nezabývalo [4, 13]

Pro optimalizaci elektrických obvodů jsou vhodné evoluční algoritmy a jejich modifikace [3, 10, 14, 15]. Z toho důvodu budou tyto algoritmy rozebrány podrobněji. Protože jsou optimalizační metody porovnávány na návrhu SC banky filtrů na čipu, jsou také uvedeny i možnosti AC analýzy periodicky spínaných obvodů a některé další aspekty jejich implementace a optimalizace.

2.1 Možnosti AC analýzy periodicky spínaných obvodů

AC analýza periodicky spínaných obvodů je oproti analýze spojitě pracujících obvodů složitější, avšak je jí třeba zejména při analýze kmitočtových charakteristik navržených filtrů.

Pro AC analýzu periodicky spínaných obvodů lze použít následující postupy [16]:

1. analýzu pomocí nábojových rovnic [17, 18], implementovanou pomocí modifikované metody uzlových napětí i pro techniku spínaných proudů [16, 18, 19],
2. řešení obecných lineárních diferenciálních rovnic pomocí metod založených na Fourierově nebo Laplaceově transformaci [20, 21],
3. obecné řešení nelineárních rovnic v časové oblasti (obvykle num. metodami) s použitím diskrétní Fourierovy transformace [22] pro analýzu kmitočtových vlastností.

Výsledkem analýzy první ze jmenovaných metod je přenosová funkce idealizovaného lineárního obvodu v Z-rovině. V případě druhé metody je to opět přenosová funkce lineárního, resp. linearizovaného obvodu, která však může zahrnovat i vliv neideálních parametrů obvodu (odpor spínačů, kmitočtovou závislost zesílení operačních

zesilovačů,...). Poslední jmenovanou metodou lze postihnout všechny vlastnosti obvodu (např. nelineární chování OZ, apod.). Tuto metodu lze využít při analýze obvodů na tranzistorové úrovni (např. při návrhu čipu), je však časově i výpočetně nejnáročnější, ale je jí třeba využívat především v případech, kdy je třeba do analýzy zahrnout vliv všech parazitních vlivů, tj. např. v případech, kdy analyzovaný obvod zpracovává signál s kmitočtovým rozsahem srovnatelným se vzorkovacím kmitočtem a vliv parazitních vlastností obvodu je značný nebo v případech, kdy je vyžadována velká přesnost obvodu (např. pro měřicí aplikace apod.).

2.2 Optimalizační algoritmy

Optimalizací obvykle rozumíme hledání řešení daného problému tak, aby toto řešení splňovalo definované požadavky. Optimalizačním algoritmem bychom tedy mohli nazvat postup, jehož cílem je nalezení nejvýhodnějšího řešení daného problému buď zcela, nebo jen z určitého hlediska.

Příkladem optimalizace může být řešení problému obchodního cestujícího, spotřeby paliva, optimalizace nákladů, či například i rozmístění nákladu na přívěsu... V elektrotechnické praxi se nejčastěji setkáme s použitím optimalizace při návrhu číslicových či analogových filtrů [5, 14, 15, 23], či parametrů operačních zesilovačů [10, 24, 25].

Vzhledem k různorodosti optimalizovaných problémů je třeba ohodnocení výhodnosti jednotlivých řešení vhodně převést na tzv. *Účelovou funkci* (v angl. literatuře často označované jako *fitness function*, či *objective function* – pro některé algoritmy je vhodné tyto pojmy rozlišovat, protože představují rozdílný princip ohodnocení). Optimalizační úlohou je najít takovou kombinaci proměnných, aby hodnota účelové funkce byla minimální či maximální (v závislosti na definici).

Hledání extrému účelové funkce můžeme provádět buď analyticky, či numericky. Analytické řešení extrémů účelové funkce však není v mnoha případech možné (např. kvůli velké složitosti problému). S rozvojem počítačové techniky se tak v poslední době optimalizační úlohy řeší pomocí numerických metod.

Optimalizační algoritmy můžeme dle principu jejich činnosti v základu dělit na [15, 26]:

- Enumerativní – výpočet pomocí procházení všech možných kombinací proměnných. Z principu jejich činnosti je lze použít jen pro malý počet proměnných omezených na určitém intervalu.
- Deterministické – založené na klasických matematických metodách, kdy prohledávaný prostor obvykle musí splňovat určité podmínky (např. spojitost, diferencovatelnost...).
- Stochastické – využívají ke své práci náhodné hledání.
- Smíšené – metody kombinující deterministické a stochastické algoritmy.

Mezi deterministické algoritmy můžeme například zařadit různé gradientní metody, horolezecký algoritmus, Newtonovu metodu a další. Příkladem stochastických metod je metoda simulovaného žíhání, evoluční či genetické algoritmy nebo stochastický horolezecký algoritmus.

Smíšené algoritmy (nebo také hybridní algoritmy) obvykle přejímají od jednotlivých algoritmů jejich výhodné vlastnosti. Např. od deterministických algoritmů typu gradientních schopnost efektivního nalezení lokálního minima, pokud se pohybujeme v jeho okolí a od stochastických metod právě výhodu náhodného hledání, která obvykle zmenšuje pravděpodobnost "uvíznutí" algoritmu v lokálním extrému.

Mezi smíšené a stochastické algoritmy také můžeme zařadit kategorii/podkategorii heuristických algoritmů, kde rozhodnutí o další iteraci optimalizačního algoritmu závisí na předchozích iteracích algoritmu [27]. Rychlost konvergence k extrému účelové funkce heuristických metod bývá obvykle dobrá, a proto se velmi často používají k řešení problémů, kde použití deterministických metod není z hlediska časové či výpočetní náročnosti účelné. Heuristické algoritmy oproti deterministickým také zpravidla nekladou vysoké nároky na účelovou funkci (spojitost, diferencovatelnost...). Pro tyto jejich vlastnosti se heuristické/stochastické metody využívají i přes to, že z principu jejich činnosti nemusí nalézt globální extrém, a pro každý průběh algoritmu mohou nalézt jinou kombinaci proměnných účelové funkce (mohou konvergovat k různým lokálním extrémům účelové funkce). Nejznámějšími zástupci heuristických algoritmů je skupina evolučních algoritmů – např. genetický algoritmus [28, 29] ,dále algoritmus mravenčí kolonie [30, 31] či rojení částic [32].

Vzhledem k výhodným vlastnostem evolučních algoritmů a jejich možnosti využití pro optimalizaci elektronických obvodů jsou základní principy těchto algoritmů uvedeny v následující části práce.

2.2.1 Evoluční algoritmy

Evoluční algoritmy lze zařadit mezi optimalizační algoritmy smíšené, které v sobě vhodně kombinují přístupy jak deterministických, tak i stochastických metod [5, 26, 29, 33, 34] Pro svou funkci potřebují účelovou funkci, která svou hodnotou vyjadřuje výhodnost daného řešení. Účelem optimalizačního algoritmu je pak najít globální minimum či globální maximum této funkce. Proměnné této funkce mohou pocházet z různých číselných oborů a obvykle jsou tyto proměnné pro konkrétní optimalizaci omezeny na nějakém intervalu. Oproti jiným algoritmům však obvykle nepožadují, aby účelová funkce byla spojitá, diferencovatelná a pod... Obvykle k jejich funkci potřebují jen její hodnotu.

Z principu jejich funkce lze tyto algoritmy s výhodou použít pro optimalizaci, kde účelová funkce obsahuje na prohledávaném prostoru více extrémů (jak lokálních, tak i globálních), kde tyto metody na rozdíl např. od klasických gradientních metod jsou schopny se z lokálních extrémů vymanit a nalézt v prohledávaném prostoru globální extrém.

Další výhodou evolučních algoritmů je, že obvykle nepotřebují přesné nastavení řídicích konstant a počáteční nastavení proměnných. I přes to jsou pak schopny konvergovat ke globálnímu extrému účelové funkce. Vhodné počáteční nastavení ale může zrychlit proces optimalizace.

Daní za robustnost je však pomalejší konvergence ve srovnání s gradientními metodami (a tedy delší optimalizační čas) a také to, že nemusí vždy konvergovat do stejného bodu optimalizačního prostoru a doba optimalizace se může pro každý průběh optimalizačního algoritmu lišit (což je důsledek používání náhodných čísel).

Mezi evoluční algoritmy můžeme zařadit např. tyto algoritmy [5, 26]:

- evoluční strategie,
- evoluční programování,
- diferenciální evoluce,
- genetické algoritmy,
- genetické programování,
- simulované žíhání,
- zakázané prohledávání,
- optimalizace mravenčí kolonií,
- rojení částic (popř. rojová optimalizace),

Principy některých těchto algoritmů budou v následujícím textu rozebrány (neboť jejich principů je v práci využito). Nejprve však budou rozebrány principy, které mají některé algoritmy společné.

2.2.2 Obecné principy evolučních algoritmů

Jak je z názvu těchto algoritmů patrné, je princip evolučních algoritmů inspirován evolučními jevy v přírodě [26, 29]. Evoluční algoritmy pracují v cyklech, kde soubor jednotlivých testovaných řešení optimalizační úlohy obvykle označujeme *populací* nebo *generací*. Jednotliví jedinci populace jsou pak složeni z jednotlivých chromozomů, kde *chromozomy* nazýváme jednotlivé proměnné v optimalizačním prostoru. Populaci si lze tedy představit jako soubor *n-rozměrných* vektorů (jedinců) jimž je vždy přiřazena odpovídající hodnota účelové funkce. Počet jedinců v populaci označme *NP*.

Náhodně vygenerovaná počáteční populace *NP* jedinců je pak nahrazena částečně nebo zcela následující generací potomků na principech podobných přírodě (v každé další iteraci se také vytváří potomci ze svých předchůdců - rodičů). Celý proces se opakuje v cyklech, dokud není splněna ukončovací podmínka. Ukončovací podmínka je obvykle dána předem definovanou hodnotou účelové funkce, která uživateli postačí, a maximálním počtem iterací algoritmu pro případ, kdy nebude možné splnit podmínku na hodnotu účelové funkce.

Způsoby, jakými se vytváří jednotlivé generace případně mezigenerace, ze které se vybírají jedinci do nové generace, se pro jednotlivé evoluční algoritmy liší. Lze však jmenovat tři hlavní principy vytváření nových jedinců následující generace:

- křížením,
- mutací,
- selekcí.

Křížení

Tento princip vytváří nové jedince tak, že z několika jedinců předchozí generace vytváří jednoho či více jedinců nových tak, že jednotlivé chromozomy či skupiny chromozomů (proměnných) převezme od jednoho jedince, další chromozom či skupinu pak od dalšího jedince atd...

Podle toho, jak algoritmus definuje jednotlivé chromozomy, lze pak dělit křížení na jednobodové, dvoubodové až vícebodové (především pro genetické algoritmy, kde chromozom obvykle představuje hodnotu jednoho bitu v binárním zápisu) [26]. Dalším

možným příkladem křížení je například použití průměru jednotlivých chromozomů, kde chromozom však již bývá reprezentován reálným číslem.

Mutace

Mutace je operace, kdy se nový jedinec vytváří pomocí malé změny náhodně vybraných proměnných/chromozomů aktuálního jedince. Tyto změny lze aplikovat přičtením malého náhodného čísla k jednotlivým proměnným, či v případě binární reprezentace u genetických algoritmů změnou logické hodnoty (0->1 nebo 1->0) některého chromozomu.

Selekce

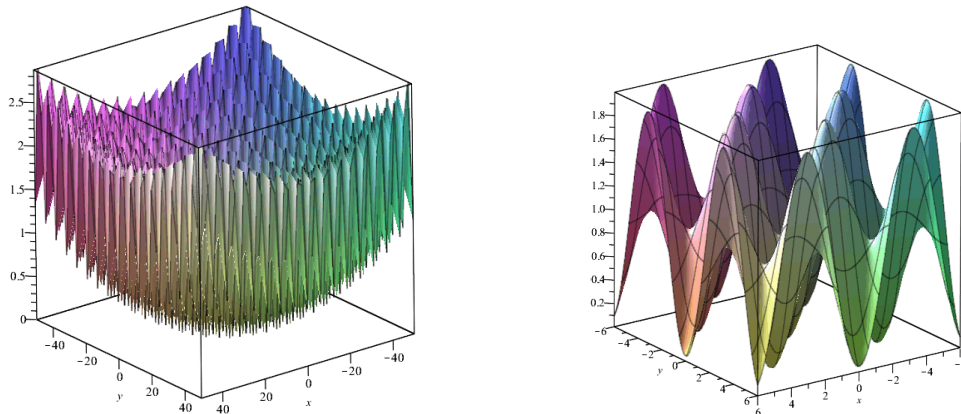
Selekcí je v případě evolučních algoritmů aplikace principu, kterým se vybírají jedinci z mezigenerace (ta vzniká na principech křížení a mutace) do následující generace. O selekci můžeme také mluvit, pokud vybíráme rodičovské vektory pro jejich kombinaci do nového potomka v mezigeneraci. Ze selekčních mechanismů jmenujme ruletový mechanismus, pořadovou selekci, elitismus, či turnajovou selekci [5, 15, 26].

2.2.3 Účelová funkce

Účelová funkce je matematická funkce $f(x_1, \dots, x_i, \dots, x_D) \rightarrow Y$, kde $Y \subseteq \mathbb{R}$, která popisuje předmět optimalizace [27]. Její funkční hodnota pro určitou kombinaci proměnných určuje výhodnost daného řešení [26]. V častém případě (například právě evolučních algoritmů) mohou být jednotlivé proměnné účelové funkce dokonce z různých množin (množiny reálných či celých čísel a pod...). Účelovou funkcí je obvykle funkce více proměnných s relativně jednoduchým zápisem, která může být definována i součtem jednotlivých účelových funkcí pro různá kritéria. Jako příklad účelové funkce uveďme Griewangkovu funkci [26, 35]:

$$f(x_1, \dots, x_i, \dots, x_D) = 1 + \sum_{i=1}^D \frac{x_i}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right), \quad (1)$$

kde D je dimenze (zvolený počet proměnných), x_i je i -tá proměnná prohledávaného prostoru. Tato funkce má globální minimum rovné 0 v bodě $(x_1, x_2, \dots, x_D) = (0, 0, \dots, 0)$. Grafické vyjádření této funkce pro 2 proměnné je znázorněno na obr. 1.



Obr. 1 Griewankova funkce pro 2 proměnné (vlevo) a v detailu (vpravo)

Optimalizační algoritmy se obvykle testují na sadě matematických funkcí více proměnných (obdobných Griewankově funkci), které obsahují velký počet lokálních extrémů a obvykle jeden globální extrém [5, 26, 36–38]. Účelové funkce a dimenze problémů jsou obvykle voleny tak, aby algoritmus prokázal svou rychlost i robustnost pro méně i vícedimenzionální problémy.

V elektrotechnické praxi jsou obvykle účelové funkce voleny ve tvaru odchylek od požadovaných parametrů nebo definovaných mezí [5, 14]. Například pro návrh hodnot součástek elektrického filtru by účelová funkce mohla být zapsána ve tvaru:

$$f(x_1, \dots, x_D) = \sum_{i=1}^N |A_i(i \cdot f_0) - A_{i,def}(i \cdot f_0)| + \sum_{m=1}^M P_m, \quad (2)$$

kde $A_i(i \cdot f_0)$ je hodnota modulu přenosu filtru pro určitý násobek základního kmitočtu f_0 , $A_{i,def}(i \cdot f_0)$ jsou požadované hodnoty modulu přenosu filtru a P_m jsou penalizační funkce, které zohledňují další parametry, jako jsou realizovatelnost řešení, velikost na čipu a pod...

V literatuře [5, 10] lze nalézt i doporučení pro tvar účelové funkce vhodné pro optimalizaci elektronických obvodů, kdy dílčí účelové funkce vyjadřují relativní odchylku charakteristik obvodu od požadovaných parametrů.

2.2.4 Omezení rozsahu proměnných

Optimalizační algoritmy během svého pracovního cyklu vytvářejí různými způsoby nové vektory potenciálních řešení daného optimalizačního problému. Hodnoty proměnných těchto vektorů ovšem nemusí nutně spadat do intervalu přípustných hodnot.

Algoritmus tuto situaci může řešit buď penalizační funkcí při ohodnocení takového řešení pomocí účelové funkce, či generováním hodnoty proměnné v daném intervalu.

Penalizační funkce

Úkolem penalizační funkce obvykle bývá zamezit optimalizačnímu algoritmu konvergovat mimo optimalizační prostor. Nabývá tedy takových hodnot, aby změnila hodnotu účelové funkce dále od předpokládané hodnoty optima. Pokud tedy algoritmus hledá globální minimum účelové funkce, pak pro nepřipustné hodnoty bude penalizační funkce nabývat vysokých kladných hodnot. Příkladem penalizační funkce pak může být funkce

$$P_1 = \begin{cases} 100 + 1000 \cdot |x_1 - x_{1max}|, & x_1 > x_{1max} \\ 0, & x_1 \leq x_{1max} \end{cases}, \quad (3)$$

kteřá nabývá nenulové kladné hodnoty v případě, že proměnná x_1 testovaného vektoru je větší než její povolená maximální hodnota x_{1max} . Výhodou takto konstruované penalizační funkce je navíc to, že zohledňuje i míru přesahu povoleného intervalu proměnné a nabývá tím větší hodnoty, čím více bylo dané hranice přesaženo. Obdobnou penalizační funkci použijeme především v případech, kdy lze očekávat, že účelová funkce bude mít mimo vyhrazený optimalizační prostor i jiné lokální extrémy srovnatelné (ne-li lepší) s extrémy uvnitř optimalizačního algoritmu.

Generace proměnné v intervalu přípustných hodnot

Dalším možným mechanismem, jak lze omezit rozsah proměnné během optimalizačního cyklu, je v případě přesahu intervalu přípustných hodnot její opětovné vygenerování v intervalu dle předpisu:

$$x_i = r \cdot (x_{i,max} - x_{i,min}) + x_{i,min}, \quad (4)$$

kde r je náhodné číslo z intervalu $\langle 0, 1 \rangle$ a $x_{i,min}$ a $x_{i,max}$ jsou hraniční hodnoty pro proměnnou x_i . Tento způsob navrácení proměnné do určeného rozsahu ale nezohledňuje, kterou mez proměnná překročila. Proto je výhodnější použít generaci nové proměnné buď v dolní nebo horní polovině daného intervalu dle předpisu

$$x_{i,nové} = \begin{cases} r \cdot \left(\frac{x_{i,max} + x_{i,min}}{2} - x_{i,min} \right) + x_{i,min}, & x_{i,původní} < x_{i,min} \\ r \cdot \left(x_{i,max} - \frac{x_{i,max} + x_{i,min}}{2} \right) + \frac{x_{i,max} + x_{i,min}}{2}, & x_{i,původní} > x_{i,max} \end{cases}, \quad (5)$$

kde význam jednotlivých symbolů je obdobný předchozímu předpisu (4).

2.2.5 Genetické algoritmy - GA

Genetické algoritmy jsou nejznámějším zástupcem evolučních algoritmů, kdy princip jejich činnosti poprvé představil J. Holland [5, 26, 28]. Označení genetický algoritmus lze přisoudit velké skupině algoritmů odvozených od klasického genetického algoritmu. Jejich praktické použití v technice pak bylo použito v [39].

Nejstarším a také nejznámějším typem GA je standardní genetický algoritmus (SGA – Simple Genetic Algorithm, nebo také Standard Genetic Algorithm), kde proměnné (chromozomy) jsou tvořeny dvouhodnotovými symboly, kterým se říká geny (nejčastěji 0 a 1) [29].

Výhodnost chromozomů je ohodnocena fitness funkcí (resp. přetransformovanou účelovou funkcí). Nejvýhodnější chromozomy jsou pak ohodnoceny nejvyšší hodnotou fitness funkce (např. číslem 1). Ostatní chromozomy jsou hodnoceny ve vztahu k nejlepšímu dle své účelové funkce.

Z takto „seřazených“ chromozomů jsou pak selekčním mechanismem (ruletová či pořadové selekce, elitismus...) vybírány chromozomy, které jsou mezi sebou kříženy. Výsledkem jsou pak nové chromozomy, které obsahují část genů od každého jedince. Dalším mechanismem vytváření nového chromozomu je také princip mutace.

V průběhu GA se může stát, že nejlepší jedinec je transformacemi a selekcí odstraněn z populace, proto se obvykle nejlepší dosažený výsledek navíc uchovává ve zvláštní proměnné a noví jedinci populace jsou vždy s tímto výsledkem porovnáváni.

Modifikací genetických algoritmů, které odstraňují některé nevýhody původního principu, byla navržena řada a jejich stručný popis lze nalézt např. v [40].

2.2.6 Diferenciální evoluce - DE

Optimalizační metoda diferenciální evoluce byla poprvé publikována v polovině devadesátých let R. Stornem a K.Pricem [36]. Vznikla postupnou modifikací genetického žíhání [26, 41], které bylo později z metody pro svou nadbytečnost vypuštěno. Vzhledem k úspěchu metody v porovnání s ostatními algoritmy [37, 42], možnosti aplikace na velkou škálu různorodých problémů a jejímu jednoduchému programovému zápisu se tato metoda stala centrem pozornosti velkého množství odborníků z různých oborů [14, 43, 44]. Metoda byla popsána samotnými autory i v dalších publikacích [37, 42, 45] a její využití lze nalézt v řadě prací [14, 43, 44, 46].

Základním principem této evoluční metody je generace nových jedinců pomocí kombinace tří jedinců a to tak, že rozdílový vektor dvou náhodně vybraných jedinců se přičte k jedinci třetímu. Pro svou činnost tato metoda potřebuje nastavení pouze několika parametrů:

- práh křížení CR ,
- mutační konstanta F ,
- počet jedinců generace NP ,
- parametr λ .

Práh křížení CR

Tento parametr může nabývat hodnot v intervalu $CR \in \langle 0,1 \rangle$ [26] a určuje pravděpodobnost, s jakou bude nový jedinec vytvořen pomocí kombinace s ostatními. Hodnota tohoto parametru se obvykle nastavuje v rozmezí $CR \in \langle 0.8, 0.9 \rangle$ [26, 47]. Čím větší je hodnota CR , tím více vzniká nových jedinců pomocí křížení. Hodnota prahu křížení ovlivňuje rychlost konvergence, kdy pro menší hodnoty se jedinci mají tendenci generovat v ortogonálních konfiguracích [48].

