

3. POPIS PROGRAMOVÉHO VYBAVENÍ

Výše popsaná teorie a algoritmy byly ověřovány paralelně probíhajícími programátorskými pracemi. Programy byly vytvářeny v prostředí Borland Pascalu 7.0. Konečným cílem je vytvoření profesionálního programu pro přesnou simulaci reálných spínaných obvodů.

V této části práce popíšeme základní výsledky úvodní etapy řešení, kterou je tvorba a ověřování výpočetních algoritmů. Nejprve uvedeme stručný popis některých programových jednotek, pak hlavní parametry vystihující výkonnost současné verze programu a nakonec ukázky simulace.

3.1. Základní programové jednotky

Jednotka **GLOBAL**.

V této jednotce jsou deklarovány globální proměnné, konstanty a typy programu.

Jednotka **KOMPLEX**.

Je to jednotka komplexní aritmetiky. Obsahuje procedury a funkce pro níže uvedené operace v komplexním oboru. Některé z procedur jsou naprogramovány ve více variantách s optimalizací buď na přesnost nebo rychlost výpočtů.

- Sčítání, odečítání, násobení a dělení dvou komplexních čísel.
- Vzájemné sčítání, odečítání, násobení a dělení reálného a komplexního čísla.
- Výpočet reciproké hodnoty komplexního čísla.
- Výpočet modulu a argumentu komplexního čísla.
- Přepočet argumentu komplexního čísla tak, aby byl v mezích $-\pi$ až $+\pi$.
- Výpočet přirozeného logaritmu komplexního čísla.
- Výpočet exponenciální funkce komplexního čísla.
- Výpočet druhé odmocniny komplexního čísla.

Jednotka **MATROM**.

Jednotka maticových operací v reálném oboru. Obsahuje mj. tyto procedury a funkce:

- Rychlé nulování vektorů a matic.
- Vytváření jednotkové matice, resp. jednotkového vektoru.
- Kopírování vektoru do jiného vektoru.
- Kopírování vektoru do daného sloupce matice.
- Kopírování daného sloupce matice do vektoru.
- Násobení dvou matic.
- Násobení matice vektorem.
- Sčítání a odečítání dvou matic.
- Násobení vektoru skalárem.
- Sčítání a odečítání vektorů.
- Násobení a dělení matice skalárem.
- Výpočet kvadratické normy matice.

- Výpočet stopy matice.
- Výpočet funkce $\exp(\mathbf{A})$ čtvercové matice $\mathbf{A}$.
- LU rozklad matice.
- Řešení soustavy lineárních rovnic metodou LU rozkladu.
- Procedura zvyšující přesnost řešení soustavy rovnic.
- Inverze matice.
- Determinant matice.
- Řešení špatně podmíněné soustavy rovnic Gauss-Jordanovou metodou.

Jednotka **MATROMC**.

Jednotka maticových operací v komplexním oboru. Obsahuje procedury a funkce jednotky **MATROM** v komplexním oboru.

Jednotka **POLY**.

Jednotka pro přesný výpočet komplexních kořenů polynomu Laguerovou metodou. Pro zvýšení přesnosti je použita technika "roots polishing".

Jednotka **EIGENVAL**.

Jednotka pro výpočet vlastních čísel komplexní matice modifikovaným QR algoritmem. Obsahuje mj. procedury pro vyvažování matice a její transformaci na Hessenbergův tvar.

Jednotka **ILT**.

Jednotka inverzní Laplaceovy transformace. Obsahuje řadu procedur, které provedou inverzní Laplaceovu transformaci komplexní maticové rovnice

$$\mathbf{Y}^M(s)\mathbf{V}(s) = \mathbf{R}(s),$$

kde $\mathbf{R}(s)$ je komplexní vektor pravé strany. Procedury se liší použitou sadou Padéových koeficientů [34] a tudíž dosažitelnou přesností transformace. Výsledkem transformace je vektor $\mathbf{v}$ v zadaném časovém okamžiku.

Jednotka **SCKOEF**.

V této jednotce je naprogramována řada důležitých algoritmů, především algoritmus hledání nezávislých branových veličin, algoritmus tvorby obvodových matic a algoritmy semisymbolické analýzy.

Součástí vyvinutého programového vybavení je řada dalších jednotek pro vytváření a čtení datových souborů, pro grafickou prezentaci výsledků simulace a jiné podpůrné činnosti.

3.2. Současné možnosti programu

V současné době je vytvořena pracovní verze programu, která pracuje podle následujícího schématu:

Zadání obvodu.

Hledání soustavy nezávislých obvodových veličin.

Sestavení soustavy obvodových rovnic.