Mutační konstanta F

Tato konstanta určuje míru, s jakou se do potenciálního nového promítne rozdílový vektor dvou náhodně vybraných jedinců. Její hodnota může ležet v intervalu $F \in \langle 0, 2 \rangle$, obvykle se ale nastavuje v rozmezí $F \in \langle 0.3, 0.9 \rangle$ [26, 33]. V literatuře lze nalézt i vztah určující spodní hranici intervalu mutační konstanty ve vztahu k ostatním parametrům optimalizace [48–50]:

$$F_{crit} = \sqrt{\frac{1 - \frac{CR}{2}}{NP}}. \quad (6)$$

Pokud by byla hodnota mutační konstanty nastavena menší, pak rozptyl generací postupně klesá [48, 49]. Dle [50] resp. [47] je spodní hranice tohoto parametru 0.4.

Počet jedinců generace NP

Počet jedinců v generaci se odvozuje od počtu proměnných účelové funkce n , kdy NP se obvykle volí jako celočíselný násobek počtu proměnných n v rozsahu $NP \in \langle 2n, 100n \rangle$.

Na volbě počtu jedinců závisí výpočetní čas jedné generace resp. iterace algoritmu. Z tohoto hlediska se zdá výhodnější použít malý počet jedinců. S vyšším počtem jedinců v generaci ovšem roste pravděpodobnost přiblížení se optimu během jedné iterace algoritmu.

Dle [48] se používaný počet jedinců ustálil na $NP = 10n$, kde [47] však uvádí jako přijatelnou volbu také rozsah $NP \in \langle 3n, 8n \rangle$. Počet potřebných jedinců však ovlivňuje i použitá varianta algoritmu.

Parametr λ

Parametr λ pro některé verze algoritmu (viz dále) ovlivňuje míru, s jakou je do mutace vektorů zohledňován i nejlepší dosud nalezený jedinec generace. Jeho hodnota se obvykle pohybuje v intervalu $\lambda \in \langle 0, 1 \rangle$ [36].

2.2.6.1 Principy DE

Základní princip DE algoritmu je obdobný ostatním evolučním algoritmům (např. GA). Pro generaci nových jedinců se používá jak mutace, tak křížení. Oproti GA ale nepoužívá princip selekce ve smyslu GA a všichni jedinci generace se podílejí na vytváření nové generace [5, 36, 37]. Základní podobu algoritmu lze rozdělit na počáteční inicializaci algoritmu a proces optimalizace. Jednotlivé části pak lze popsat následovně.

Inicializace algoritmu:

- nastavení řídicích konstant algoritmu (F, CR, NP, λ),
- vytvoření počáteční generace jedinců (pomocí generátoru náhodných čísel),
- definování účelové funkce,
- definování ukončovací podmínky algoritmu.

Optimalizační proces:

Pro všechny jedince populace se provede následující postup tvorby nového jedince (označme aktuálního jedince indexem j):

1. Generace dvou až pěti náhodných přirozených čísel r_1, r_2, r_3, r_4, r_5 , z rozsahu $\langle 1, NP \rangle$. Tato čísla musí být od sebe navzájem odlišná (neboť představují čísla jedinců, kteří se účastní křížení/mutace).
2. Generace náhodného přirozeného čísla L z intervalu $L \in \langle 1, n \rangle$, kde n je počet proměnných jedince.
3. Výpočet tzv. šumového vektoru \vec{v} ze tří až pěti rodičů dle předpisů uvedených níže.
4. Pro všechna čísla indexů i proměnných, ze kterých je složen jedinec $\{1, 2, \dots, n\}$ je vygenerováno náhodné číslo q_i z intervalu $\langle 0, 1 \rangle$ a s jejich pomocí je vytvořen zkušební vektor \vec{z} (zkušební jedinec), kde jednotlivé parametry jsou dány vztahem

$$z_i = \begin{cases} v_i & i = L \text{ nebo } q_i < CR \\ x_{i,j,G} & \text{jinak} \end{cases}, \quad (7)$$

kde $x_{i,j,G}$ představuje i -tý prvek jedince j , z aktuální generace G

5. Pro vygenerovaný vektor \vec{z} vypočteme hodnotu účelové funkce ÚF.
6. Do následující generace $G + 1$ pak postupuje vektor $\overrightarrow{x_{j,G+1}}$ dle následujícího předpisu:

$$\overrightarrow{x_{j,G+1}} = \begin{cases} \vec{z} & \text{ÚF}(\vec{z}) \leq \text{ÚF}(\overrightarrow{x_{j,G}}) \\ \overrightarrow{x_{j,G}} & \text{jinak} \end{cases}. \quad (8)$$

Tento postup vytváření nové generace jedinců se opakuje až do splnění ukončovací podmínky.

Výpočet šumového vektoru \vec{v} - varianty DE

Během vývoje algoritmu bylo definováno několik způsobů, jakými vytvářet šumový vektor vektoru \vec{v} z jedinců aktuální generace $\overrightarrow{x_{y,G}}$, kde index y představuje číslo jedince v generaci. Nejběžnější varianty výpočtu šumového vektoru uveďme v přehledové tab. 1, kde $\overrightarrow{x_{best,G}}$ je vektor s nejlepší dosaženou hodnotou účelové funkce [5, 15, 26, 42].

Tab. 1 Nejběžnější varianty výpočtu šumového vektoru

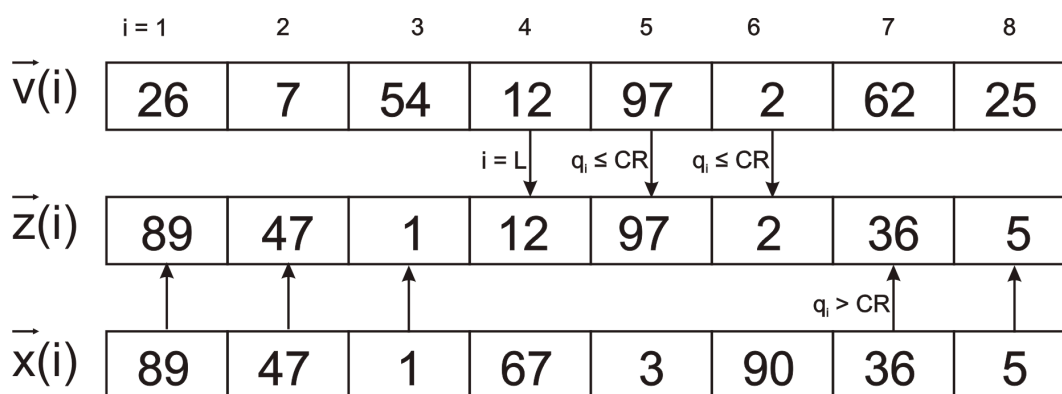
| Označení varianty | Předpis vytvoření šumového vektoru |
|-----------------------|---|
| DE/rand/1/bin | $\vec{v} = \overline{x_{r_1,G}} + F(\overline{x_{r_2,G}} - \overline{x_{r_3,G}})$ |
| DE/best/1/bin | $\vec{v} = \overline{x_{best,G}} + F(\overline{x_{r_2,G}} - \overline{x_{r_3,G}})$ |
| DE/rand-to-best/1/bin | $\vec{v} = \overline{x_{r_1,G}} + \lambda(\overline{x_{best,G}} - \overline{x_{r_1,G}}) + F(\overline{x_{r_2,G}} - \overline{x_{r_3,G}})$ |
| DE/best/2/bin | $\vec{v} = \overline{x_{best,G}} + F(\overline{x_{r_1,G}} + \overline{x_{r_2,G}} - \overline{x_{r_3,G}} - \overline{x_{r_4,G}})$ |
| DE/rand/2/bin | $\vec{v} = \overline{x_{r_1,G}} + F(\overline{x_{r_2,G}} + \overline{x_{r_3,G}} - \overline{x_{r_4,G}} - \overline{x_{r_5,G}})$ |
| DE/rand/1/exp | $\vec{v} = \overline{x_{r_1,G}} + F(\overline{x_{r_2,G}} - \overline{x_{r_3,G}})$ |
| DE/best/1/exp | $\vec{v} = \overline{x_{best,G}} + F(\overline{x_{r_2,G}} - \overline{x_{r_3,G}})$ |
| DE/rand-to-best/1/exp | $\vec{v} = \overline{x_{r_1,G}} + \lambda(\overline{x_{best,G}} - \overline{x_{r_1,G}}) + F(\overline{x_{r_2,G}} - \overline{x_{r_3,G}})$ |
| DE/best/2/exp | $\vec{v} = \overline{x_{best,G}} + F(\overline{x_{r_1,G}} + \overline{x_{r_2,G}} - \overline{x_{r_3,G}} - \overline{x_{r_4,G}})$ |
| DE/rand/2/exp | $\vec{v} = \overline{x_{r_1,G}} + F(\overline{x_{r_2,G}} + \overline{x_{r_3,G}} - \overline{x_{r_4,G}} - \overline{x_{r_5,G}})$ |

Označení algoritmů odpovídá již zavedené praxi, kdy za prvním lomítkem uvádíme, z jakého jedince (vektoru) má být šumový vektor vytvářen, za druhým lomítkem uvádíme počet rozdílových vektorů jedinců a za třetím lomítkem uvádíme způsob křížení šumového vektoru s aktuálním jedincem (nejčastěji *bin* nebo *exp*). Binomickému křížení (*bin*) odpovídá předpis uvedený v bodu 4 optimalizačního procesu. Exponenciální křížení (*exp*) [50] probíhá následujícím způsobem:

- do vektoru \vec{z} je nakopírován aktuální jedinec $x_{i,j,G}$,
- je vygenerováno náhodné číslo L z přirozených čísel v rozsahu $L \in \langle 1, n \rangle$, obdobně jako v bodu 2 optimalizačního procesu,
- od tohoto parametru (proměnné) jsou další proměnné kopírovány do zkušebního vektoru \vec{z} patřící šumovému vektoru \vec{v} až do chvíle, kdy pro proměnnou neplatí podmínka $q_i \leq CR$. Zápis takové funkce algoritmu může být následující:

$$z_i = \begin{cases} x_{i,j,G} & i < L \\ v_i & i = L \text{ a } i > L \text{ dokud platí } q_i \leq CR. \\ x_{i,j,G} & \text{jinak} \end{cases} \quad (9)$$

Princip exponenciální křížení je znázorněn na obr. 2.



Obr. 2. Princip exponenciálního křížení, převzato z [50]

Nevýhody DE

Ačkoli lze metodu diferenciální evoluce obecně považovat za jednoduchou a robustní, může při nevhodně zvolené velikosti generace dojít ke stavu stagnace. Tento stav lze pozorovat především v případech, kdy optimalizace využívá malou velikost generace. V tom případě totiž může dojít snadno ke stavu, kdy je počet možných kombinací jedinců, které mohou případně nahradit aktuálního jedince, tak nízký, že nelze mezi nimi nalézt jedince s lepší účelovou funkcí. Pokud celá generace obsahuje jen takové jedince, dochází ke stagnaci algoritmu, i když je generace jedinců stále rozmanitá. Z toho důvodu se nedoporučuje volit velmi nízký počet jedinců a nepoužívat hraniční hodnoty řídicích konstant [51].

2.2.6.2 Modifikace DE

Metoda diferenciální evoluce je velmi efektivní algoritmus, který ukázal své schopnosti již od prvních publikací [37], kdy je třeba nastavit pouze několik řídicích konstant a velikost generace. Velikost generace spolu s nastavením řídicích konstant má ale vliv na jeho efektivitu (viz. např. problém stagnace výše) a nastavení je závislé na optimalizačním problému. Efektivitu algoritmu také ovlivňuje samotný výběr varianty výpočtu šumového vektoru \vec{v} .

Problém výběru varianty výpočtu šumového vektoru a nastavení řídicích konstant byl například řešen v [52], kde algoritmus SaDE vytváří jedince pomocí dvou různých způsobů generace šumového vektoru (*DE/rand1/bin* a *DE/current-to-best/2/bin*). Počet jedinců, kteří jsou takto jednotlivými způsoby generováni, se pak dle jejich předchozích výsledků mění. Algoritmus SaDE také mění nastavení řídicích konstant po definovaném počtu iterací.

Dalším přístupem k vylepšení rychlosti a konvergence algoritmu je nastavování řídicích konstant F a CR za běhu optimalizace. Algoritmus jDE [48, 53] nastavení řídicích konstant F a CR mění za běhu pomocí pravděpodobnosti jejich změny pro každého jedince zvlášť (tj. jedinec s číslem i bude generován pomocí mutační konstanty F_i a prahu křížení CR_i). V každé iteraci algoritmu se pak může mutační konstanta F_i a práh křížení CR_i náhodně měnit v definovaném rozsahu s pravděpodobností τ_1 resp. τ_2 . S přibývajícím počtem iterací je pak doporučeno zvyšovat dolní meze parametrů F a CR [54].

Efektivita algoritmu diferenciální evoluce také závisí na vyváženém poměru mezi prohledáváním celého optimalizačního prostoru a schopnosti efektivně prohledat lokální oblast optimalizačního prostoru. Tohoto vyvážení se některé metody snaží docílit sdružováním jedinců do skupin tzv. „sousedních“ jedinců. Jednotlivé skupiny pak mohou tvořit jednotlivé optimalizační celky, kde však některé informace z těchto jednotlivých skupin mohou být sdíleny mezi jednotlivými skupinami. Jako příklad metody DE využívající princip sousedství uvedme metodu DEGL [55].

Modifikací metody diferenciální evoluce bylo publikováno v posledních letech velmi mnoho. Souhrn jednotlivých modifikací DE z posledních let uvádí [48] a [56].

2.2.7 Rojení částic - PSO

Algoritmus rojení částic byl poprvé publikován R.Eberhartem a J. Kennedym v roce 1995 [32, 57]. Jeho myšlenkou bylo původně napodobení pohybu hejna ptáků (roje včel), kteří hledají nejvyšší místo v jejich okolí. Podstata metody je založena na sdílení informací o nejlepších místech celého hejna a jednotlivých jedinců.

Metoda pracuje s generací (populací) náhodně vytvořených jedinců obdobně jako ostatní evoluční algoritmy. Oproti ostatním evolučním algoritmům nevytváří nové jedince pomocí křížení, mutace a selekce, ale jedinci se pohybují optimalizačním prostorem určitou rychlostí \vec{v} . Rychlost jejich pohybu je ovšem upravována v každé iteraci na základě nejvýhodnější pozice každého jedince a nejvýhodnější nalezené pozice celého hejna (populace). Algoritmus lze popsat následujícím způsobem (označme opět aktuálního jedince indexem j):

Inicializace algoritmu

1. nastavení řídicích konstant algoritmu C_1, C_2 ,
2. vytvoření počáteční populace jedinců (pomocí generátoru náhodných čísel) a k nim příslušné počáteční vektory rychlostí \vec{v} ,
3. aktuální pozice jedinců se ohodnotí účelovou funkcí UF a tyto pozice se zapíší do paměti jako nejlepší pozice jedinců $\overrightarrow{x_{j,best}}$ spolu s jejich hodnotou účelové funkce,
4. určí se jedinec s nejlépeší účelovou funkcí, který označme $\overrightarrow{x_{Global\ best}}$.

Optimalizační proces

Pro všechny jedince populace se provede následující postup tvorby nového jedince.:

1. Jedinec $\overrightarrow{x_{j,G}}$ se přesune s rychlostí \vec{v} do nové pozice dle rovnice

$$\overrightarrow{x_{j,G+1}} = \overrightarrow{x_{j,G}} + \overrightarrow{v_{j,G+1}} . \quad (10)$$

2. Nová pozice jedince v nové generaci / iteraci algoritmu se ohodnotí účelovou funkcí $UF(\overrightarrow{x_{j,G+1}})$.
3. Hodnota účelové funkce nové pozice jedince $UF(\overrightarrow{x_{j,G+1}})$ se porovná jak s nejlepší hodnotou účelové funkce jedince $UF(\overrightarrow{x_{j,best}})$, tak i s nejlepší hodnotou účelové funkce populace $UF(\overrightarrow{x_{Global\ best}})$. Pokud je hodnota nového jedince výhodnější, zapíše se do paměti výhodných pozic spolu s hodnotou účelové funkce jedince (buď jen jako $\overrightarrow{x_{j,best}}$ nebo i $\overrightarrow{x_{Global\ best}}$).
4. Vypočte se nový vektor rychlosti pro novou iteraci dle vztahu:

$$\overrightarrow{v_{j,G+1}} = \overrightarrow{v_{j,G}} + C_1 r_1 (\overrightarrow{x_{j,best}} - \overrightarrow{x_{j,G}}) + C_2 r_2 (\overrightarrow{x_{Global\ best}} - \overrightarrow{x_{j,G}}) , \quad (11)$$

kde r_1 a r_2 jsou náhodná čísla vygenerovaná v intervalu $\langle 0, 1 \rangle$. S touto rychlostí se v další iteraci bude měnit pozice jedince $\overrightarrow{x_{j,G}}$ (výpočet nové rychlosti je vhodné provádět pro všechny jedince současně až po ukončení přesouvání všech jedinců).

Tento postup se pro všechny hodnoty opakuje do té doby, než je splněn požadavek na hodnotu účelové funkce (její minimalizaci, či maximalizaci).

Hodnoty řídicích konstant C_1 a C_2 doporučovali Eberhart a Kennedy nastavit rovné $C_1 = C_2 = 2$, aby výsledný koeficient zohlednění nejlepších jedinců byl v průměru roven jedné (při generování náhodných čísel r_1 a r_2 s normálním rozložením).

2.2.7.1 Modifikace PSO

V roce 1998 Eberhart a Shi publikovali úpravu výpočtu nové rychlosti jedince tak, aby bylo možné jedince brzdit či zrychlovat ve svém pohybu a do výpočtu rychlosti tedy přidali další řídicí konstantu, tzv. setrvačnost w (v anlg. originále inertia weight) [58]. Výpočet nové rychlosti jedince pak probíhá dle vztahu:

$$\overrightarrow{v_{j,G+1}} = w \cdot \overrightarrow{v_{j,G}} + C_1 r_1 (\overrightarrow{x_{j,best}} - \overrightarrow{x_{j,G}}) + C_2 r_2 (\overrightarrow{x_{Global\ best}} - \overrightarrow{x_{j,G}}). \quad (12)$$

Zároveň doporučili nastavovat konstantu setrvačnosti v rozmezí $w \in (0.9, 1.2)$. Vyšší hodnoty setrvačnosti než 1 podporují globální prohledávání, nižší pak lokální prohledávání. Dále také navrhují hodnotu setrvačnosti v průběhu iterací postupně snižovat.

Algoritmus PSO má pro vyšší hodnoty setrvačnosti a především konstant C_x tendenci zrychlovat jedince. Aby nedocházelo k velkému zrychlení, které by vedlo k rychlé změně hodnot jednotlivých proměnných každého jedince (a tedy rychlé konvergenci jedinců k hranici přípustných hodnot), je obvykle do algoritmu zavedena maximální rychlost v_{max} , se kterou se mohou jedinci pohybovat.

Další modifikací metody PSO je tzv. constriction factor χ [59], jehož velikost je obvykle navázána na velikosti konstant C_1 a C_2 . Úprava rychlosti jedinců pak probíhá dle vztahu:

$$\overrightarrow{v_{j,G+1}} = \chi [w \cdot \overrightarrow{v_{j,G}} + C_1 r_1 (\overrightarrow{x_{j,best}} - \overrightarrow{x_{j,G}}) + C_2 r_2 (\overrightarrow{x_{Global\ best}} - \overrightarrow{x_{j,G}})]. \quad (13)$$

Hodnota constriction factoru χ je obvykle volena dle vztahu:

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \quad (14)$$

kde $\varphi = C_1 + C_2$ a zároveň $\varphi > 4$.

Výše uvedené modifikace představují pouze shrnutí základních principů modifikací PSO. Odvozené modifikace PSO pak obvykle využívají parametry setrvačnosti a constriction factoru, které v průběhu optimalizačního procesu různým

způsobem upravují [60–62]. Zajímavou variantou úpravy z poslední doby je např. úprava inspirovaná Butterwortovou aproximací filtrů [63].

Některé varianty PSO nevyužívají ke své činnosti pouze jedno globální optimum, ale sdružují jedince do skupin tzv. sousedních částic. Tyto skupiny používají svá globální optima, která jsou platná pouze pro konkrétní skupinu. Sousední jedinci však nemusí být voleni podle geometrické blízkosti a také skupiny nemusí být složeny ze stále stejných jedinců [64, 65].

Principy PSO metody popsané výše vyžadují ovšem spojitost jednotlivých proměnných. Proto byly také vyvinuty modifikace pro diskrétní případy [66, 67].

2.2.8 Nelder – Mead simplexový algoritmus

Tento algoritmus byl publikován v roce 1965 J. A. Nelderem a R. Meadem [68] jako vylepšení algoritmu [69]. Nelder – Mead simplexový algoritmus (NM algoritmus) lze zařadit do kategorie algoritmů přímo prohledávající optimalizační prostor a jeho princip je založen na postupných úpravách simplexu tak, aby se hodnota účelové funkce minimalizovala. Jeho základem jsou transformace bodu s nejvyšší hodnotou účelové funkce pomocí tzv. centroidu (resp. těžiště). Principiální fungování algoritmu lze shrnout do následujícího postupu.

1. Je vygenerováno $(m+1)$ náhodných bodů (vektorů proměnných), které jsou následně seřazeny dle své hodnoty účelové funkce UF .

$$UF(\vec{X}_1) < UF(\vec{X}_2) < \dots < UF(\vec{X}_l) \dots < UF(\vec{X}_m) < UF(\vec{X}_{m+1}).$$

2. Z m bodů s menší hodnotou účelové funkce než je bod s největší hodnotou účelové funkce \vec{X}_{m+1} se vypočte centroid (těžiště) dle vztahu:

$$\vec{C} = \frac{1}{m} \sum_{i=1}^m \vec{X}_i. \quad (15)$$

3. Vypočte se reflexe \vec{R} bodu s největší hodnotou účelové funkce \vec{X}_{m+1} dle vztahu:

$$\vec{R} = \vec{C} + \alpha(\vec{C} - \vec{X}_{m+1}), \quad (16)$$

kde $\alpha > 0$ je nastavený koeficient reflexe obvykle v rozmezí $\alpha \in (0, 1)$. Pokud účelová funkce reflektovaného bodu leží v rozmezí $UF(\vec{X}_1) < UF(\vec{R}) < UF(\vec{X}_m)$, pak iterace algoritmu končí.

4. Pokud je účelová hodnota reflektovaného bodu menší než účelová funkce nejlepšího bodu $UF(\vec{R}) < UF(\vec{X}_1)$, pokračuje algoritmus expanzí dle předpisu

$$\vec{E} = \vec{C} + \gamma(\vec{R} - \vec{C}), \quad (17)$$

kde γ je nastavený koeficient expanze, kde musí platit: $\alpha < \gamma$ a zároveň $\gamma > 1$.

Pokud účelová funkce $UF(\vec{E}) < UF(\vec{R})$, pak je bod \vec{X}_{m+1} nahrazen bodem \vec{E} a iterace končí. V opačném případě je bod \vec{X}_{m+1} nahrazen bodem \vec{R} a iterace končí.

5. Pokud $UF(\vec{X}_m) < UF(\vec{R}) < UF(\vec{X}_{m+1})$, pokračuje algoritmus vnější kontrakcí dle předpisu

$$\vec{K} = \vec{C} + \beta(\vec{R} - \vec{C}), \quad (18)$$

kde β je nastavený koeficient kontrakce, pro který musí platit $\beta \in (0, 1)$. Pokud $UF(\vec{X}_{m+1}) < UF(\vec{R})$, pokračuje algoritmus vnitřní kontrakcí

$$\vec{K}' = \vec{C} - \beta(\vec{C} - \vec{X}_{m+1}) . \quad (19)$$

Po výpočtu kontrakce se porovná hodnota účelové funkce $UF(\vec{K})$ nebo $UF(\vec{K}')$ s hodnotou účelové funkce nejhoršího bodu $UF(\vec{X}_{m+1})$.

Pokud platí $UF(\vec{K}$ resp. $\vec{K}') < UF(\vec{X}_{m+1})$, pak \vec{K} resp. \vec{K}' nahradí bod \vec{X}_{m+1} v simplexu.

6. Pokud $UF(\vec{K}$ resp. $\vec{K}') > UF(\vec{X}_{m+1})$, dochází k zúžení simplexu, tj. body $\vec{X}_l \neq \vec{X}_1$ jsou přiblíženy k nejlepšímu bodu \vec{X}_1 dle předpisu:

$$\vec{X}_l' = \vec{X}_1 + \delta(\vec{X}_l - \vec{X}_1) , \quad (20)$$

kde konstanta δ se volí v rozsahu $\delta \in (0, 1)$.

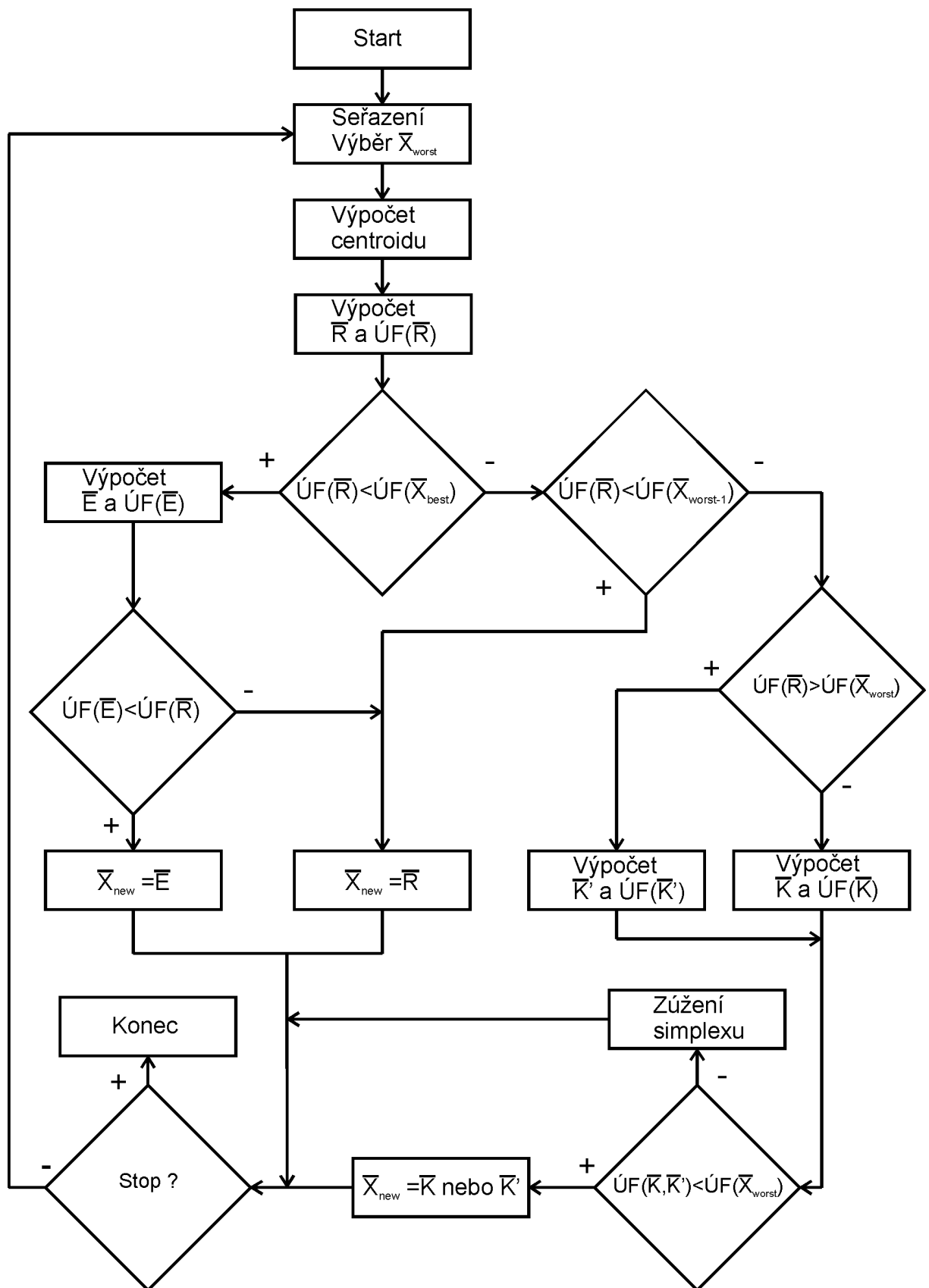
Postup se opakuje, dokud není splněna ukončovací podmínka, popř. dokud nejsou všichni jedinci stejní. Obvyklé nastavení řídicích konstant je: $\alpha = 1$, $\beta = 0.5$, $\gamma = 2$, $\delta = 0.5$. Vývojový diagram NM simplexového algoritmu je uveden na obr. 3.

2.2.8.1 Modifikace NM simplexového algoritmu

Ačkoli je simplexový algoritmus efektivním algoritmem pro minimalizaci funkcí, jeho efektivita s rostoucí dimenzí řešeného problému klesá [70]. Dle [71] lze například tuto vlastnost algoritmu potlačit vhodnou volbou řídicích konstant, jejichž velikost je navázána na dimenzi optimalizačního problému.

Největší nevýhoda NM algoritmu je ovšem v jeho sekvenčním průběhu, kdy se transformuje pouze bod s nejhorší účelovou funkcí a algoritmus nevyužívá paralelizace operací (s výjimkou zužování simplexu). V roce 2007 byla publikována metoda paralelizace celého algoritmu [72], kde se celý postup výpočtu reflexe, expanze a kontrakce provádí pro skupinu několika „nejhorších“ bodů. Další způsob paralelizace výpočtu simplexové metody, kdy autor dopředu vypočítává účelové funkce všech možných vývoju algoritmu (reflexe, expanze, kontrakce), aby si z řešení následně vybral, byl znázorněn v [73].

Zrychlením optimalizace elektronického obvodu (konkrétně optimalizace OZ) se zabývá i [25], kde je výpočet centroidu nahrazen váhovaným těžištěm pomocí účelové funkce jednotlivých bodů.



Obr. 3 Vývojový diagram MN simplexového algoritmu

2.2.9 Hybridní algoritmy DE, PSO, NM simplexu

Jednotlivé algoritmy DE, PSO a NM simplex jsou robustními algoritmy, které lze aplikovat na velkou škálu optimalizačních problémů. V předchozích kapitolách bylo uvedeno několik jejich vylepšení, která mají obvykle za cíl omezit vliv nastavení řídicích parametrů (případně je nastavovat adaptivně). Velmi často se však také využívají algoritmy, které kombinují více principů do jednoho funkčního celku (tzv. hybridní algoritmy).

Způsobů, jakými je možno spojit více principů, je mnoho. Nejčastějším principem je střídání použití jednotlivých metod, [43, 74, 75]. Dalším způsobem je využití obou metod zároveň [76].

Prezentované algoritmy DE, PSO a NM simplexu jsou velmi často navzájem hybridizovány tak, aby se obvykle průběh algoritmu zrychlil nebo se omezila možnost uvíznutí algoritmu.

Zástupcem algoritmu, spojujícího metodu DE a PSO je algoritmus HPSDE [43], kde metoda PSO je použita v případě, že algoritmus DE nenalezne pro aktuálního jedince lepší řešení. Jedinec generovaný pomocí PSO je ovšem přijat do nové generace pouze v případě, že hodnota jeho účelové funkce je výhodnější, než je pro aktuální pozici jedince. Obdobný princip algoritmu [43] je i algoritmus uvedený v [74].

Příkladem nezvyklého hybridního algoritmu spojujícího algoritmus DE a PSO je metoda [76], kde pro větší diverzitu jednotlivých řešení jsou jednotlivé proměnné nového jedince generovány s jistou pravděpodobností metodou DE a ostatní proměnné metodou PSO.

Spojením algoritmu DE a NM simplexového algoritmu se zabývá např. [77], kde simplexový algoritmus je spuštěn pro skupinu nejlepších jedinců generace v případě, že hlavní algoritmus DE stagnuje. Výsledkem simplexové metody je pak jedinec, který nahradí nejhoršího jedince celé populace.

Příkladem spojení simplexové metody spolu s metodou PSO je metoda [78].

2.3 Zhodnocení současného stavu problematiky

Návrhem postupů a optimalizací různých typů analogových obvodů se již zabývalo několik vědeckých prací (jak spojitých obvodů [10, 15, 25, 79], tak periodicky spínaných [2–5]). Pro periodicky spínané obvody, které pracují v krajních podmínkách (např. funkce až do poloviny vzorkovacího kmitočtu) se výrazně zvyšuje vliv neidealit na výsledné vlastnosti obvodu. V případě vlivu nelinearit je třeba celý obvod analyzovat nejprve v časové oblasti (nejlépe na tranzistorové úrovni). Kmitočtové vlastnosti systému lze pak získat následnou transformací časové odezvy. Pokud se jedná o rozsáhlejší obvodový celek, může celá analýza výpočetnímu programu trvat i několik desítek minut.

Při návrhu takových obvodů pak návrhář nutně musí volit postup, kdy navrhovaný obvod bude optimalizovat až poté, kdy již v průběhu návrhu uvažoval i vliv některých neideálních parametrů (například vliv konečného zesílení a šířky pásma operačního zesilovače) [2, 80].

Stále je potřeba výsledný návrh na tranzistorové úrovni optimalizovat, což je výpočetně a časově náročné [2]. Problematikou optimalizace spínaných struktur se zabýval [4, 5, 13], kde však byl problém optimalizace řešen pouze v rámci lineárního spínaného obvodu, tj. pouze při aplikaci zjednodušeného lineárního modelu operačního zesilovače, či tranzistoru. Navíc se autor v práci nezabývá spolehlivostí jeho řešení.

Při optimalizaci reálného obvodu včetně nelinearit je z výše zmíněných důvodů potřeba zvolit algoritmus s rychlou konvergencí ke globálnímu optimu (popř. alespoň k dostačujícímu lokálnímu extrému) a dostatečnou spolehlivostí při malém počtu potřebných simulací.

3 Cíle disertační práce

Aplikace optimalizačních algoritmů na reálné spínané obvody je obtížná a není příliš zmapována (viz kapitola 2.3). Tato práce navazuje na výsledky školícího pracoviště v této oblasti, kde se vhodnou skupinou algoritmů pro optimalizovaný návrh na čipu ukázala skupina evolučních algoritmů a mapuje a navrhuje postupy optimalizace na reálném periodicky spínaném obvodě.

Cíle této práce lze shrnout do následujících bodů:

- prozkoumat a porovnat vlastnosti různých evolučních algoritmů při aplikaci na reálné periodicky spínané obvody,
- navrhnout optimalizační algoritmus tak, aby se zlepšily jeho vlastnosti pro použití na reálných periodicky spínaných obvodech,
- navržený optimalizační algoritmus aplikovat na návrh konkrétního periodicky spínaného analogového obvodu na tranzistorové úrovni v reálné technologii a zhodnotit rychlost konvergence a spolehlivost navrhovaných optimalizačních algoritmů.

4 Výsledky disertační práce

Pro porovnání optimalizačních metod byla zvolena implementace dvoukanálové banky filtrů pomocí obvodů se spínanými kapacitami [A6]. Ačkoliv jsou banky filtrů především využívány v číslicových obvodech [81, 82], jak je ukázáno v [83] a [A1, A3], je možná i její implementace pomocí techniky spínaných kapacit. SC obvod zde pracuje v krajním případě, kdy zpracovává vstupní signály v kmitočtovém pásmu až do poloviny vzorkovacího kmitočtu.

Zhodnocení možných optimalizačních kritérií bylo nejprve provedeno na implementaci banky třetího řádu na plošném spoji, kde se ukázala nutnost implementovat obvod na čipu. Výsledky měření na realizaci pak byly použity pro formulaci účelové funkce pro optimalizaci jednotlivých částí banky filtrů.

Dále byl proveden návrh banky filtrů čtvrtého řádu na čipu, který řeší některé problémy při implementaci pomocí techniky spínaných kapacit, a která opět zpracovává signály s kmitočtovým rozsahem až do poloviny vzorkovacího kmitočtu.

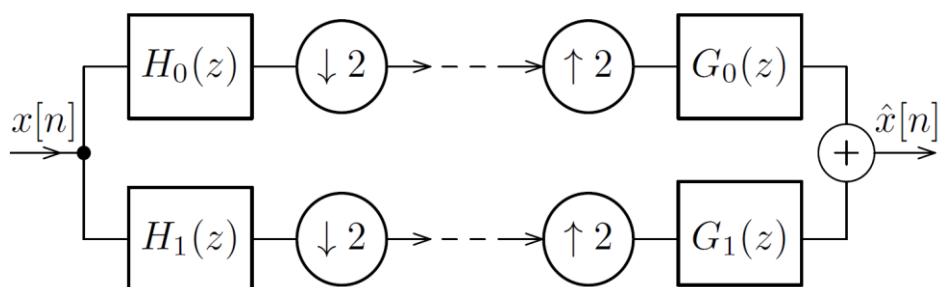
Na návrhu tohoto obvodu jsou porovnávány jednotlivé evoluční algoritmy a jejich navrhované hybridní algoritmy, a to jak z hlediska spolehlivosti, tak rychlosti konvergence.

Výsledky disertační práce jsou v této kapitole členěny následujícím způsobem.

- V podkapitole 4.1 je nejprve uvedena banka filtrů třetího řádu a především aspekty její implementace na plošném spoji.
- Podkapitola 4.2 ukazuje aspekty optimalizačních úloh a diskutuje vliv struktury operačních zesilovačů pro spínané obvody.
- V podkapitole 4.3 je ukázán návrh jak samotné banky čtvrtého řádu, tak i její implementace na čipu v technologii AMS 0.35 μm .
- Podkapitola 4.4 obsahuje návrhy hybridních algoritmů pro optimalizaci periodicky spínaných systémů na čipu.
- V podkapitole 4.5 jsou zhodnoceny vlastnosti jednotlivých optimalizačních algoritmů, a to jak z hlediska spolehlivosti, tak z hlediska rychlosti konvergence.

4.1 Implementace banky filtrů třetího řádu

Banka filtrů, jejíž návrh je podrobně rozebrán [83] [A1], rozkládá vstupní vzorkovaný signál na dva subpásmové signály s polovičním vzorkovacím kmitočtem. Tyto signály jsou poté syntetizující částí banky opět složeny do signálu s původním vzorkovacím kmitočtem $f_{clk} = 16$ kHz. Blokové schéma implementované banky filtrů je uvedeno na obr. 4.



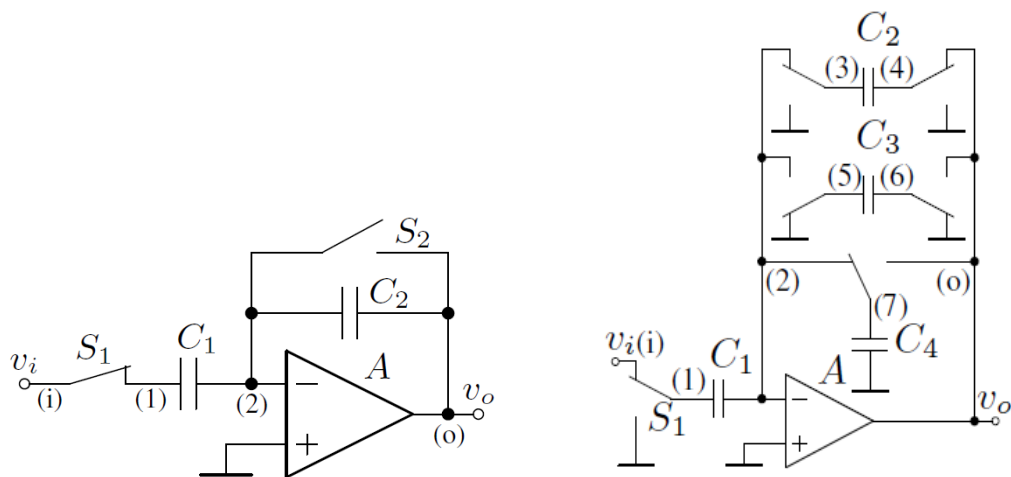
Obr. 4 Blokové schéma implementované banky filtrů třetího řádu

Celkový přenos banky filtrů je konstantní pro celý kmitočtový rozsah zpracovávaného vstupního signálu. Jednotlivé bloky realizují přenosové funkce H_0 , H_1 a G_0 , G_1 dle následujícího předpisu

$$H_0(z) = \frac{1}{2} G_0(z) = \frac{1}{2} \frac{0.5z^3 + z^2 + z + 0.5}{z(z^2 + 0.5)}, \quad (21)$$

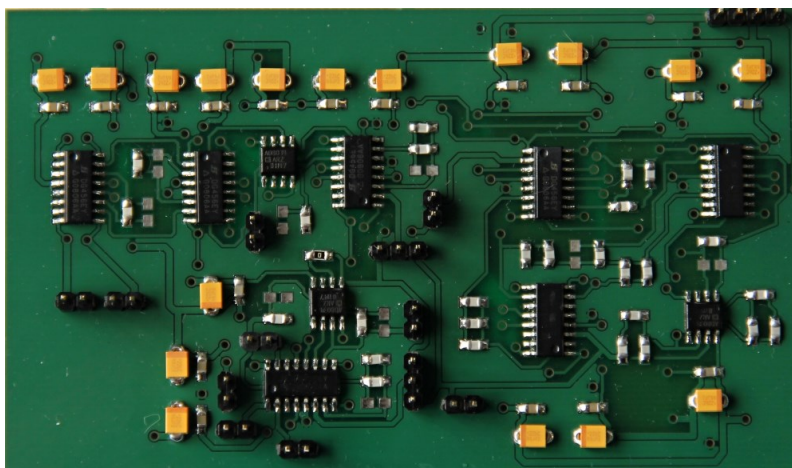
$$H_1(z) = -\frac{1}{2} G_1(z) = -\frac{1}{2} \frac{-0.5z^3 + z^2 - z + 0.5}{z(z^2 + 0.5)}, \quad (22)$$

Implementace filtrů realizujících přenosové funkce H_0 , H_1 resp, G_0 a G_1 byla provedena pomocí kaskádně řazených SC filtrů druhého a prvního řádu. Bloky druhého řádu byly realizovány pomocí Fleischer-Laker bikvadratické sekce (viz obr. 10 dále) a bloky prvního řádu (derivátor a „integrátor“) pomocí bloků, jejichž obvodové schéma je uvedeno na obr. 5



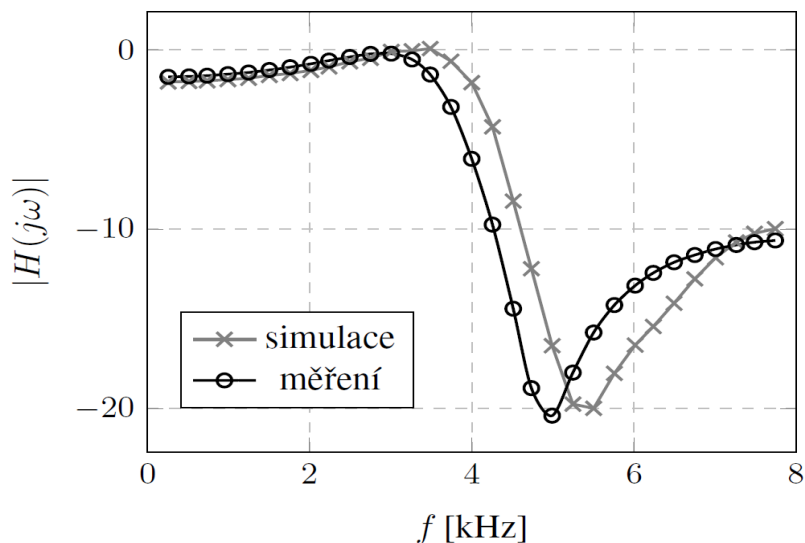
Obr. 5 Obvodové schéma SC derivátoru (vlevo) a „integrátoru“ (vpravo) při implementaci banky filtrů třetího řádu

Realizace celého obvodu a výsledky měření na desce plošného spoje (viz obr. 6) jsou podrobně rozebrány v [A3]. Při měření se ukázalo, že diskretní implementace SC banky filtrů je problematická. Při použití diskretních spínačů bylo totiž nutné nastavovat časování spínačů s tzv. dead-timem v řádu několika desítek nanosekund tak, aby nedocházelo k situaci, kdy by byly spínače řízené z různých fází hodinového signálu sepnuty současně.



Obr. 6 Realizace banky filtrů třetího řádu na desce plošného spoje

Dalším problémem při realizaci banky bylo obtížné nastavení hodnoty kapacit tak, aby byla výsledná kmitočtová charakteristika jednotlivých bloků v toleranci vůči ideální kmitočtové charakteristice (viz obr. 7, kde je zobrazena posunutá kmitočtová charakteristika realizované bikvadratické sekce typu dolní propust v porovnání s ideálním průběhem).



Obr. 7 Porovnání měřené kmitočtové charakteristiky bikvadratické sekce typu dolní propust s ideálním průběhem kmitočtové charakteristiky

Při ověřování funkce dvoukanálové SC banky třetího řádu [A1, A3] na plošném spoji se ukázal vliv konkrétní implementace sekcí prvního řádu, kde správná funkce „integrátoru“ velmi závisela na přesném časování diskretních spínacích prvků [A3]. Použitý operační zesilovač navíc nedisponoval dostatečnou fázovou jistotou a výstup měl tendenci zakmitávat po přechodech mezi jednotlivými fázemi časovacích signálů.

Nejen výše uvedené implementační aspekty, kdy problematický byl především blok realizující funkci integrátoru, ukázaly, že obvod tohoto typu, který zpracovává vstupní signály až do poloviny vzorkovacího kmitočtu, je třeba implementovat a optimalizovat na čipu.

Zejména z výše uvedených důvodů (problémy při realizaci bloků prvního řádu) bude v podkapitole 4.3 navržena dvoukanálová banka filtrů čtvrtého řádu. Před ní je však ještě zařazena podkapitola 4.2 diskutující praktické aspekty optimalizace reálné struktury a vliv parametrů operačního zesilovače na spínané obvody.

4.2 Praktické aspekty optimalizací reálných struktur

Při optimalizaci reálných analogových struktur pomocí spojení simulačního prostředí a řídicího optimalizačního programu je třeba řešit i praktické aspekty běhu optimalizačního algoritmu. Vyjmenujme nyní několik těchto aspektů:

- Simulační nástroj musí být schopen výsledky simulace předat ve strojově čitelném formátu (tabulka, seznamu hodnot).
- Obdobně dle předchozího bodu musí být i optimalizační nástroj schopen předat nové parametry simulátoru (hodnoty proměnných).
- Nastavení jednotlivých parametrů simulátoru musí být takové, aby neovlivnilo výsledky simulace, ale zároveň také umožnilo simulátoru snadněji konvergovat i pro hraniční řešení. Například svodová vodivost mezi uzly g_{\min} v simulačních nástrojích bývá nastavena na hodnotu v řádech 10^{-12} S. Z autorovy zkušenosti simulací SC obvodů je obvykle pro lepší konvergenci simulací potřeba tento parametr nastavit o něco výše. Autor ve svých simulacích používal hodnotu 10^{-9} S. Dalším příkladem je vhodné nastavení simulační metody (pro SC obvody lze doporučit *gear* metodu).
- Při spojení řídicího optimalizačního programu se simulátorem by měl být zajištěn mechanismus předávání informace o ukončení simulace a výsledku simulace. Pokud tedy simulace nekonverguje, je třeba i tuto informaci předat optimalizačnímu nástroji, který by na ni měl umět reagovat.
- Při paralelních simulacích musí být zajištěn mechanismus, který zajistí, aby výsledky byly ukládány separátně a bylo možné od sebe rozlišit jednotlivé simulace.

Autor při své práci využíval jako řídicí program optimalizačního procesu systému Maple [84], kde simulace reálného obvodu pak realizoval simulátor WinSpice [85]. Výše zmíněné aspekty spojení dvou programů také autor při své práci řešil.

4.2.1 Vliv kvality generování náhodných čísel

Pro stochastické algoritmy je zásadní kvalita náhodných čísel, které se používají při běhu algoritmu. Vzhledem k principu algoritmu diferenciální evoluce, kdy se nový jedinec vytváří pomocí diference alespoň dvou dalších vektorů, je tato metoda potenciálně citlivá na lineární nezávislost jednotlivých vektorů, obzvláště v případě, kdy je populace relativně malá.

Proto je nutné, aby byli jedinci vytvářeni pomocí kvalitního generátoru náhodných čísel. Zvláště při použití malé velikosti populace NP je tedy dobré po vygenerování první generace zkontrolovat, zda jsou jedinci lineárně nezávislí. Jedinci by také měli pokrývat celý optimalizační prostor „rovnoměrně“.

Vzhledem k předpokládané reprezentaci jednotlivých jedinců vektory (popř. řádky nebo sloupce dvourozměrného pole, či matice), lze kontrolu nezávislosti vektorů provést velmi snadno a rychle (především v matematických programech).

Autor ve své práci využíval jak interního generátoru náhodných čísel řídicího programu implementovaného v systému Maple [84], tak i externího HW generátoru náhodných čísel [86].

4.2.2 Operační zesilovače využívané v SC obvodech

Operační zesilovač je důležitou součástí moderních elektronických systémů. Na jeho reálných parametrech pak závisí vlastnosti jednotlivých analogových částí obvodu. Při návrhu SC obvodu je důležité dbát především na tyto parametry:

- stejnosměrné zesílení (zisk) $A_0[-]$,
- šířka pásma f_0 [MHz],
- maximální hodnota zatěžovací kapacity $C_{L\ max}$ [pF],
- rychlost přeběhu SR [V/μs],
- doba ustálení $t_{settling}$ [s] (popř. také doba zotavení),
- vstupní napěťová nesymetrie V_{off} ,
- vstupní klidové proudy I_{in+} [μA], I_{in-} [μA],
- a další...

Ze jmenovaných parametrů je nejdůležitějších prvních pět.

Budeme-li mluvit o návrhu SC filtrů, pak lze v literatuře najít obecná doporučení na velikost zisku a především šířky pásma, tak aby byl jejich vliv zanedbatelný. Jedná se především o doporučený minimální poměr spínacího kmitočtu k šířce pásma zpracovávaného signálu, dále pak poměr šířky pásma operačního zesilovače ke spínacímu kmitočtu [80]. Pokud nelze tato doporučení splnit, pak je nutné výsledný obvod optimalizovat (viz např.[4, 80]).

Uvažujeme-li vnitřní topologii OZ, pak se nejčastěji používají jedno či dvoustupňové OZ tak, aby se pokud možno minimalizoval vliv parazitních (nedominantních) pólů a nul přenosu a celé zpětnovazební zapojení bylo stabilní (každý

další stupeň do obvodu přidává další póly a nuly, které mohou vést k nestabilitě celého systému).

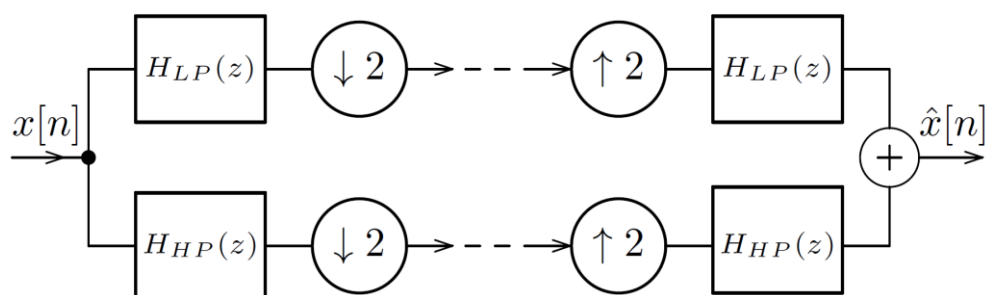
Při návrhu SC obvodů by se z výše zmíněných důvodů mohlo na první pohled zdát výhodné použití jednostupňového transkonduktančního operačního zesilovače (OTA), kde dominantní pól je obvykle závislý na velikosti kapacitní zátěže zesilovače. To ovšem znamená, že pro požadovanou hodnotu rychlosti přeběhu musí diferenčním párem protékat dostatečně velký proud (závislý na velikosti zátěžové kapacity).

Další nevýhodou jednostupňového zesilovače typu OTA je velikost zesílení samotného zesilovače, kterou je třeba zajistit pro správnou funkci SC obvodu. Toto zesílení lze do jisté míry navýšit kaskodováním proudového zrcadla, které tvoří dynamickou zátěž diferenčního páru, a zvyšováním transkonduktance samotných tranzistorů diferenčního páru. Zvětšení transkonduktance tranzistorů diferenčního páru ovšem vede k celkovému zvětšení velikosti těchto tranzistorů a tedy k velké vstupní kapacitě operačního zesilovače [80].

Z těchto důvodů se tedy při návrhu SC obvodů setkáváme nejčastěji se dvoustupňovými zesilovači s relativně jednoduchou vnitřní strukturou - buď Milerovským typem OZ, či kaskodovaným zesilovačem s výstupním bufferem. Výstupní buffer je pak buď zesilovač se společným sourcem nebo drainem (častější varianta).

4.3 Návrh SC banky filtrů čtvrtého řádu

Při měření na bance filtrů třetího řádu se ukázaly především nevýhody použití bloků prvního řádu, kde např. byl „integrátor“ velmi citlivý na přesné nastavení časovacích signálů (viz. kapitola 4.1). Proto byla navržena banka filtrů čtvrtého řádu.



Obr. 8. Blokové schéma navržené banky filtrů čtvrtého řádu

Navržená dvoukanálová banka filtrů čtvrtého řádu je blokově znázorněna na obr. 8, a obdobně jako u banky třetího řádu obsahuje filtry typu dolní propust (označených H_{LP}) a filtry typu horní propust (H_{HP}). Tyto filtry rozkládají vstupní diskrétní signál se

vzorkovacím kmitočtem 16 kHz na dva subpásmové signály se vzorkovacím kmitočtem 8 kHz. Signály pak mohou být přeneseny s nižší vzorkovacím kmitočtem, nebo nějakým způsobem zpracovány, a poté opět syntetizující částí banky převzorkovány a složeny opět do signálu se vzorkovacím kmitočtem 16 kHz. Přenos celé soustavy by měl být v absolutní hodnotě roven jedné (fázová charakteristika je vzhledem k IIR implementaci nelineární).

Přenosové funkce filtrů typu dolní propust H_{LP} a H_{HP} byly odvozeny v [A6] a jsou v následujícím tvaru:

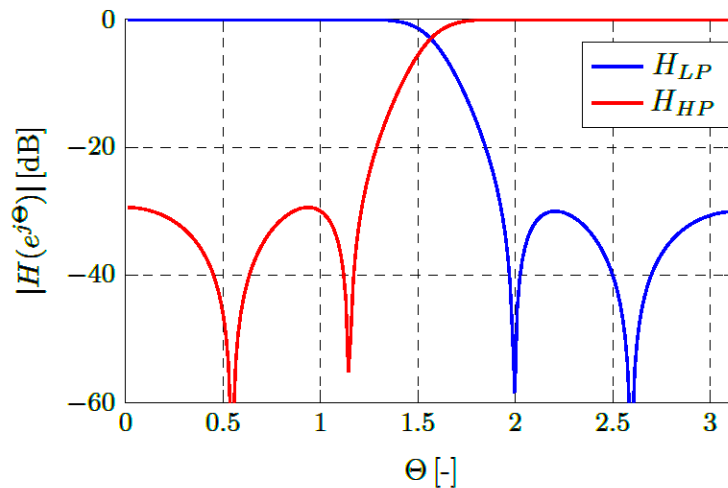
$$H_{LP}(z) = \frac{0.167z^4 + 0.423z^3 + 0.569z^2 + 0.423z + 0.167}{z^4 - 0.0205z^3 + 0.721z^2 - 0.0084z + 0.0576}, \quad (23)$$

$$H_{HP}(z) = \frac{0.173z^4 - 0.437z^3 + 0.587z^2 - 0.437z + 0.173}{z^4 - 0.0205z^3 + 0.721z^2 - 0.0084z + 0.0576}. \quad (24)$$

Amplitudové kmitočtové charakteristiky jednotlivých filtrů jsou vykresleny na obr. 9. Přenosové funkce byly pro kaskádní implementaci pomocí bikvadratických sekcí rozděleny na dvě funkce druhého řádu následovně:

$$H_{LP}(z) = H_{LP1}(z) H_{LP2}(z) = \frac{0.409z^2 + 0.336z + 0.409}{z^2 - 0.01226z + 0.092} \frac{0.409z^2 + 0.698z + 0.409}{z^2 - 0.0082z + 0.629}, \quad (25)$$

$$H_{HP}(z) = H_{HP1}(z) H_{HP2}(z) = \frac{0.415z^2 - 0.710z + 0.415}{z^2 - 0.0082z + 0.629} \frac{0.415z^2 - 0.341z + 0.415}{z^2 - 0.0122z + 0.092}. \quad (26)$$



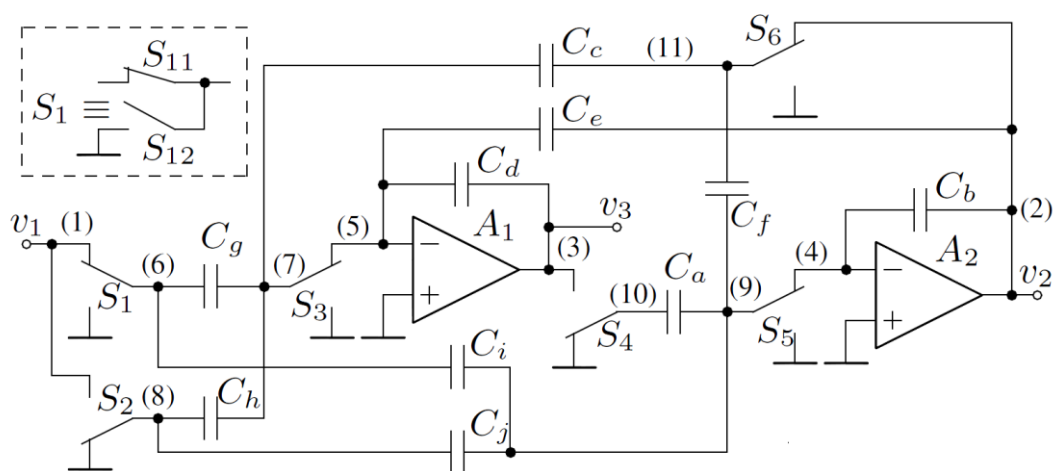
Obr. 9. Amplitudové kmitočtové charakteristiky navržených přenosových funkcí

4.3.1 Implementace bikvadratických sekcí

Pro implementaci přenosových funkcí (25) a (26) na čipu pomocí obvodu se spínanými kapacitami byla zvolena Fleischer-Laker topologie [19] znázorněná na obr. 10.

Tato topologie je tzv. stray insensitive (potlačuje vliv parazitních kapacit) a umožňuje implementaci všech přenosových funkcí 2. řádu.

Normované hodnoty kapacit pro jednotlivé přenosové funkce banky 4. řádu jsou uvedeny v tab. 2.



Obr. 10. Fleischer-Laker topologie bikvadratické sekce se spínanými kapacitami

Pro správnou funkci celé banky je nutné, aby přenos řetězce filtrů banky (jak analyzující části, tak syntetizující části) byl splněn s přípustnou tolerancí. Každou bikvadratickou sekci je tedy třeba navrhnut a optimalizovat s tolerancí tak, aby výsledná přenosová charakteristika obvodu nebyla příliš ovlivněna.

Tab. 2. Normované hodnoty kapacit pro idealizované bikvadratické sekce

| | H_{LP1} | H_{LP2} | H_{HP1} | H_{HP2} |
|-------|-----------|-----------|-----------|-----------|
| C_a | 1.25 | 1.25 | 1.25 | 1.25 |
| C_b | 0.591 | 0.628 | 0.628 | 0.591 |
| C_c | 0.863 | 1.296 | 1.296 | 0.863 |
| C_d | 1 | 1 | 1 | 1 |
| C_e | 0.4 | 0 | 0 | 0.4 |
| C_f | 0.408 | 0.371 | 0.371 | 0.408 |
| C_g | 1.422 | 1.712 | 0.597 | 0.892 |
| C_h | 0.5 | 0.5 | 0.5 | 0.5 |
| C_i | 0.408 | 0.408 | 0.415 | 0.415 |
| C_j | 1.033 | 1.033 | 1.040 | 1.040 |

4.3.2 Návrh operačního zesilovače pro banku filtrů

Pro správnou funkci obvodů se spínanými kapacitami je třeba volit operační zesilovač s vysokým zesílením a velkou šířkou pásma tak, aby jeho parametry významně neovlivnily funkci obvodu (viz kapitoly 4.2.2 a 4.3). Funkce SC obvodu je ale závislá i na velko-signálových parametrech (např. rychlosti přeběhu, době ustálení...), jejichž vliv lze dopředu odhadnout jen obtížně. Dalším problémem může být i rozptyl parametrů operačního zesilovače či použitých spínačů.

Při realizaci SC banky filtrů třetího řádu na plošném spoji byl zvolen diskretní operační zesilovač s velkým zesílením (cca 92dB) a šířkou pásma (80MHz pro neinvertující zapojení) [89] - i přes to, že spínací kmitočet byl „pouhých“ 16 kHz. Při praktické realizaci se ukázalo, že i když byly parametry OZ při simulaci dostačující, při konkrétní implementaci s diskretními spínači a s požadovanou funkcí banky filtrů do poloviny vzorkovacího kmitočtu by bylo třeba tyto parametry ještě vylepšit a to včetně fázové jistoty OZ.

Byla stanovena maximální odchylka od idealizovaného průběhu kmitočtové charakteristiky pro jednu bikvadratickou sekci na 10 mdB (viz. kapitola 4.3.3). Analýzou idealizovaného obvodu, kde se započítávaly pouze odpory spínačů v sepnutém stavu, stejnosměrný zisk A_0 a šířka pásma operačního zesilovače f_0 , bylo zjištěno, že součin stejnosměrného zesílení a šířky pásma musí být

$$A_0 \cdot f_0 \geq 800 \cdot 10^9 \quad (27)$$

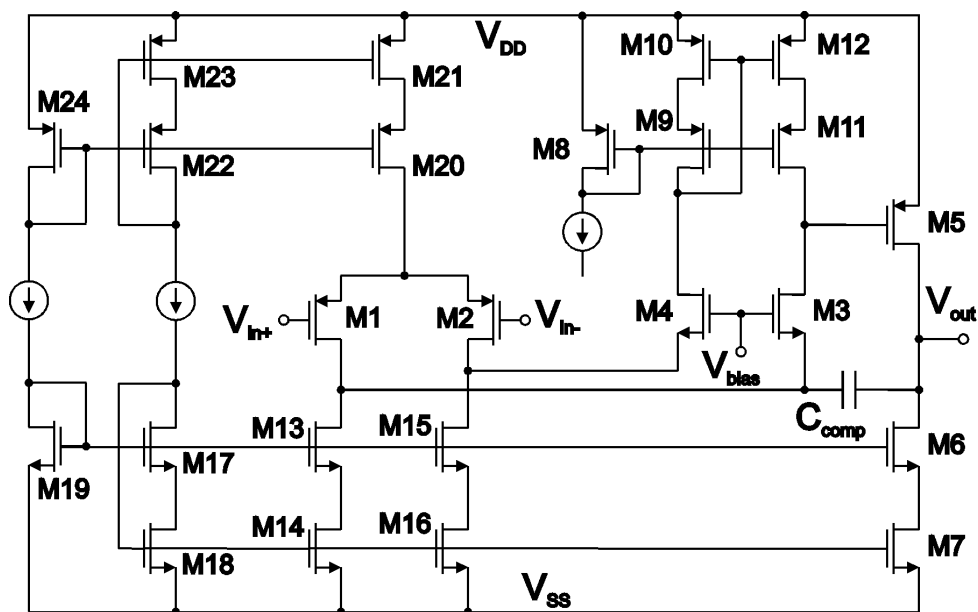
a kvůli poklesu zisku operačního zesilovače by šířka pásma měla být alespoň

$$f_0 \geq 10 \cdot \text{MHz}. \quad (28)$$

Vzhledem k přísným požadavkům na uvedené parametry OZ, jejich rozptylu a především vlivu dalších parametrů diskretních OZ na funkci SC obvodu, je vhodné celý systém implementovat na tranzistorové úrovni.

Návrh banky filtrů byl tedy proveden na čipu v technologii AMS 0.35 μm , včetně operačního zesilovače. Klíčovým parametrem pro návrh OZ byla šířka pásma větší než 10 MHz a fázová jistota alespoň 60°.

Topologií realizovaného operačního zesilovače byla zvolena přeložená kaskoda s výstupním bufferem zobrazená na obr. 11 [87, 88]. Pro implementaci byla zvolena varianta se vstupním diferenčním párem z tranzistorů P-MOS, protože s ní bylo možno dosáhnout vyšší hodnoty fázové jistoty.



Obr. 11 Topologie navrženého operačního zesilovače

Výsledné parametry navrženého operačního zesilovače jsou následující:

$$\begin{aligned}
 A_0 &= 75.3 \text{ dB}, & \varphi_{\text{marg}} &= 73^\circ, \\
 f_0 &= 12.7 \text{ MHz}, & V_{\text{off}} &= 58 \mu\text{V}, \\
 t_{\text{settling}} &= 200 \text{ ns}, & C_{L \text{ max}} &= 3 \text{ pF}
 \end{aligned}$$

Z parametrů navrženého operačního zesilovače je patrné, že jeho stejnosměrné zesílení není dostatečné a spolu s dalšími parametry (jak lineárními tak nelineárními) bude ovlivňovat funkci jednotlivých bikvadratických sekcí. Rozměry jednotlivých tranzistorů jsou uvedeny v tab. 3.

Tab. 3 Rozměry tranzistorů operačního zesilovače z obr. 11

| Tranzistor | Rozměry W / L [μm] |
|-----------------|------------------------------------|
| M1,M2 | 450/5 |
| M3,M4 | 100/0.35 |
| M5 | 88/0.7 |
| M6,M7 | 36/0.7 |
| M8 | 15/0.35 |
| M9,M10,M11,M12 | 60/0.35 |
| M13,M14,M15,M16 | 72/0.7 |
| M17,M18 | 18/0.7 |
| M19 | 12/2 |
| M22,M23 | 124/0.35 |
| M24 | 44/2 |

4.3.3 Optimalizace navržené banky filtrů

Jak již bylo uvedeno, absolutní hodnota přenosu celé banky filtrů by měla být rovna jedné v celém pracovním kmitočtovém pásmu, přičemž analyzující část banky rozděluje signál do dvou kmitočtových pásem, která jsou přenášena odděleně (viz obr. 8). Aby byla zachována tato vlastnost, je při optimalizaci nutné do účelové funkce promítnout jak podmínku hodnoty celkového přenosu, tak i podmínku kmitočtového oddělení.

Z hlediska náročnosti analýzy obvodu a tedy celkové doby optimalizace je také vhodné úlohu optimalizace banky filtrů rozdělit na menší funkční celky a ty optimalizovat separátně. Tím lze celou úlohu optimalizace do značné míry paralelizovat a využít. Takový postup je pak i při běhu optimalizačního algoritmu výhodnější, neboť klesá velikost optimalizačního prostoru a tím i potřebný čas k nalezení optima.

Zkrácení simulačního času (a tím i času optimalizace) lze také dosáhnout použitím nelineárních behaviorálních modelů jednotlivých částí spínaného systému. Pokud bychom pro operační zesilovač a ostatní součásti použili místo tranzistorového obvodu jeho model výsledný čas optimalizace klesne (z toho důvodu také např. [5] urychluje optimalizaci dosazováním do semisymbolického vztahu).

Při testování optimalizačních algoritmů uvedených dále (kapitola 4.4) autor této práce obvod zjednodušil. Při implementaci spínačů pomocí tzv. analogového spínače (paralelní spojení PMOS a NMOS transistoru jako spínače) byl čas jedné simulace přibližně 2 minuty (tzn. jedna iterace algoritmu pro 45 jedinců na 4 jádrovém počítači při paralelním výpočtu generace a 45 jedincích trvá přibližně 22,5 minut). Ukázalo se, že spínač lze pro uvažovanou aplikaci nahradit s postačující přesností pomocí lineárního modelu. Spínač byl proto pro účely hodnocení algoritmů nahrazen idealizovaným spínačem, který modeloval odpor v sepnutém i rozepnutém stavu a parazitní kapacity spínače (také nábojovou injekci).

V případě navrhované banky jsou optimalizovány separátně jednotlivé bikvadratické sekce, kdy amplitudová kmitočtová charakteristika každé sekce musí být s definovanou tolerancí blízko ideálnímu průběhu pro každou sekci. Z důvodu kaskádního řazení jednotlivých sekcí a funkce banky filtrů jako celku byla zvolena podmínka, kdy se amplitudová kmitočtová charakteristika každé bikvadratické sekce nesmí od ideální lišit o více než 10 mdB.

Vzhledem k tomu, že normované hodnoty kapacitorů pro jednotlivé bikvadratické sekce byly již stanoveny v kapitole 4.3.1, lze předpokládat, že výsledné hodnoty kapacit budou po optimalizaci v blízkém okolí těchto hodnot. Optimalizační procedura tedy nemusí generovat velikosti kapacitorů zcela náhodně. Hodnoty kapacit jsou pro počáteční generaci jedinců vytvářeny náhodně v intervalu $\pm 10\%$ od ideální hodnoty kapacity (s normálním rozložením).

4.4 Navržené hybridní algoritmy pro periodicky spínané obvody

Jak již bylo ukázáno v kapitole 2.2, evolučních algoritmů spolu s jejich modifikacemi popřípadě hybridizací bylo již navrženo velké množství. Každá z metod je více či méně vhodná na řešení určité skupiny problémů.

Při optimalizaci reálných periodicky spínaných obvodů je třeba při výběru algoritmu brát v úvahu i nezanedbatelný čas vyhodnocení účelové funkce (čas simulace). Proto by algoritmy aplikované na reálné spínané obvody měly být schopné jak globálního prohledávání optimalizačního prostoru, tak i lokálního prohledávání při nízkém počtu vyhodnocení účelové funkce. Právě podmínka nízkého počtu potřebných simulací je klíčovou pro výběr vhodného algoritmu.

Pro zrychlení konvergence evolučních algoritmů se obvykle využívá úprava řídicích konstant za běhu algoritmu (viz např. jDE a SaDE popsané v kapitole 2.2.6.2). Tento způsob může pomoci v situaci, kdy byly řídicí konstanty při inicializaci algoritmu zvoleny nevhodným způsobem. V případě požadavku nízkého počtu iterací tento typ vylepšení konvergence algoritmu ale není příliš vhodný. Z principu totiž potřebuje jistý počet iterací algoritmu, než mechanismus nastavení řídicích konstant určí vhodné hodnoty těchto konstant.

Z výše zmíněného důvodu vyplývá, že vhodnějším způsobem zlepšení konvergence a spolehlivosti algoritmu je v případě výpočetně náročných vyhodnocení účelových funkcí použití hybridních metod, které kombinují výhodné vlastnosti jednotlivých algoritmů. Proto se autor zaměřil ve své práci na návrh hybridních evolučních algoritmů.

4.4.1 Algoritmus DE-PSO1

Jak bylo již uvedeno v kapitole 2.3, lze diferenciální evoluci považovat za vhodnou metodu pro řešení optimalizace periodicky spínaných obvodů. Při nízkém počtu jedinců populace (což je na první pohled žádoucí) má ovšem tento algoritmus tendenci ke

stagnaci, kdy se účelová funkce po dlouhou dobu nemění, popř. jedinci celé populace předčasně konvergují k lokálnímu extrému účelové funkce [51]. Stagnaci algoritmu při použití hraniční velikosti generace se snaží předejít autorem navržený hybridní algoritmus DE-PSO1 [A2,A4,A5].

Algoritmus DE-PSO1 se snaží omezit možnou stagnaci metody *DE/rand-to-best/1/bin* tím, že při neúspěšném hledání lepšího bodu pro konkrétního jedince po určitém počtu iterací začne jedince přesouvat v optimalizačním prostoru pomocí principů metody rojení částic (PSO). Autorovi se osvědčilo přesouvání jedince pomocí PSO po třetím neúspěšném pokusu o nalezení nástupce aktuálního jedince s lepší hodnotou účelové funkce.

Algoritmus PSO byl pro hybridizaci s DE vybrán především pro svou vlastnost efektivně nalézt oblast s předpokládaným globálním optimem a také proto, že z principu jeho činnosti lze předpokládat, že posouvání jedinců bude vykonávat v rozsahu přípustných hodnot proměnných a tak omezí případy, kdy bude nutné nepřipustné hodnoty proměnných nějak penalizovat, či dokonce znovu generovat proměnnou v definovaném rozsahu.

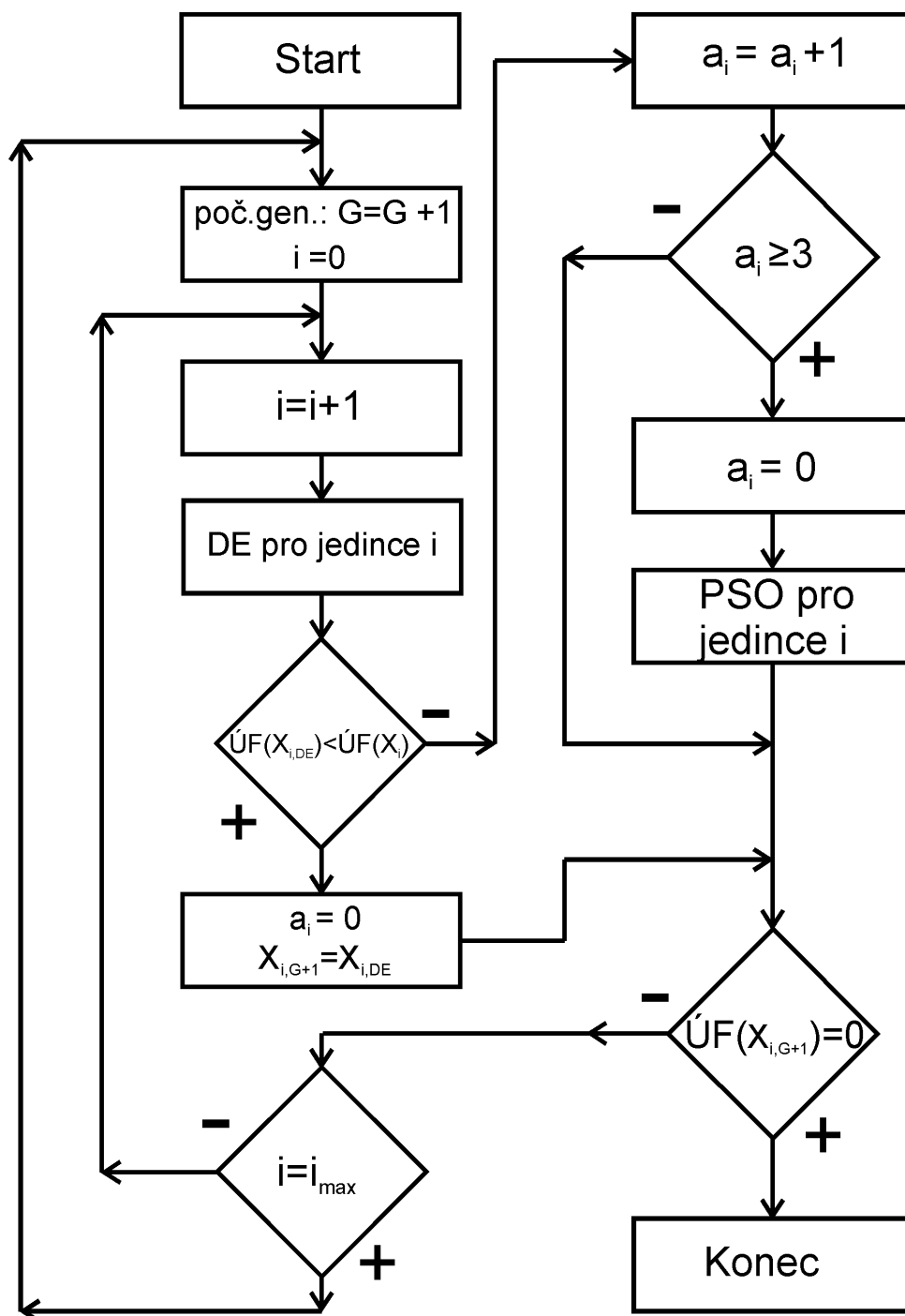
Algoritmus, který podobným způsobem spojuje obě metody, byl již uveden v teoretickém rozboru (v kapitole 2.2.9). U algoritmu [43] ovšem neúspěšný jedinec není nahrazen novým jedincem pomocí metody PSO vždy, ale pouze v případě, že má lepší hodnotu účelové funkce než aktuální neúspěšný jedinec. Algoritmus [43] tedy hledá výhodnější „pozici“ účelové funkce v blízkém okolí aktuálního neúspěšného jedince.

Vývojový diagram popsaného navrženého hybridního algoritmu DE-PSO1 je uveden na obr. 12, kde počet neúspěšných jedinců je uchováván pomocí jednorozměrného pole označeného písmenem **a** pro každého jedince zvlášť.

4.4.2 Algoritmus DE-PSO2

Algoritmu DE-PSO2 je modifikací algoritmu DE-PSO1. Algoritmus pracuje na stejném principu jako DE-PSO1 s rozdílem ve využití výsledků předchozích iterací. Na rozdíl od metody DE-PSO1, kde DE algoritmus kombinuje pouze aktuální pozice jedinců (nezáleží, jakou metodou byly získány), DE-PSO2 algoritmus využívá pro výpočet nové pozice pomocí metody DE pouze uložené nejlepší dosažené pozice. K tomu využívá zapamatovaných nejlepších pozic jednotlivých jedinců nutných pro běh algoritmu PSO spolu s hodnotou účelové funkce těchto nejlepších pozic.

Tento přístup minimalizuje situace, kdy je metoda PSO aktivována vícekrát za sebou a odtlačuje jedince příliš rychle od sebe a od možného řešení (jedinec je pro výpočet nové pozice pomocí DE metody fixován po několika iteraci).



Obr. 12 Vývojový diagram navrženého hybridního algoritmu DE-PSO1

4.4.3 Algoritmus PSO-DEm

Dominantní v případě tohoto navrženého hybridního algoritmu je metoda PSO. Podobně jako metoda DE-PSO2 využívá nejlepších „pozic“ jednotlivých agentů, které zpracovává algoritmus DE. Oproti PSO metodě jsou však v paměti uchovávány poslední tři nejlepší pozice (obecně \underline{s} posledních nejlepších pozic), které jsou náhodně vybírány pro použití v metodě DE.

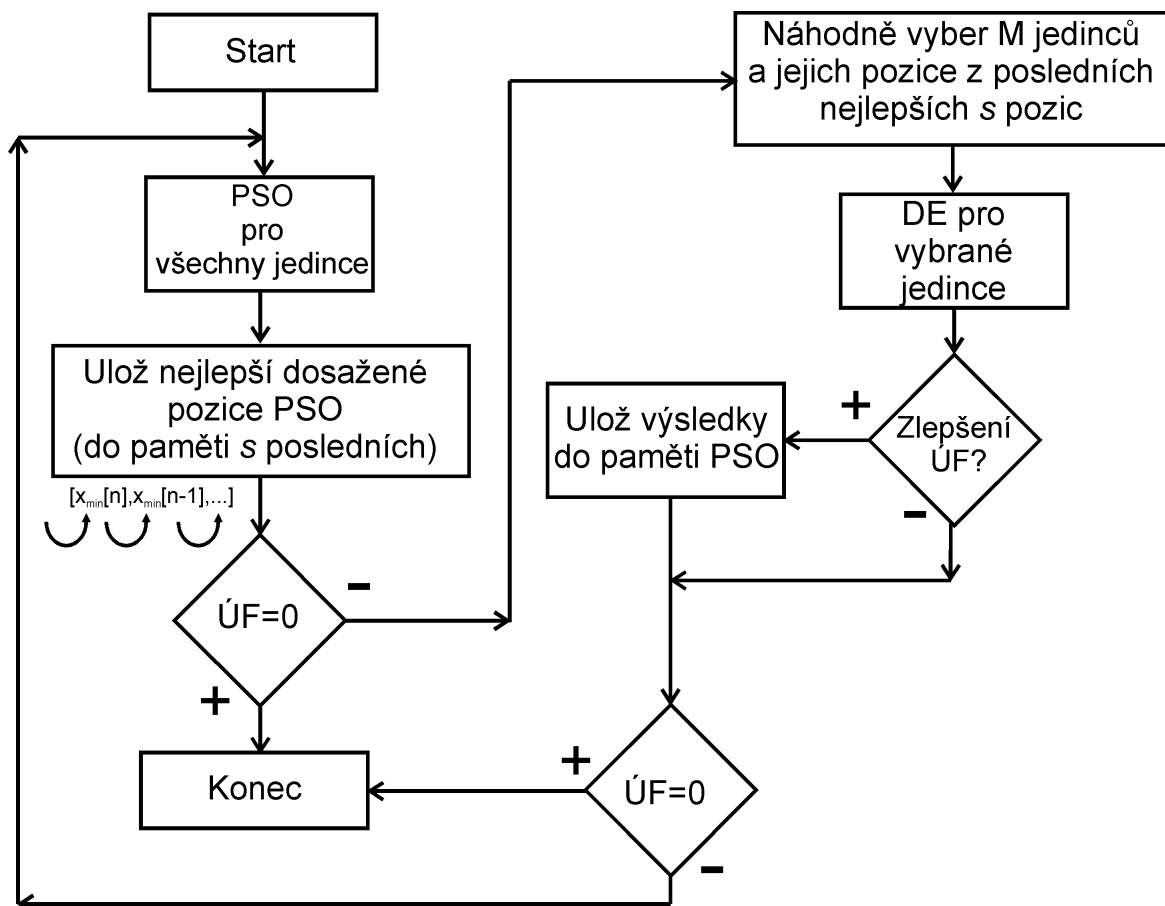
Celý algoritmus pracuje tak, že se nejprve provede celá iterace PSO metody a pro vybraný počet jedinců pak i metoda DE, která kombinuje výsledky jedinců z předchozích tří úspěšných iterací. Počet jedinců, kteří jsou případně nahrazeni dle DE postupu, je dobré volit přiměřený, aby se výrazně nezvýšil nutný počet vyhodnocení účelové funkce. Pro PSO-DEm metodu lze doporučit vazbu počtu vybraných jedinců na počet jedinců generace. Používanou variantou autora je v DE části algoritmu využít počet jedinců vypočtený jako počet jedinců generace NP modulo dimenze optimalizačního prostoru n (počet proměnných). Pro tyto vybrané jedince je provedena metoda *DE/current-to-best/1/bin* a výpočet šumového vektoru probíhá dle rovnice:

$$\vec{v} = \overrightarrow{X_{curr,G}} + \lambda(\overrightarrow{X_{Glob.best,G}} - \overrightarrow{X_{curr,G}}) + F(\overrightarrow{X_{r2,r2Opt[rand(1..3)]}} - \overrightarrow{X_{r3,r3Opt[rand(1..3)]}}), \quad (29)$$

kde $\overrightarrow{X_{r2,r2Opt[rand(1..3)]}}$ a $\overrightarrow{X_{r3,r3Opt[rand(1..3)]}}$ jsou pozice náhodně vybraných jedinců s indexem $r2$ a $r3$, které byly náhodně vybrány z jejich předchozích tří nejlepších pozic (což je v zápisu označeno zápisem $r..Opt[rand(1..3)]$), $\overrightarrow{X_{Glob.best,G}}$ je pozice dosud nalezeného nejlepšího výsledku a $\overrightarrow{X_{curr,G}}$ je pozice nejlepšího výsledku aktuálního jedince. Mutační konstanta F a konstanta zohledňující nejlepší výsledek λ mají stejný význam jako u algoritmu standardního algoritmu DE.

Pokud je pomocí DE nalezen bod s lepší účelovou funkcí než je nejlepší nalezený bod jedince, je tato pozice zapsána do paměti nejlepších výsledků jedince (případně i jako nejlepší nalezený výsledek).

Vývojový diagram PSO-DEm metody je uveden na obr. 13.



Obr. 13 Vývojový diagram navrženého hybridního algoritmu PSO-DEm

4.4.4 Algoritmus DESX

Algoritmus DESX [A2, A5] je hybridním spojením Nelder-Mead simplexové metody (NM) a metody DE. Jeho hlavní myšlenkou je použití Nelder-Mead transformací pro všechny jedince (nejen nejhoršího jedince) a nahrazení operace zmenšení simplexu metodou DE. Algoritmus lze tak snadno programově paralelizovat, neboť se transformační předpis provádí pro všechny jedince stejným způsobem a zúžení simplexu je nahrazeno přesunutím bodu dle DE metody.

Oproti Nelder-Mead algoritmu je však výhodnější použít větší množství jedinců než vyžaduje klasický NM algoritmus. Důvodem je fakt, že algoritmus DESX ze své podstaty postupně stahuje jedince k těžišti (centrálnímu bodu, či centroidu) a je tedy potřeba, aby byl optimalizační prostor dostatečně prohledán před případnou konvergencí k předpokládanému extrému účelové funkce (v okolí centroidu).

Dalším rozdílem oproti klasickému NM algoritmu je také výpočet centrálního bodu. NM algoritmus využívá centrálního bodu určeného dle vzorce (15). Pokud by algoritmus DESX používal stejné definice, pak by v situaci, kdy by všechny body byly

transformovány např. reflexí, zůstával centrální bod stejný. Proto je centrální bod v každé iteraci určován s ohledem na hodnotu účelových funkcí jednotlivých jedinců, obdobně jako v [25]. Pro minimalizaci hodnoty účelové funkce se pozice centrálního bodu vypočte:

$$\vec{C} = \frac{\sum_{i=1}^m (1 + OF_{max} - OF_i) \vec{X}_i}{\sum_{i=1}^m 1 + OF_{max} - OF_i}, \quad (30)$$

kde OF_{max} je maximální hodnota dané generace a OF_i je hodnota účelové funkce jedince \vec{X}_i .

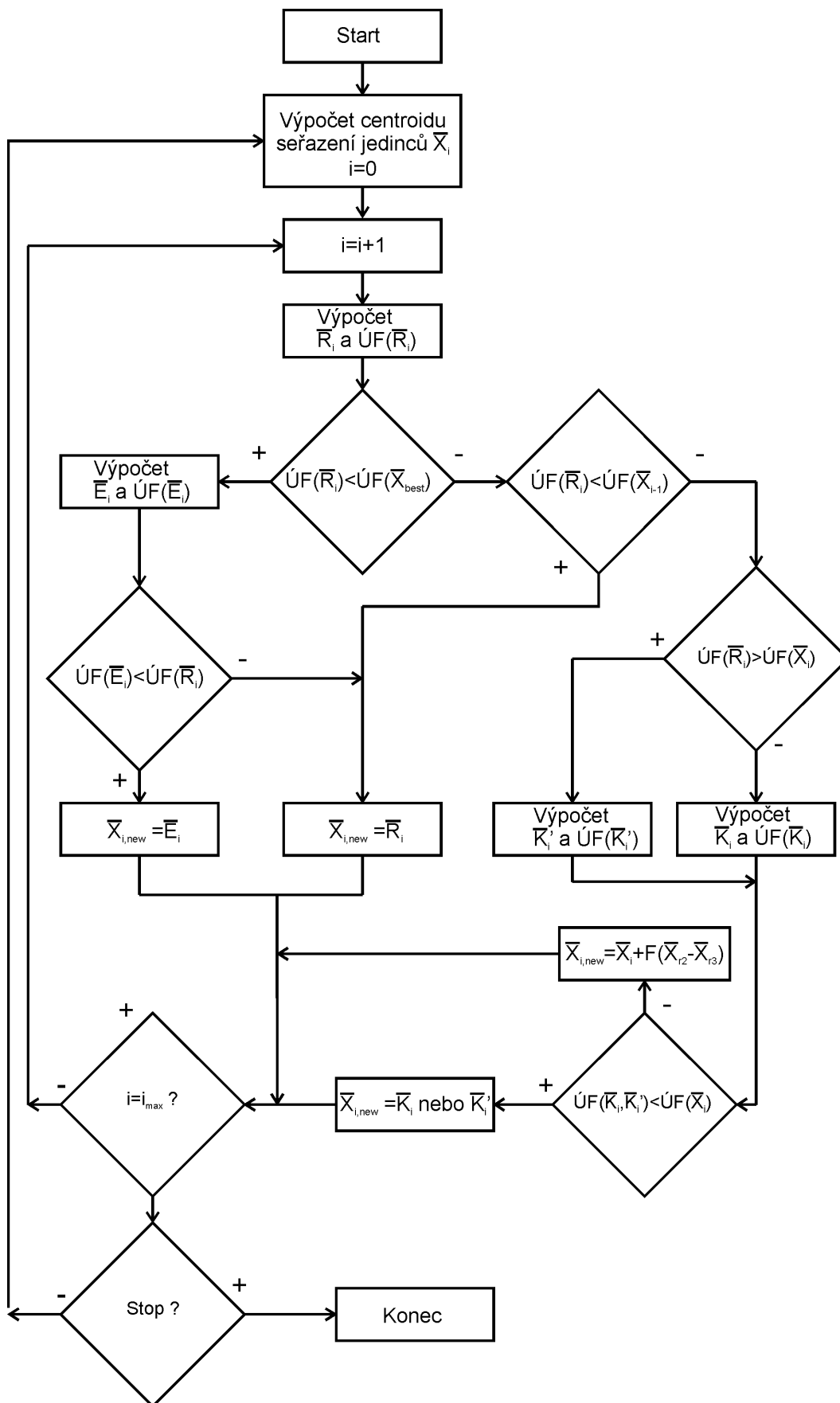
Jak již bylo uvedeno výše, DESX využívá operací reflexe, expanze a kontrakce pro všechny jedince, kteří mohou být před každou iterací seřazeni podle své hodnoty účelové funkce (může být pro běh algoritmu výhodnější). Oproti NM algoritmu pak v podmínkách figurují hodnoty účelové funkce aktuálního jedince (NM algoritmus transformuje vždy jedince s nehorší hodnotou účelové funkce). Pokud se transformuje jedinec s indexem i (\vec{X}_i), pak jeho hodnota účelové funkce je pro běh algoritmu brána jako hodnota nehoršího jedince. Za druhého nehoršího jedince je považován jedinec s nejbližší nižší hodnotou účelové funkce s označením \vec{X}_{i-1} .

Jedinec generace s nejlepší hodnotou účelové funkce je díky tomuto přístupu často transformován pomocí rovnice inspirované DE algoritmem. Jedinci, jejichž transformace by u NM algoritmu vedly k zužování simplexu, jsou přesouváni na nové pozice pomocí algoritmu *DE/current/1/bin*, kde šumový vektor je vypočten dle předpisu:

$$\vec{v}_{i,G+1} = \vec{x}_{i,G} + F(\vec{x}_{r_2,G} - \vec{x}_{r_3,G}). \quad (31)$$

V algoritmu DESX je ovšem tento jedinec přijat vždy do nové generace. Protože operace DE algoritmu může nahradit nejlepšího jedince populace méně výhodným jedincem, musí být nejlepší dosažený výsledek uchovávan separátně např. pomocí vektoru \vec{x}_{best} i s jeho hodnotou účelové funkce $UF(\vec{x}_{best})$.

Vývojový diagram metody DESX je uveden na obr. 14, kde i je index právě zpracovávaného jedince \vec{X}_i pomocí operací reflexe \vec{R}_i , expanze \vec{E}_i nebo kontrakce \vec{K}_i , obdobně jako u NM algoritmu (viz kapitola 2.2.8).



Obr. 14 Vývojový diagram DESX algoritmu

4.5 Hodnocení navržených hybridních algoritmů

Zhodnocení spolehlivosti a rychlosti konvergence jednotlivých algoritmů je ukázáno na optimalizaci banky filtrů, jejíž návrh je rozebrán v kapitole 4.3. Protože je banka filtrů realizována jednotlivými kaskádně řazenými SC filtry typu bikvadratické sekce, jsou optimalizační algoritmy porovnávány na optimalizacích jednotlivých bikvadratických sekcí.

Optimalizační úlohou pro porovnávané algoritmy je nastavení hodnot kapacitorů bikvadratických sekcí tak, aby výsledná amplitudová kmitočtová charakteristika jednotlivých bikvadratických sekcí ležela uvnitř tolerančního pásma. Toleranční pásmo je tvořeno pomocí ideální amplitudové kmitočtové charakteristiky, ke které je buď přičtena, nebo odečtena hodnota tolerance rozdílu od ideálního průběhu. Toleranční pásmo definují dvě mezní křivky amplitudových kmitočtových charakteristik $B_H(j2\pi f)$ a $B_L(j2\pi f)$, které vzniknou přičtením, resp. odečtením toleranční hodnoty $D = 10$ mdB od ideální amplitudové kmitočtové charakteristiky $H_{ideal}(j2\pi f)$.

4.5.1 Použitá účelová funkce

Pro běh optimalizačního algoritmu pomocí této podmínky je definována účelová funkce $\dot{U}F$. Pro výpočet její funkční hodnoty je třeba vyčíslit kmitočtovou charakteristiku jednotlivých bikvadratických sekcí v ekvidistantních bodech - kmitočtech. Účelová funkce $\dot{U}F$ je pak definována jako suma odchylek amplitudové kmitočtové charakteristiky od tolerančního pásma ve 250 bodech charakteristiky dle vztahu

$$\dot{U}F = \sum_{i=1}^{250} \begin{cases} H_{opt}(i \cdot j2\pi f_0) - B_H(i \cdot j2\pi f_0), & H_{opt}(i \cdot j2\pi f_0) > B_H(i \cdot j2\pi f_0) \\ B_L(i \cdot j2\pi f_0) - H_{opt}(i \cdot j2\pi f_0), & H_{opt}(i \cdot j2\pi f_0) < B_L(i \cdot j2\pi f_0) \\ 0, & jinak \end{cases} \quad (32)$$

kde $B_H(i \cdot j2\pi f_0)$ a $B_L(i \cdot j2\pi f_0)$ jsou hranice tolerančního pásma amplitudové kmitočtové charakteristiky pro i násobek kmitočtu f_0 , $|H_{ideal}(i \cdot j2\pi f_0)|$ je hodnota ideální amplitudové kmitočtové charakteristiky pro kmitočet $f = if_0$ a $|H_{opt}(i \cdot j2\pi f_0)|$ je hodnota amplitudové kmitočtové charakteristiky optimalizované bikvadratické sekce pro kmitočet $f = if_0$. Pokud ve všech násobcích $i = 1 \dots 250$ základního kmitočtu f_0 leží amplitudová kmitočtová charakteristika optimalizované bikvadratické sekce uvnitř tolerančního pásma, tj. mezi mezními hodnotami $B_L(i \cdot j2\pi f_0)$ a $B_H(i \cdot j2\pi f_0)$, je hodnota účelové funkce rovna nule a optimalizace končí.

4.5.2 Metodika hodnocení optimalizačních algoritmů

Cílem práce je navrhnout vhodný algoritmus pro optimalizaci reálných periodicky spínaných obvodů zahrnujícím nelinearity jednotlivých součástí. Pro porovnání optimalizace je mimo spolehlivosti algoritmu také zásadní počet vyčíslení hodnoty účelové funkce (resp. počet provedení analýzy obvodu), neboť je tento krok výpočetně nejnáročnější a především na něm závisí i čas potřebný k optimalizaci obvodu. Proto je nutné, aby optimalizační algoritmus rychle konvergoval k optimu a našel optimum při nejmenším počtu kroků. Toho musí být algoritmus schopen pro různé počáteční generace jedinců. Z toho vyplývá, že konvergenci algoritmů je nutno porovnávat na průměrných hodnotách počtu vyčíslení účelové funkce pro daný algoritmus a optimalizační úlohu.

Neméně důležitým parametrem je i spolehlivost algoritmu při hledání řešení, tj. je nutné určit, s jakou pravděpodobností algoritmus dokonverguje k optimu, resp. jakou má algoritmus tendenci ke stagnaci.

Z této úvahy vychází i kritéria hodnocení optimalizačních algoritmů při aplikaci na reálný nelineární periodicky spínaný obvod a to:

- průměrná hodnota počtu vyčíslení účelové funkce N [-],
- míra spolehlivosti optimalizačního algoritmu SR [%] (success rate),
- průběh konvergence optimalizačního algoritmu.

Jednotlivá kritéria jsou podrobněji popsána níže a jsou pak využita při porovnání navržených a výchozích optimalizačních metod a to pro každou bikvadratickou sekci zvlášť. Vyhodnocení kritérií je získáno z alespoň deseti optimalizačních běhů jednotlivých algoritmů pro všechny bikvadratické sekce.

Míra spolehlivosti optimalizačního algoritmu SR [%] (success rate)

Tento parametr vyjadřuje podíl úspěšně dokončených optimalizací ke všem spuštěným optimalizacím vyjádřený v procentech a je definován jako:

$$SR = \frac{\text{počet úspěšně dokončených optimal}}{\text{počet optimal. celkem}} \cdot 100 \text{ [%]}. \quad (33)$$

Průměrná hodnota počtu vyčíslení účelové funkce N [-]

Průměrná hodnota počtu vyčíslení účelové funkce udává průměrný počet vyčíslení účelové funkce (simulací) nutných k úspěšně dokončené optimalizaci.

Parametr N je v následujícím vyhodnocení vypočten pouze z úspěšně dokončených optimalizací, tj. optimalizací, které byly ukončeny s hodnotou účelové funkce rovnou 0. Pokud je míra spolehlivosti optimalizačního algoritmu nulová (SR=0%), pak číslo vyjadřuje průměrný počet simulací před ukončením optimalizace pomocí konvergenčního pravidla (ukončovací podmínky).

Průběh konvergence optimalizačního algoritmu

Pomocí tohoto kritéria lze hodnotit vývoj účelové funkce nejlepšího jedince UF_{best} v průběhu optimalizace a zároveň je možné jeho obdobu použít jako podmínku ukončení algoritmu, pokud optimalizace stagnuje.

Pro hodnocení průměrného vývoje účelové funkce UF_{best} v průběhu optimalizace je tento vývoj účelových funkcí nejprve normován na hodnotu účelové funkce nejlepšího jedince počáteční generace. Běh optimalizačního algoritmu se spustí několikrát s různou počáteční generací jedinců. Tím se získá několik normovaných křivek vývoje účelové funkce, které jsou pak průměrovány. Tím je obdržena průměrná normovaná účelová funkce $UF_{normovaná}$, která je vykreslena v závislosti na počtu vyhodnocení účelových funkcí (jednotlivé metody potřebují různý počet vyhodnocení účelové funkce za jednu iteraci).

4.5.3 Nastavení algoritmů a testu

Jednotlivé optimalizační algoritmy jak navržené hybridní (z kapitoly 4.4), tak jejich původní verze (uvedené v kapitole 2.2) byly testovány pro nejméně deset různých počátečních generací pro všechny bikvadratické sekce banky filtrů. V průběhu optimalizací byla kontinuálně vyhodnocována účelová funkce UF_{best} a ve chvíli, kdy se její hodnota nezměnila po 100 iteracích algoritmu, byl běh algoritmu zastaven (ukončovací podmínka algoritmu). Pro NM simplexový algoritmus byla tato ukončovací podmínka zmírněna na 400 iterací, neboť se jedná o jediný algoritmus, který nebyl paralelizován.

Algoritmy využívaly autorem experimentálně zjištěná nastavení řídicích konstant, kde pro navržené hybridní algoritmy byla použita stejná nastavení řídicích konstant jako pro originální algoritmy (viz tab. 4). Všechny řídicí parametry byly po celou dobu běhu algoritmu konstantní s výjimkou parametru λ , který je pro každého jedince vytvořeného pomocí DE v každé iteraci algoritmu volen náhodně z intervalu $\langle 0.1, 1.4 \rangle$.

Pro porovnání algoritmů byl vybrán hraniční případ počtu jedinců generace, který se rovnal pětinásobku počtu proměnných (počtu kapacitorů bikvad. sekce) $NP = 5n = 45$.

Tab. 4 Použitá nastavení řídicích konstant

| DE | | | PSO | | | NM Simplex | | | |
|-----|-----|----------------------------|-----|-------|-------|------------|---------|----------|----------|
| CR | F | λ | w | C_1 | C_2 | α | β | γ | δ |
| 0.9 | 0.5 | $\langle 0.1, 1.4 \rangle$ | 1 | 1.4 | 0.7 | 1 | 0.5 | 2 | 0.5 |

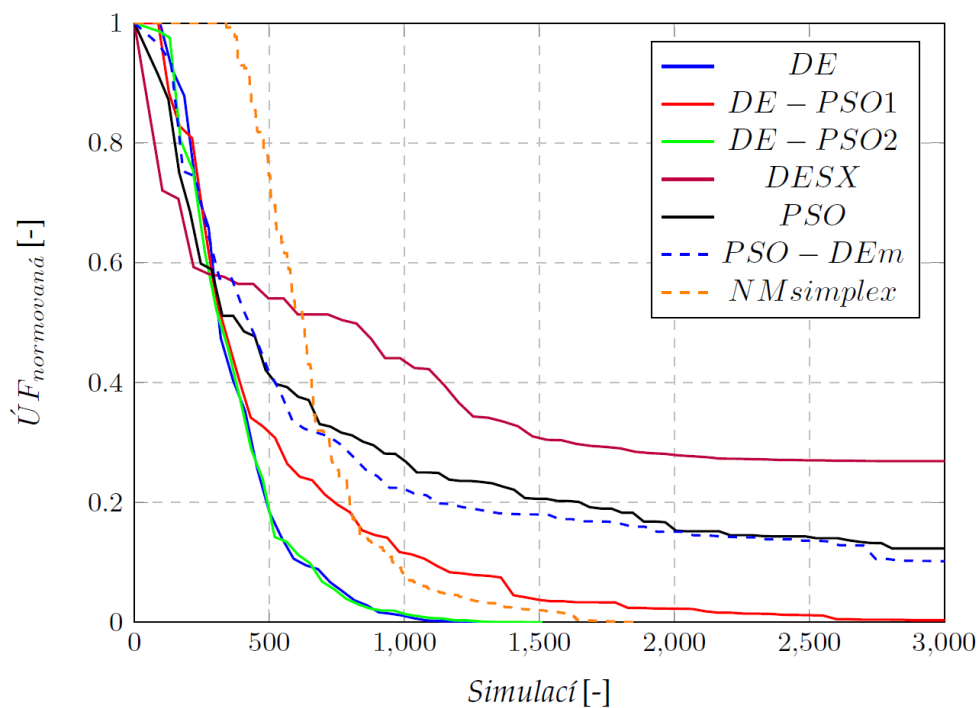
Protože je přenos navrhovaných filtrů závislý na poměrech kapacit, byla při optimalizaci zafixována velikost hodnoty kapacitoru C_a a to na hodnotu $C_a = 125fF$ pro všechny bikvadratické sekce. Povolený rozsah hodnot kapacitorů byl nastaven na $C_x \in \langle 30 fF, 500 fF \rangle$. Pokud algoritmus vygeneruje hodnotu mimo tento rozsah, je tato hodnota upravena do rozsahu přípustných hodnot dle předpisu (5).

4.5.4 Výsledky porovnání algoritmů

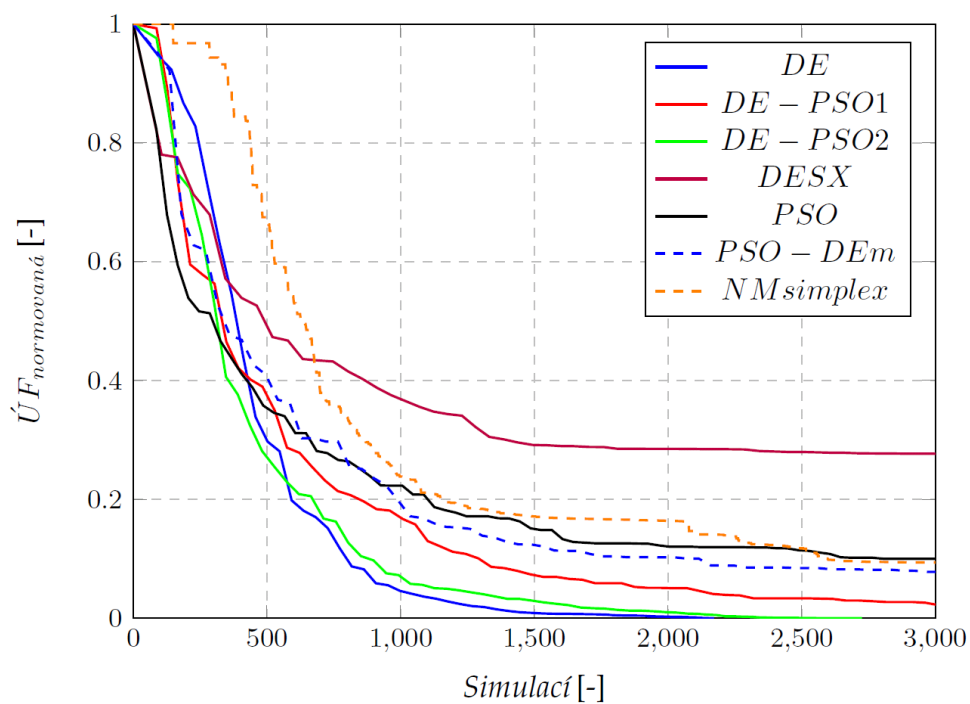
Navržené algoritmy byly testovány na optimalizaci všech typů bikvadratických sekcí dle prezentované metodiky a nastavení (viz výše) a průměrný počet nutných vyhodnocení účelových funkcí (simulací) spolu s vyhodnocením míry spolehlivosti nalezení požadovaného optima jsou shrnuty v tab. 5. Průběhy konvergence účelové funkce ($\dot{U}F_{normovaná}$) všech algoritmů pro všechny bikvadratické sekce jsou pak ukázány na obr. 15 až obr. 18.

Tab. 5 Průměrný počet vyčíslení účelové funkce N a success rate SR jednotlivých optimalizačních metod při ukončovací podmínce 100 iterací bez vývoje $\dot{U}F$

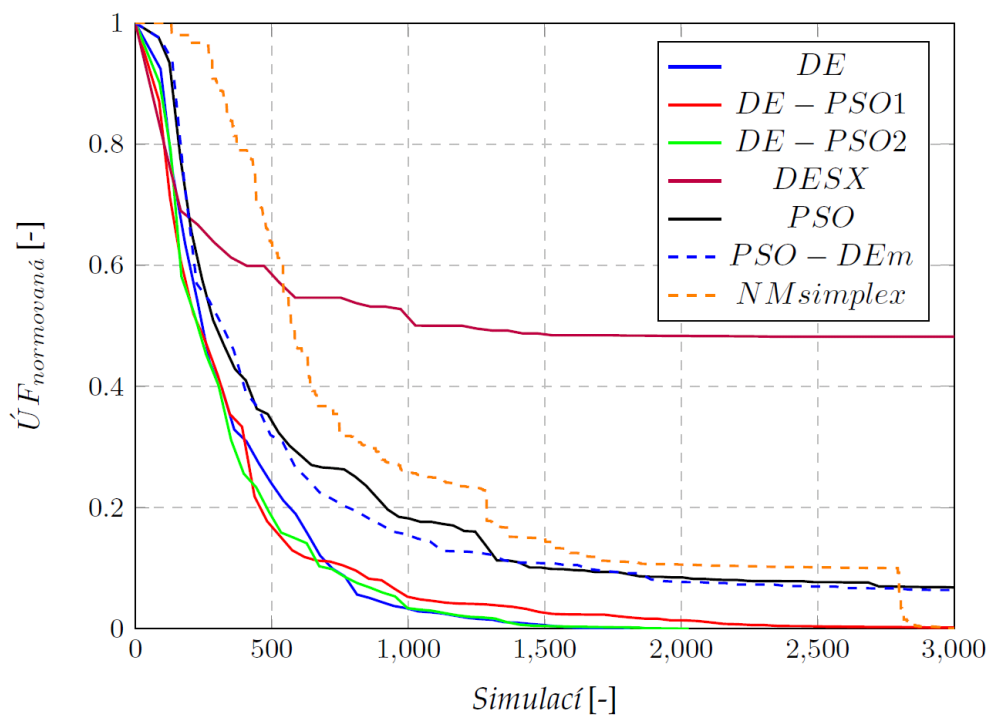
| | H_{LP1} | | H_{LP2} | | H_{HP1} | | H_{HP2} | |
|---------------------------------|-------------|------------|-------------|------------|-------------|------------|-------------|------------|
| | N | SR | N | SR | N | SR | N | SR |
| | [-] | [%] | [-] | [%] | [-] | [%] | [-] | [%] |
| DE/current-to-best/1/bin | 1333 | 100 | 2172 | 67 | 1811 | 67 | 1292 | 100 |
| DE-PSO1 | 6146 | 100 | 10299 | 100 | 5939 | 100 | 3976 | 100 |
| DE-PSO2 | 1512 | 100 | 2725 | 100 | 2027 | 100 | 1417 | 100 |
| DESX | 6509 | 20 | 7770 | 0 | 5810 | 0 | 5872 | 50 |
| PSO | 38058 | 10 | 84942 | 0 | 47677 | 0 | 32127 | 0 |
| PSO-DEm | 48754 | 0 | 77855 | 0 | 55755 | 0 | 37140 | 0 |
| NM Simplex | 3143 | 100 | 5371 | 60 | 3495 | 80 | 2370 | 100 |



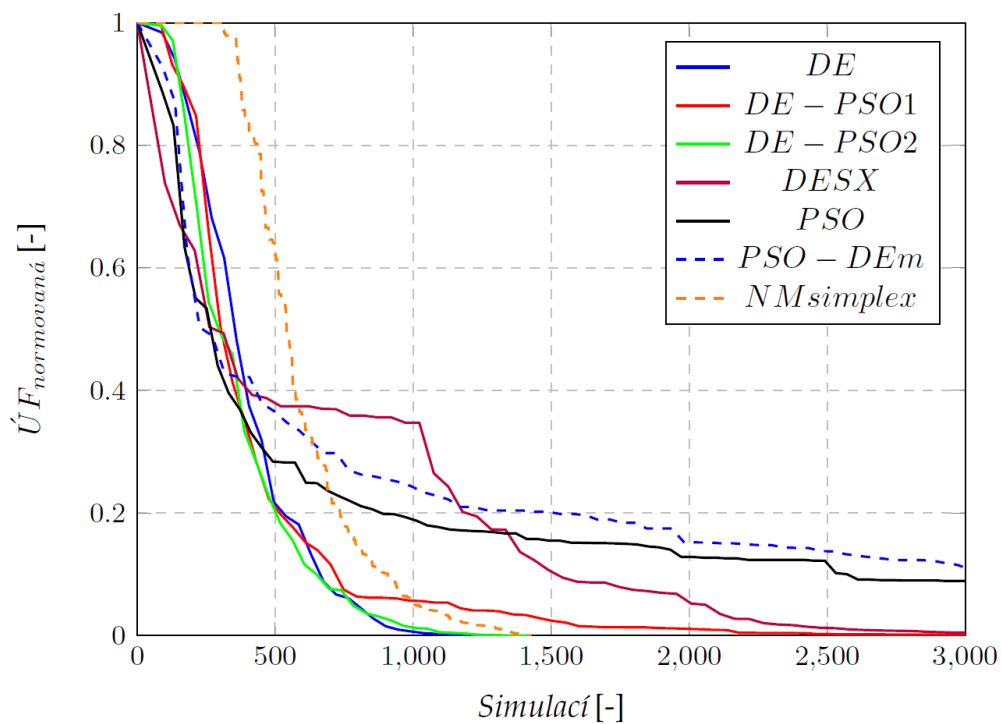
Obr. 15. Průběh normované účelové funkce v závislosti na počtu simulací pro bikvadratickou sekci H_{LP1}



Obr. 16. Průběh normované účelové funkce v závislosti na počtu simulací pro bikvadratickou sekci H_{LP2}



Obr. 17. Průběh normované účelové funkce v závislosti na počtu simulací pro bikvadratickou sekci H_{HP1}



Obr. 18. Průběh normované účelové funkce v závislosti na počtu simulací pro bikvadratickou sekci H_{HP2}

Metoda klasické diferenciální evoluce je v porovnání s ostatními algoritmy nejrychlejší metodou při optimalizaci navrženého periodicky spínaného obvodu (viz. tab. 5). Její spolehlivost však není pro všechny úlohy stoprocentní. Pro bikvadratické sekce H_{LP2} a H_{HP1} se projevila tendence algoritmu *DE/current-to-best/1/bin* ke stagnaci a předčasné konvergenci algoritmu. Tato varianta algoritmu DE ovšem při experimentech nejrychleji konvergovala k výsledku.

Nejspolehlivějšími metodami testu se ukázaly navržené hybridní algoritmy DE-PSO1 a DE-PSO2, které byly schopny nalézt minimum účelové funkce s nejvyšší spolehlivostí *SR* pro všechny optimalizované bikvadratické sekce. Porovnáme-li průměrný počet simulací (N) algoritmů DE-PSO s algoritmem DE, pak algoritmus DE-PSO2 v nejhorším případě (H_{LP2}) potřeboval pouze o 25 % více vyhodnocení účelové funkce (simulací) než algoritmus DE.

Dobrou konvergenci k řešení ukázal také Nelder-Mead simplexový algoritmus, jenž používal k výpočtu centroidu vztah (30) (obdobný jako v [25]). Průměrně byl však NM algoritmus schopen optimalizovat navržené spínané filtry zhruba při dvojnásobném počtu simulací oproti algoritmu DE. NM algoritmus však nebyl na rozdíl od ostatních algoritmů paralelizován a tak výsledný čas optimalizace je více než dvojnásobný.

Při testu se algoritmy založené na rojení částic (PSO a PSO-DEm) ukázaly jako málo efektivní optimalizační metody pro periodicky spínané obvody. Jejich spolehlivost je při použití ukončovací podmínky, která hlídá rychlost konvergence algoritmu, velmi malá a tyto optimalizace jsou obvykle ukončeny s neúspěchem při řádově větším počtu simulací.

Spolehlivost „PSO algoritmů“ je však zkreslena ukončovací podmínkou, kdy rychlost konvergence těchto metod s přiblížením k optimu klesá. Pokud podmínku odstraníme, jejich spolehlivost se zvýší (především pro H_{LP1} a H_{HP2}) ale na úkor potřebného času na optimalizaci, který se může rapidně zvýšit (viz tab. 6). Hodnoty počtu vyčíslení účelových funkcí N v tab. 6 jsou pro algoritmy, kdy $SR < 20\%$, hodnotami, kdy byla optimalizace manuálně ukončena.

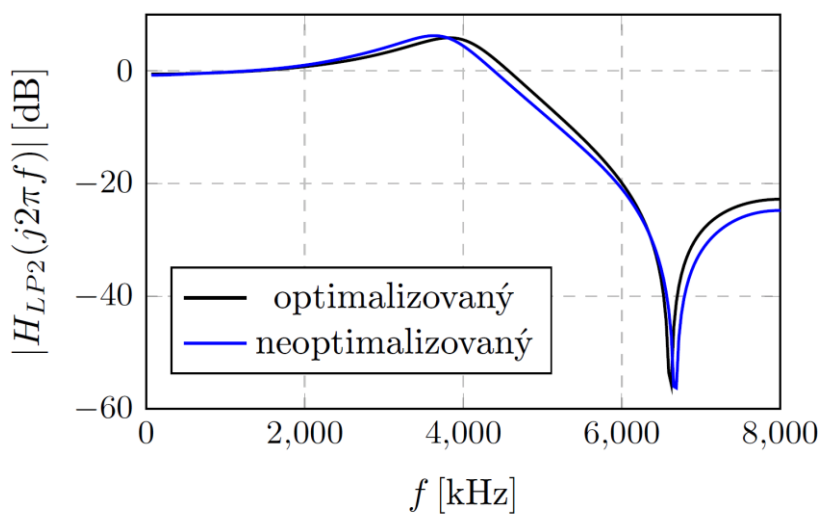
Spolehlivost algoritmu DESX se také ukázala jako velmi malá. U tohoto algoritmu rozmanitost jedinců s přibývajícemi iteracemi klesá a metoda je citlivá na počet jedinců v generaci NP . Průměrně byla metoda ukončena po 140 iteracích algoritmu, kdy se účelová funkce nejlepšího jedince nelišila o více než 0.1 % od účelové funkce nejhoršího jedince.

Tab. 6 Průměrný počet vyčíslení účelové funkce N a success rate SR jednotlivých optimalizačních metod bez ukončovací podmínky 100 iterací bez vývoje ÚF

| | H_{LP1} | | H_{LP2} | | H_{HP1} | | H_{HP2} | |
|-------------------|-----------|------------|-----------|------|-----------|------------|-----------|------------|
| | N | SR | N | SR | N | SR | N | SR |
| | [-] | [%] | [-] | [%] | [-] | [%] | [-] | [%] |
| PSO | 43030 | 80 | 91614 | 20 | 56229 | 100 | 43375 | 100 |
| PSO-DEm | 58411 | 100 | 93500 | 10 | 71059 | 80 | 44178 | 100 |
| NM Simplex | 3143 | 100 | 9812 | 70 | 5006 | 100 | 2370 | 100 |

Rychle klesající konvergenci DESX algoritmu, kdy se vývoj účelové funkce velmi rychle zpomaluje, lze vidět i na průbězích účelové funkce na obr. 15 až obr. 17. Pro optimalizaci bikvadratické sekce H_{HP2} , lze však pozorovat na obr. 18 zrychlení konvergence po prvních přibližně tisících simulacích (a také spolehlivost je pro tento případ vyšší). Algoritmu DESX tak není vhodný pro optimalizace s malým počtem jedinců generace.

Optimalizované hodnoty kapacitorů pro všechny typy bikvadratických sekcí jsou spolu s procentuálními odchylkami od idealizovaných hodnot (viz. norm. hodnoty v tab. 2) uvedeny v tab. 7. Amplitudová kmitočtová charakteristika jak neoptimalizované, tak optimalizované bikvadratické sekce H_{LP2} je vynesena na obr. 19. Vzhledem k parametrům optimalizace, kdy byla bikvadratická sekce optimalizována na maximální rozdíl 10 mdB od ideální charakteristiky, se optimalizovaná verze kryje s ideální kmitočtovou charakteristikou.



Obr. 19 Amplitudová kmitočtová charakteristika optimalizované a neoptimalizované bikvadratické sekce H_{LP2}

Tab. 7 Hodnoty kapacitorů po optimalizaci a procentuální odchylky hodnot kapacitorů
oproti neoptimalizovaným filtru v tab. 2

| | H_{LP1} | | H_{LP2} | | H_{HP1} | | H_{HP2} | |
|-------|-----------|----------------|-----------|----------------|-----------|----------------|-----------|----------------|
| | C [fF] | ΔC [%] | C [fF] | ΔC [%] | C [fF] | ΔC [%] | C [fF] | ΔC [%] |
| C_a | 125,00 | - | 125,00 | - | 125,00 | - | 125,00 | - |
| C_b | 58,60 | 0,68 | 62,79 | 0,02 | 61,92 | 1,41 | 58,07 | 1,74 |
| C_c | 87,03 | -1,20 | 130,80 | -0,93 | 132,84 | -2,50 | 87,78 | -1,72 |
| C_d | 99,77 | 0,23 | 99,11 | 0,89 | 101,95 | -1,95 | 100,93 | -0,93 |
| C_e | 40,33 | -0,82 | 0 | - | 0 | - | 40,44 | -1,09 |
| C_f | 40,76 | 0,59 | 37,40 | -0,82 | 36,94 | 0,43 | 40,99 | -0,46 |
| C_g | 142,99 | -0,70 | 170,08 | 0,65 | 60,42 | -1,21 | 89,60 | -0,45 |
| C_h | 50,03 | -0,06 | 48,07 | 3,86 | 51,66 | -3,32 | 49,78 | 0,44 |
| C_i | 40,91 | 0,21 | 41,65 | -2,09 | 39,20 | 5,54 | 41,48 | 0,05 |
| C_j | 103,17 | -0,17 | 101,51 | 1,73 | 104,32 | -0,31 | 102,69 | 1,26 |

5 Závěr

Předložená disertační práce se věnuje aplikaci optimalizačních algoritmů na reálné spínané obvody a navazuje na práci [5]. Práce se soustřeďuje na optimalizaci reálného nelineárního spínaného obvodu, kdy čas numerické analýzy obvodu není nezanedbatelný. Hlavním cílem je určení a navržení optimalizační metody na bázi evolučních algoritmů, která i pro nízký počet jedinců generace konverguje k řešení spolehlivě a přijatelně rychle.

Práce poté navrhuje tři varianty hybridních algoritmů a zdůrazňuje důvod, proč není účelné použití adaptivních algoritmů (viz kap. 4.4). Práce také navrhuje metodiku testování algoritmů tak, aby se zohlednila i doba běhu algoritmu (resp. zohledňuje nezanedbatelný simulační čas).

Pro testování jednotlivých algoritmů je navržena realizace dvoukanálové banky filtrů 4. řádu, které byly implementovány v AMS 0.35 μm technologii včetně návrhu vlastního operačního zesilovače.

Testování jednotlivých algoritmů probíhá na navržených SC bikvadratických sekcích, kde sekce obsahují i vlastní navržený OZ a je cílem hodnotit jednotlivé algoritmy na pokud možno co nejvíce reálné struktuře.

Navržený hybridní algoritmus na bázi DE a PSO nazvaný DE-PSO2 pak v porovnání s ostatními algoritmy překonává z hlediska spolehlivosti výsledky ostatních algoritmů při zvýšení počtu simulací pouze o 25% oproti méně spolehlivému algoritmu DE.

Detailní přehled všech přínosů práce je uveden níže.

5.1 Přínosy práce

Přínosy práce lze rozdělit do dvou kategorií a to na teoretické a praktické přínosy.

Teoretické přínosy

Hlavním teoretickým přínosem jsou tři navržené hybridní evoluční algoritmy založené na algoritmech DE a PSO (nazvané zkratkami DE-PSO1, DE-PSO2, PSO-DEm), kde především algoritmy DE-PSO1 a DE-PSO2 zlepšují spolehlivost optimalizace spínaných obvodů při současném použití hraniční velikosti počtu jedinců generace.

Dalším teoretickým přínosem souvisejícím s předchozím bodem je stanovení kritérií, podle kterých by měly být hodnoceny optimalizační algoritmy při optimalizaci reálných nelineárních periodicky spínaných obvodů. Autor především zdůrazňuje nutné kritérium spolehlivosti nalezení optima a porovnávání algoritmů pomocí nutného počtu simulací.

Vedlejší teoretické přínosy

Vedlejší přínosem je postup rozdělení optimalizace banky filtrů na části tak, aby se zrychlil proces optimalizace.

Teoretickým přínosem je i navrhovaná implementace banky filtrů pomocí techniky spínaných kapacitorů, která zpracovává diskrétní analogové signály až do poloviny vzorkovacího kmitočtu.

Dalším vedlejším přínosem je modifikovaná (hybridní) verze Nelder-Mead algoritmu nazvaná DESX, kdy je postup Nelder-Mead algoritmu paralelizován pro všechny jedince generace, kde je poté prakticky ukázána rychle se zpomalující konvergence v prvních iteracích tohoto algoritmu.

Praktické přínosy

Hlavní praktický přínos je ve využití navržených hybridních algoritmů při optimalizaci jednotlivých sekcí dvoukanálové banky filtrů technikou spínaných kapacitorů. Výsledkem je optimalizovaný návrh jednotlivých bloků banky filtrů, který je také využit ke zhodnocení navržených hybridních algoritmů.

Dalším praktickým přínosem je implementace banky filtrů pomocí techniky spínaných kapacitorů.

Dále je to "statistické"vyhodnocení, které přineslo cenné výsledky pro zhodnocení jednotlivých metod a bylo z celé práce časově nejnáročnější.

Vedlejší praktické přínosy

Vedlejším praktickým přínosem je navržený operační zesilovač v technologii AMS 0.35 μm a jeho využití v navržených bikvadratických sekcích.

Dalším vedlejším přínosem je implementace banky filtrů třetího řádu, která byla realizována na plošném spoji, a která byla využita při identifikaci problémů při praktické realizaci SC obvodů pracujících až do poloviny vzorkovacího kmitočtu.

Celkový přínos

Celkovým přínosem práce je podrobné prozkoumání problematiky optimalizace reálných periodicky spínaných obvodů i s ohledem na nelinearity obvodu, a to jak z hlediska spolehlivosti, tak i rychlosti konvergence. Z analýzy problematiky pak navrhuje spojení vhodných optimalizačních algoritmů tak, aby se některé z jejich vlastností zlepšily.

5.2 Další možný směr výzkumu

Další směr výzkumu v problematice optimalizace periodicky spínaných obvodů vidí autor ve studiu závislostí nastavení řídicích konstant algoritmů a velikosti generace pro jednotlivé optimalizační metody.

Kromě obecných postupů optimalizace by bylo možné zahrnout do hodnocení i rychlost jednotlivých optimalizačních algoritmů, pokud se připočítají i další parametry návrhu jako je například snížení rozptylu parametrů obvodu přes všechny technologické krajní případy (tzv. technologické rohy) a uvažování vlivu nepřesností časovacích signálů či rozptylu hodnot jednotlivých součástí.

Dalším námětem na případné pokračování práce je rozšíření studie použitelnosti a efektivnosti jednotlivých algoritmů pro více typů analogových periodicky spínaných soustav (například i nábojových pump...).

6 Literatura

- [1] RADEV, Nikolay, Kantcho IVANOV, Kalin STANCHEV, Simona PETRAKIEVA, Nikos MASTORAKIS, Valeri MLADENOV a Kliment Ohridski ST. Left-LUD SC ladder filter with compensation for finite amplifier gain and offset voltage. In: *11th WSEAS International Conference on CIRCUITS*. 2007, s. 157–161.
- [2] SERRA, Hugo, Rui SANTOS-TAVARES a Nuno PAULINO. A top-down optimization methodology for SC filter circuit design. In: *2014 IEEE International Symposium on Circuits and Systems (ISCAS)* [online]. B.m.: IEEE, 2014, s. 1672–1675. ISBN 978-1-4799-3432-4. Dostupné z: doi:10.1109/ISCAS.2014.6865474
- [3] DOLÍVKA, Lukáš a Jiří HOSPODKA. Ways to optimize analogue switched circuits. *Radioengineering*. 2008, **17**(4), 48–54. ISSN 12102512.
- [4] DOLÍVKA, Lukáš a Jiří HOSPODKA. Switched-Capacitor Filter Optimization with Respect to Switch On-State Resistance and Features of Real Operational Amplifiers. *Radioengineering*. 2007, **16**(2), 34–39.
- [5] DOLÍVKA, Lukáš. *Návrh a optimalizace diskrétně pracujících obvodů s využitím evolučních algoritmů*. B.m., 2012. ČVUT v Praze, FEE.
- [6] MICHAEL, Sherif. Modeling and optimization techniques of electronic devices using genetic algorithm. *ITI 2008 - 30th International Conference on Information Technology Interfaces* [online]. 2008, 751–756. Dostupné z: doi:10.1109/ITI.2008.4588505
- [7] ZHANG, Jun, Yuan SHI a Zhi-Hui ZHAN. Power electronic circuits design: A particle swarm optimization approach. *Simulated Evolution and Learning* [online]. 2008, (60573066), 605–614. ISSN 03029743. Dostupné z: doi:10.1007/978-3-540-89694-4_61
- [8] BARARI, Mansour, Hamid Reza KARIMI a Farhad RAZAGHIAN. Analog Circuit Design Optimization Based on Evolutionary Algorithms. *Mathematical Problems in Engineering* [online]. 2014, **2014**(1), 1–12. ISSN 1024-123X. Dostupné z: doi:10.1155/2014/593684
- [9] O. J. USHIE AND M. ABBOD. " Intelligent Optimization Methods for Analogue Electronic Circuits: GA and PSO Case Study," . *International Conference on Machine Learning, Electrical and Mechanical Engineering (ICMLEME'2014)*.

- 2014, 193–199.
- [10] KUBAŘ, Miloslav a Jiří JAKOVENKO. A powerful optimization tool for analog integrated circuits design. *Radioengineering*. 2013, **22**(3), 921–931. ISSN 12102512.
- [11] GAJDA, Zbysek a Lukas SEKANINA. Gate-level optimization of polymorphic circuits using cartesian genetic programming. *2009 IEEE Congress on Evolutionary Computation, CEC 2009* [online]. 2009, 1599–1604. Dostupné z: doi:10.1109/CEC.2009.4983133
- [12] KARAKATIC, S., V. PODGORELEC a M. HERICKO. Optimization of combinational logic circuits with genetic programming. *Elektronika ir Elektrotechnika* [online]. 2013, **19**(7), 86–89. ISSN 13921215. Dostupné z: doi:10.5755/j01.eee.19.7.5169
- [13] DOLÍVKA, Lukáš. Using Evolutionary Algorithms for Optimization of Analogue Electronic Filters. In: Eisuke KITA, ed. *Evolutionary Algorithms*. Rijeka: InTech, 2011, s. 381–406. ISBN 978-953-307-171.
- [14] STORN, R. Differential evolution design of an IIR-filter. *IEEE International Conference on Evolutionary Computation* [online]. 1996, 268–273. Dostupné z: doi:doi: 10.1109/ICEC.1996.542373
- [15] ŽIŠKA, Přemysl. *Využití evolučních algoritmů v syntéze lineárních časově invariantních (LTI) obvodů*. B.m., 2006. ČVUT v Praze, FEE.
- [16] BIČÁK, Jan a Jiří HOSPODKA. Frequency Response of Switched Circuits in SPICE. In: *ECCTD' 03*. 2003, s. 333–336.
- [17] KURTH, Carl F. a George F. MOSCHYTZ. Nodal Analysis of Switched-Capacitor Networks. *IEEE Transactions on Circuits and Systems* [online]. 1979, **26**(2), 93–105. Dostupné z: doi:10.1109/TCS.1979.1084613
- [18] BIČÁK, Jan, Jiří HOSPODKA a Pravoslav MARTINEK. Analysis of SI Circuits in MAPLE Program. In: *ECCTD' 01*. 2001, s. 121–124.
- [19] MOHAN, P. V. Ananda, V. RAMACHANDRAN a M. N. S. SWAMY. *Switched Capacitor Filters: Theory, Analysis, and Design*. B.m.: Prentice Hall, 1995. ISBN 0-13-879818-4.
- [20] OPAL, Ajoy a Jiri VLACH. Analysis and Sensitivity of Periodically Switched Linear Networks. *IEEE Transactions on Circuits and Systems* [online]. 1989, **36**(4), 522–532. ISSN 00984094. Dostupné z: doi:10.1109/31.92884
- [21] BIOLEK, Dalibor. Modeling of Periodically Switched Networks by Mixed -

- Description. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* [online]. 1997, **44**(8), 750–758. Dostupné z: doi:10.1109/81.611272
- [22] BIČÁK, Jan a Jiří HOSPODKA. Evaluation of Various Approaches to Frequency Response Calculation of Switched Circuits in Spice. In: *SMO'05 Proceedings of the 5th WSEAS international conference on Simulation, modelling and optimization*. 2005, s. 641–645.
- [23] ZAPLATÍLEK, Karel, Přemysl ŽIŠKA a Karel HÁJEK. Practice utilization of algorithms for analog filter group delay optimization. *Radioengineering*. 2007, **16**(1), 7–15. ISSN 1805-9600.
- [24] PRAJAPATI, Pankaj P. a Mihir V. SHAH. Two stage CMOS operational amplifier design using particle swarm optimization algorithm. In: *2015 IEEE UP Section Conference on Electrical Computer and Electronics (UPCON)* [online]. B.m.: IEEE, 2015, s. 1–5. ISBN 978-1-4673-8507-7. Dostupné z: doi:10.1109/UPCON.2015.7456700
- [25] MALLARD, Thomas. An improved Nelder-Mead method for analog design optimisation applied to deep sub-micron technology. In: *25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communities Technologies (ISSC 2014/CICT 2014)* [online]. B.m.: Institution of Engineering and Technology, 2014, s. 164–168. ISBN 978-1-84919-924-7. Dostupné z: doi:10.1049/cp.2014.0678
- [26] ZELINKA, Ivan, Zuzana OPLATKOVÁ, Miloš ŠEDA, Pavel OŠMERA a František VČELARĚ. *Evoluční výpočetní techniky - principy a aplikace*. Praha: BEN - technická literatura, 2009. ISBN 978-80-7300-218-3.
- [27] WEISE, Thomas. *Global Optimization Algorithms—Theory and Application* [online]. 2009. Dostupné z: doi:doi=10.1.1.64.8184
- [28] HOLLAND, John Henry. *Adaptation in natural and artificial systems*. B.m.: University of Michigan Press, 1975. ISBN 0-472-08460-7.
- [29] MAŘÍK, Vladimír, Olga ŠTĚPÁNKOVÁ a Jiří LUŽANSKÝ. *Umělá inteligence (3)*. Praha: Academia, 2001. ISBN 9789004310087.
- [30] DORIGO, Marco. *Optimization, Learning and Natural Algorithms*. Milan, Italy, 1992. Dipartimento di Elettronica, Politecnico di Milano.
- [31] DORIGO, Marco a L.M. GAMBARDELLA. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on*

- Evolutionary Computation* [online]. 1997, **1**(1), 53–66. ISSN 1089778X. Dostupné z: doi:10.1109/4235.585892
- [32] EBERHART, Russell C a James KENNEDY. A new optimizer using particle swarm theory. In: *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science* [online]. 1995, s. 39–43. ISBN 0-7803-2676-8. Dostupné z: doi:10.1109/MHS.1995.494215
- [33] MAŘÍK, Vladimír, Olga ŠTĚPÁNKOVÁ a Jiří LUŽANSKÝ. *Umělá inteligence (4)*. Praha: Academia, 2003. ISBN 80-200-1044-0.
- [34] CORNE, David, Marco DORIGO a Fred GLOVER, ed. *New Ideas in Optimization*. B.m.: McGraw-Hill, 1999. ISBN 0-07-709506-5.
- [35] JAMIL, Momin a Xin-She YANG. A Literature Survey of Benchmark Functions For Global Optimization Problems. *Int. Journal of Mathematical Modelling and Numerical Optimisation* [online]. 2013, **4**(2), 150–194. ISSN 2040-3607. Dostupné z: doi:10.1504/IJMMNO.2013.055204
- [36] STORN, R a K PRICE. *Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces*. B.m.: ICSI. 1995. TR: International Computer Science Institute
- [37] STORN, R. a K. PRICE. Differential Evolution. A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* [online]. 1997, **11**, 341–359. Dostupné z: doi:10.1023/A:1008202821328
- [38] SHI, Y a Russell C EBERHART. Empirical study of particle swarm optimization. *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on* [online]. 1999, **3**, 1–1950 Vol. 3. ISSN 1089-778X. Dostupné z: doi:10.1109/CEC.1999.785511
- [39] GOLBERG, David E. *Genetic Algorithms in Search Optimization & Machine Learning* [online]. 1989. ISBN 0-201-15767-5. Dostupné z: doi:10.1007/3-540-44673-7
- [40] MICHALEWICZ, Zbigniew. *Genetic Algorithms + Data Structures = Evolution Programs* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996. ISBN 978-3-642-08233-7. Dostupné z: doi:10.1007/978-3-662-03315-9
- [41] PRICE, Kenneth. Genetic Annealing. *Dr. Dobb's Journal*. 1994, **220**, 127–132.
- [42] STORN, Rainer. On the usage of differential evolution for function optimization. In: *Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American* [online]. B.m.: IEEE, 1996, s. 519–523.

- ISBN 0-7803-3225-3. Dostupné z: doi:10.1109/NAFIPS.1996.534789
- [43] NWANKWOR, E., A. K. NAGAR a D. C. REID. Hybrid differential evolution and particle swarm optimization for optimal well placement. *Computational Geosciences* [online]. 2013, **17**(2), 249–268. ISSN 1420-0597. Dostupné z: doi:10.1007/s10596-012-9328-9
- [44] BEIRAMI, Hamed, Ali Zargar SHABESTARI a Mohammad Mahdi ZERAFAT. Optimal PID plus fuzzy controller design for a PEM fuel cell air feed system using the self-adaptive differential evolution algorithm. *International Journal of Hydrogen Energy* [online]. 2015, **40**(30), 9422–9434. ISSN 03603199. Dostupné z: doi:10.1016/j.ijhydene.2015.05.114
- [45] STORN, Rainer. Differential Evolution - Trends and Open Questions. In: *Advances in Differential Evolution* [online]. 2008, s. 1–31. ISBN 978-3-540-68827-3. Dostupné z: doi:10.1007/978-3-540-68830-3_1
- [46] WANG, Xiang a Guoyi XU. Hybrid differential evolution algorithm for traveling salesman problem. *Procedia Engineering* [online]. 2011, **15**, 2716–2720. ISSN 18777058. Dostupné z: doi:10.1016/j.proeng.2011.08.511
- [47] GÄMPERLE, R., S.D. MÜLLER a P. KOUMOUTSAKOS. A parameter study for differential evolution. In: *WSEAS Int. Conf. on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*. 2002, s. 293–298.
- [48] DAS, Swagatam a Ponnuthurai Nagaratnam SUGANTHAN. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation* [online]. 2011, **15**(1), 4–31. ISSN 1089778X. Dostupné z: doi:10.1109/TEVC.2010.2059031
- [49] ZAHARIE, Daniela. Critical values for the control parameters of differential evolution algorithms. In: *Proceedings of Mendel 2002, 8th International Conference on Soft Computing, R. Matoušek and P. Ošmera, Eds., Brno, Czech Republic, 7. 2002*, s. 62–67.
- [50] PRICE, K. V., R. M. STORN a J. A. LAMPINEN. *Differential Evolution - A Practical Approach to Global Optimization*. New York: Springer, 2005. ISBN 978-3540209508.
- [51] LAMPINEN, J a I ZELINKA. On Stagnation of the Differential Evolution Algorithm. In: *Proceedings of MENDEL 2000, 6th International Mendel Conference on Soft Computing*. 2000, s. 76–83. ISBN 80-214-1609-2.
- [52] QIN, A.K. a P.N. SUGANTHAN. Self-adaptive Differential Evolution Algorithm

- for Numerical Optimization. In: *2005 IEEE Congress on Evolutionary Computation* [online]. B.m.: IEEE, 2005, s. 1785–1791. ISBN 0-7803-9363-5. Dostupné z: doi:10.1109/CEC.2005.1554904
- [53] BREST, Janez, Sao GREINER, Borko BOSKOVIC, Marjan MERNIK a Viljem ZUMER. Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. *IEEE Transactions on Evolutionary Computation* [online]. 2006, **10**(6), 646–657. ISSN 1089-778X. Dostupné z: doi:10.1109/TEVC.2006.872133
- [54] BREST, J., A. ZAMUDA, I. FISTER a B. BOSKOVIC. Some Improvements of the Self-Adaptive jDE Algorithm. In: *2014 IEEE Symposium on Differential Evolution (SDE)* [online]. B.m.: IEEE, 2014, s. 1–8. ISBN 978-1-4799-4461-3. Dostupné z: doi:10.1109/SDE.2014.7031537
- [55] DAS, Swagatam, Ajith ABRAHAM, Uday K. CHAKRABORTY a Amit KONAR. Differential Evolution Using a Neighborhood-Based Mutation Operator. *IEEE Transactions on Evolutionary Computation* [online]. 2009, **13**(3), 526–553. ISSN 1089-778X. Dostupné z: doi:10.1109/TEVC.2008.2009457
- [56] DAS, Swagatam, Sankha Subhra MULLICK a P.N. SUGANTHAN. Recent advances in differential evolution – An updated survey. *Swarm and Evolutionary Computation* [online]. 2016, **27**, 1–30. ISSN 22106502. Dostupné z: doi:10.1016/j.swevo.2016.01.004
- [57] KENNEDY, James a Russell C EBERHART. Particle swarm optimization. In: *Neural Networks, 1995. Proceedings., IEEE International Conference on* [online]. 1995, s. 1942–1948 roč.4. ISBN 0-7803-2768-3. Dostupné z: doi:10.1109/ICNN.1995.488968
- [58] SHI, Yuhui a Russell C EBERHART. A modified particle swarm optimizer. In: *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)* [online]. 1998, s. 69–73. ISBN 0-7803-4869-9. Dostupné z: doi:10.1109/ICEC.1998.699146
- [59] EBERHART, R.C. a Y. SHI. Comparing inertia weights and constriction factors in particle swarm optimization. In: [online]. 2000, s. 84–88. ISBN 0-7803-6375-2. Dostupné z: doi:10.1109/CEC.2000.870279
- [60] IDE, Azuma a Keiichiro YASUDA. A basic study of adaptive particle swarm optimization. *Electrical Engineering in Japan (English translation of Denki*

- Gakkai Ronbunshi*) [online]. 2005, **151**(3), 41–49. ISSN 04247760. Dostupné z: doi:10.1002/ej.20077
- [61] SHI, Yuhui a R.C. EBERHART. Fuzzy adaptive particle swarm optimization. In: *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)* [online]. 2001, s. 101–106. ISBN 0-7803-6657-3. Dostupné z: doi:10.1109/CEC.2001.934377
- [62] ZHAN, Zhi-Hui, Jun ZHANG, Yun LI a Henry Shu-Hung CHUNG. Adaptive particle swarm optimization. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society* [online]. 2009, **39**(6), 1362–1381. ISSN 1941-0492. Dostupné z: doi:10.1109/TSMCB.2009.2015956
- [63] ZHU, Xianming a Hongbo WANG. A new inertia weight control strategy for particle swarm optimization. In: *ADVANCES IN MATERIALS, MACHINERY, ELECTRONICS II: Proceedings of the 2nd International Conference on Advances in Materials, Machinery, Electronics (AMME 2018)* [online]. B.m.: American Institute of Physics, 2018, s. 040095-1-040095-5. ISBN 9780735416543. Dostupné z: doi:10.1063/1.5033759
- [64] KENNEDY, James, Russell C EBERHART a Yuhui SHI. *Swarm Intelligence*. B.m.: Morgan Kaufmann, 2001. ISBN 9781558605954.
- [65] LEE, Kwang Y. a Mohamed A. EL-SHARKAWI, ed. *Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems*. B.m.: Wiley-IEEE Press, 2008. ISBN 9780471-45711-4.
- [66] KENNEDY, J. a R.C. EBERHART. A discrete binary version of the particle swarm algorithm. In: *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation* [online]. 1997, s. 4104–4108. ISBN 0-7803-4053-1. Dostupné z: doi:10.1109/ICSMC.1997.637339
- [67] YOSHIDA, Hirotaka, Kenichi KAWATA, Yoshikazu FUKUYAMA, Shinichi TAKAYAMA a Yosuke NAKANISHI. A Particle swarm optimization for reactive power and voltage control considering voltage security assessment. *IEEE Transactions on Power Systems* [online]. 2000, **15**(4), 1232–1239. ISSN 08858950. Dostupné z: doi:10.1109/59.898095
- [68] NELDER, J. A. a R. MEAD. A Simplex Method for Function Minimization. *The Computer Journal* [online]. 1965, **7**(4), 308–313. ISSN 0010-4620. Dostupné z: doi:10.1093/comjnl/7.4.308

- [69] SPENDLEY, W., G. R. HEXT a F. R. HIMSWORTH. Sequential Application of Simplex Designs in Optimisation and Evolutionary Operation. *Technometrics* [online]. 1962, **4**(4), 441–461. ISSN 0040-1706. Dostupné z: doi:10.1080/00401706.1962.10490033
- [70] HAN, Lixing a Michael NEUMANN. Effect of dimensionality on the Nelder-Mead simplex method. *Optimization Methods and Software* [online]. 2006, **21**(1), 1–16. ISSN 10556788. Dostupné z: doi:10.1080/10556780512331318290
- [71] GAO, Fuchang a Lixing HAN. Implementing the Nelder-Mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications* [online]. 2012, **51**(1), 259–277. ISSN 0926-6003. Dostupné z: doi:10.1007/s10589-010-9329-3
- [72] LEE, Donghoon a Matthew WISWALL. A parallel implementation of the simplex function minimization routine. *Computational Economics* [online]. 2007, **30**(2), 171–187. ISSN 09277099. Dostupné z: doi:10.1007/s10614-007-9094-2
- [73] MARIANO, Artur, Paulo GARCIA a Tiago GOMES. SW and HW speculative Nelder-Mead execution for high performance unconstrained optimization. In: *2013 International Symposium on System on Chip (SoC)* [online]. B.m.: IEEE, 2013, s. 1–5. ISBN 978-1-4799-1191-2. Dostupné z: doi:10.1109/ISSoC.2013.6675279
- [74] PANT, Millie, Radha THANGARAJ, Crina GROSAN a Ajith ABRAHAM. Hybrid differential evolution - Particle Swarm Optimization algorithm for solving global optimization problems. In: *2008 Third International Conference on Digital Information Management* [online]. B.m.: IEEE, 2008, s. 18–24. ISBN 978-1-4244-2916-5. Dostupné z: doi:10.1109/ICDIM.2008.4746766
- [75] BO YANG, YUNPING CHEN a ZUNLIAN ZHAO. A Hybrid Evolutionary Algorithm by Combination of PSO and GA for Unconstrained and Constrained Optimization Problems. In: *2007 IEEE International Conference on Control and Automation* [online]. B.m.: IEEE, 2007, s. 166–170. ISBN 978-1-4244-0817-7. Dostupné z: doi:10.1109/ICCA.2007.4376340
- [76] HAO, Zhi-Feng, Guang-Han GUO a Han HUANG. A Particle Swarm Optimization Algorithm with Differential Evolution. In: *2007 International Conference on Machine Learning and Cybernetics* [online]. B.m.: IEEE, 2007, s. 1031–1035. ISBN 978-1-4244-0972-3. Dostupné z: doi:10.1109/ICMLC.2007.4370294
- [77] WU, Lianghong, Yaonan WANG, Xiaofang YUAN a Shaowu ZHOU. A hybrid

- simplex differential evolution algorithm. In: *2010 Chinese Control and Decision Conference* [online]. B.m.: IEEE, 2010, s. 3039–3045. ISBN 978-1-4244-5181-4. Dostupné z: <http://ieeexplore.ieee.org/document/5498661/>
- [78] LI, Ji-min, Yeong-Cheng LIOU a Li-jun ZHU. Optimization of PID parameters with an improved simplex PSO. *Journal of Inequalities and Applications* [online]. 2015, **2015**(1), 325. ISSN 1029-242X. Dostupné z: doi:10.1186/s13660-015-0785-2
- [79] KUBAŘ, Miloslav. *Novel Optimization Tool for Analog Integrated Circuits Design*. B.m., 2013. b.n.
- [80] UN-KU MOON. CMOS high-frequency switched-capacitor filters for telecommunication applications. *IEEE Journal of Solid-State Circuits* [online]. 2000, **35**(2), 212–220. ISSN 0018-9200. Dostupné z: doi:10.1109/4.823446
- [81] VAIDYANATHAN, P. Quadrature mirror filter banks, M-band extensions and perfect-reconstruction techniques. *IEEE ASSP Magazine* [online]. 1987, **4**(3), 4–20. ISSN 0740-7467. Dostupné z: doi:10.1109/MASSP.1987.1165589
- [82] VAIDYANATHAN, P.P. Multirate digital filters, filter banks, polyphase networks, and applications: a tutorial. *Proceedings of the IEEE* [online]. 1990, **78**(1), 56–93. ISSN 00189219. Dostupné z: doi:10.1109/5.52200
- [83] HOSPODKA, Jiri, Pavel SOVKA a Bohumil PSENICKA. Design and realization of a filter bank by switched capacitor technique. In: *2011 20th European Conference on Circuit Theory and Design (ECCTD)* [online]. B.m.: IEEE, 2011, s. 753–756. ISBN 978-1-4577-0617-2. Dostupné z: doi:10.1109/ECCTD.2011.6043852
- [84] *Internetová stránka tvůrce programu Maple, firmy Maplesoft, a division of Waterloo Maple Inc.* [online]. [vid. 2018-08-20]. Dostupné z: <https://www.maplesoft.com/>
- [85] *Internetová stránka tvůrce programu WinSpice* [online]. [vid. 2018-08-20]. Dostupné z: <http://www.winspice.com/>
- [86] UBLD.IT. *TrueRNG v3 – Hardware Random Number Generator* [online]. [vid. 2018-08-20]. Dostupné z: http://ubld.it/truerng_v3
- [87] BAKER, R. Jacob. *CMOS: Circuit Design, Layout, and Simulation*. B.m.: Wiley-IEEE Press, 2011. ISBN 978-0-470-88132-3.
- [88] SANSEN, Willy M. C. *Analog Design Essentials* [online]. Boston, MA: Springer US, 2006. The International Series in Engineering and Computer Science.

ISBN 978-0-387-25746-4. Dostupné z: doi:10.1007/b135984

[89] *Analog devices, Inc., AD8033 Data Sheet, Rev C* [online] April 2008 [vid. 2018-8-25], URL: <http://www.analog.com/>

7 Seznam autorových publikací

7.1 Publikace vztahující se k tématu práce

Publikace v impaktovaných časopisech

[A1] NÁHLÍK, Jiří, Jiří HOSPODKA, Pavel SOVKA a Bohumil PŠENIČKA. Implementation of a Two-Channel Maximally Decimated Filter Bank using Switched Capacitor Circuits. *Radioengineering*. 2013, **22**(1), s. 167-173. ISSN 1210-2512. Dostupné z: http://www.radioeng.cz/fulltexts/2013/13_01_0167_0173.pdf

[A2] NÁHLÍK, Jiří, Jiří HOSPODKA, Pavel SOVKA, Brendan MULLANE a Ondřej ŠUBRT. Performance Optimization Methods for Switched-Capacitor Biquadratic Filters. *Journal of Electrical Engineering*. 2018, ISSN 1339-309X. Dostupné z: <http://iris.elf.stuba.sk/JEEEC/> (přijat k publikaci 28.8.2018) [65%]

Publikace na konferencích indexovaných ve WoS

[A3] NÁHLÍK, Jiří, Jiří HOSPODKA, Pavel SOVKA a Bohumil PŠENIČKA. Switched Capacitor Circuit Implementation of Two-Channel Filter Bank. In: *18th 2013 International Conference on Applied Electronics*. 18th International Conference on Applied Electronics 2013. Plzeň, 10.09.2013 - 12.09.2013. Pilsen: University of West Bohemia. 2013, s. 193-196. ISSN 1803-7232. ISBN 978-80-261-0166-6. [30%]

[A4] NÁHLÍK, Jiří, Jiří HOSPODKA, Pavel SOVKA, Ondřej ŠUBRT a Adán Bonilla CHÁVEZ. Optimization Methods for Switched Capacitor Circuits. In: *2015 International Conference on Applied Electronics*. 2015 International Conference on Applied Electronics 2015. Plzeň, 08.09.2015 - 09.09.2015. Plzeň: Západočeská univerzita. 2015, s. 165-168. ISSN 1803-7232. ISBN 978-80-261-0385-1.

[A5] NÁHLÍK, Jiří, Jiří HOSPODKA a Ondřej ŠUBRT. SC Filter Optimization Performance by Hybrid Simplex Algorithm, In: *2018 15th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, Prague, Czech Republic, 2018, pp. 285-288. doi: 10.1109/SMACD.2018.8434885 [70%]

Publikace ve sbornících konferencí

[A6] NÁHLÍK Jiří, Brendan MULLANE, Jiří HOSPODKA, Pavel SOVKA, a Daniel O'HARE Optimized switched capacitor biquads for two-channel quadrature-mirror filter bank . In: *IET Irish Signals & Systems Conference 2014*. IET Irish Signals & Systems Conference 2014. Limerick, 26.06.2014 - 27.06.2014. Stevenage: IET Publishing Group. 2014, s. 412-417. ISBN 978-1-84919-924-7.

[A7] NÁHLÍK Jiří. Simulation of the Switched Capacitor Filter Bank by WinSpice program. In: *POSTER 2012 - 16th International Student Conference on Electrical Engineering*. POSTER 2012 - 16th International Student Conference on Electrical Engineering. Prague, 17.05.2012. Praha: Czech Technical University in Prague. 2012, s. 1-4. ISBN 978-80-01-05043-9.

[A8] NÁHLÍK Jiří a Jiří HOSPODKA. Techniky implementace banky filtrů pomocí SC obvodů. In: ČMEJLA, R., RUSZ, J., a ŠPULÁK, D., eds. *LETNÍ DOKTORANDSKÉ DNY 2012. 2. LETNÍ DOKTORANDSKÉ DNY*. Praha, 24.05.2012 - 25.05.2012. Praha: České vysoké učení technické v Praze. 2012, s. 24-29. ISBN 978-80-01-05050-7. Dostupné z: http://sami.fel.cvut.cz/LDD12/LDD12_sbornik_prispevku.pdf

[A9] NÁHLÍK Jiří a Jiří HOSPODKA. Realizace banky filtrů technikou spínaných kapacitorů [online]. In: ČMEJLA, R., ŠPULÁK, D., a RUSZ, J., eds. *III. LETNÍ DOKTORANDSKÉ DNY 2013*. Letní doktorandské dny 2013. Praha, 23.05.2013 - 24.05.2013. Praha: České vysoké učení technické v Praze, Fakulta elektrotechnická. 2013, s. 6-12. ISBN 978-80-01-05251-8. Dostupné z: <http://sami.fel.cvut.cz/LDD13/sborn%C3%ADk%20LDD%202013.pdf>

[A10] NÁHLÍK Jiří a Jiří HOSPODKA. Optimalizační metody pro reálné SC struktury [online]. In: RUSZ, J., ČMEJLA, R., a SEDLÁK, J., eds. *V. Letní doktorandské dny 2015*. Letní doktorandské dny 2015. Praha, 28.05.2015 - 29.05.2015. Praha: ČVUT FEL, Katedra teorie obvodů. 2015, s. 23-29. ISBN 978-80-01-05749-0. Dostupné z: http://sami.fel.cvut.cz/LDD15/Sbornik_LDD2015.pdf

[A11] NÁHLÍK Jiří a Jiří HOSPODKA. Efektivita optimalizačních metod pro SC obvody [online]. In: *978-80-01-06161-9. LDD'17 Letní doktorandské dny*. Praha, 31.05.2017. Praha: České vysoké učení technické v Praze. 2017, s. 20-24. ISBN 978-80-01-06161-9. Dostupné z: <http://sami.fel.cvut.cz/ldd17/> [85%]

Funkční vzorky

[A12] NÁHLÍK Jiří a Jiří HOSPODKA. Banka filtrů se spínanými kapacitorem. [Funkční vzorek]. 2013

Citace autorových publikací

Publikace [A1] byla citována v

MILIĆ Miljana a Vančo LITOVSKI. Oscillation-based testing method for detecting switch faults in High-Q SC biquad filters In: *Facta universitatis - series: Electronics and Energetics*, 2015, **28(2)**, s. 223-236. doi. 10.2298/FUEE1502223M

7.2 Ostatní publikace a výsledky nevztahující se k tématu práce

[A13] HOSPODKA Jiří a Jiří NÁHLÍK. Analýza elektronických obvodů a programy pro jejich simulaci. *Sdělovací technika*. 2013, **61**(2/2013), s. 5-9. ISSN 0036-9942. Dostupné z: <http://www.floowie.com/cs/cti/st-unor-2013a/#/strana/6/> [30%]

[A14] BORTEL, Radoslav. Jiří HOSPODKA, Jan ŠEBEK, Jiří NÁHLÍK a Pavel SOVKA. Modul elektroencefalografu s rozhraním pro optický stimulator. [Funkční vzorek]. 2016 [15%]