**Semisymbolická analýza zobecněných přenosových funkcí v oblasti operátoru z .
Kmitočtová analýza.**

Současná verze programu umožňuje řešit dvoufázové spínané obvody, obsahující ideální OZ a reálné OZ s jedním kmitočtem lomu, ideální zesilovače napětí, kapacitory, rezistory a spínače s nulovým i nenulovým odporem v sepnutém stavu. Složitost analyzovaného obvodu je omezena pamětí počítače (nyní orientačně: 40 uzlů, 30 kapacitorů, 30 rezistorů, 10 OZ, 50 spínačů). Program je průběžně testován a již v současné době poskytuje cenné služby při ověřování navržených zapojení spínaných filtrů. Počítá amplitudovou a fázovou kmitočtovou charakteristiku analyzovaného filtru včetně možnosti současného vykreslování několika průběhů při variaci parametrů. Výpočet probíhá za předpokladu buzení vstupní brány signály typu "Sample-Hold" podle variant, které jsou uvedeny v části 2.4. K dispozici je i výpočet pólů filtru v oblasti z a tudíž i informace o jeho stabilitě.

Součástí vývoje popisovaných algoritmů bylo jejich opakované testování a následné upravování. Ve zprávě je uvedena jejich konečná podoba.

V následující části podrobíme program jednoduchému "globálnímu" testu a poté ukážeme jeho možnosti při analýze složitějšího spínaného obvodu.

3.3. Příklady testování programu

Korektní ověření správnosti a přesnosti analýzy by mohlo vypadat takto:

- Zjistíme analytické vztahy pro koeficienty obvodových funkcí spínaného obvodu v oblasti z a dosazením číselných hodnot parametrů obvodu stanovíme jejich numerické hodnoty.
- Porovnáme koeficienty s výsledkem semisymbolické analýzy testovaného programu.

Tento postup je dobře aplikovatelný na idealizované SC obvody nízkého řádu, obsahující pouze ideální spínače s nulovým odporem v sepnutém stavu a nulovou vodivostí v rozepnutém stavu, ideální kapacitory a ideální nesetrvačné transformační členy. Pro tyto obvody lze totiž bez větších problémů stanovit systémovou funkci v oblasti z v symbolickém, resp. semisymbolickém tvaru. Pro tyto účely je možné využít například program COCOS [20].

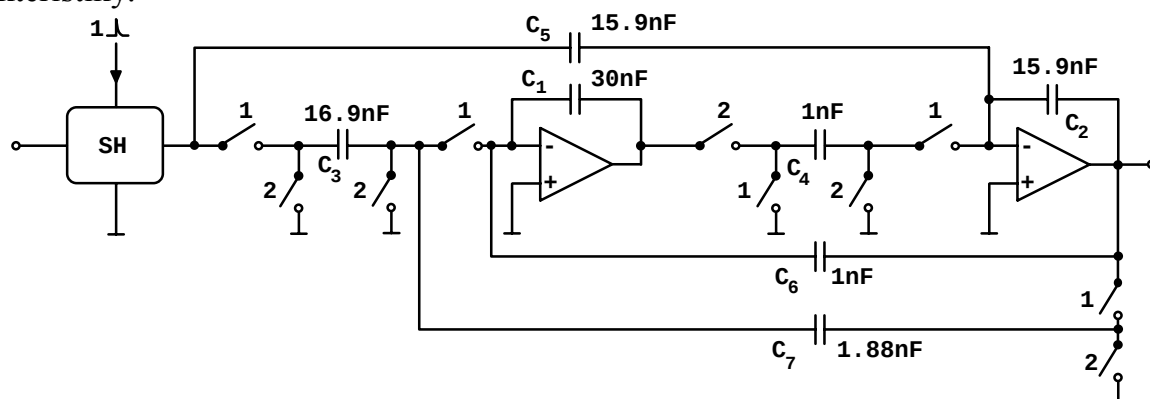
Obdobné výstupy však prakticky nejsou dostupné pro třídu reálných spínaných obvodů. Pak se musíme spokojit s méně průkazným testováním, založeným například na srovnání s výstupy jiných simulátorů nebo na posouzení reálnosti získaných výsledků v kmitočtové oblasti, t.j. zda kmitočtová charakteristika vykazuje očekávaný průběh.

Následuje ukázka testování přesnosti analýzy jak idealizovaných, tak i reálných spínaných obvodů.

Analýza idealizovaných SC obvodů

V [6] je publikováno zapojení spínaného eliptického filtru typu dolní propust s potlačením přenosu na kmitočtu rovném $0.03f_v$ a s rezonančním zesílením na kmitočtu

$0.01f_v$, kde f_v je vzorkovací kmitočet. Schéma filtru s konkrétními hodnotami součástek je na obr.10. Filtr byl syntetizován za předpokladu ideálních součástek a zpětně analyzován programem SCSK, který jsme v minulosti vytvořili pro analýzu idealizovaných SC obvodů. Výsledky analýzy potvrdily záměry syntézy. Analýza s uvažováním reálných vlastností součástek však nemohla být provedena, neboť nebyl k dispozici potřebný program. Měření na funkčním vzorku ovšem odhalilo podstatné odchylky od navrhované kmitočtové charakteristiky.



Obr.10. Příklad analyzovaného SC filtru.

V [5] je proveden výpočet systémových funkcí filtru v oblasti z . Jmenovatel všech systémových funkcí je stejný a nezávisí na způsobu výběru vstupních a výstupních vzorků. Jeho symbolický zápis je následující:

$$1 + a_1 z^{-1} + a_2 z^{-2},$$

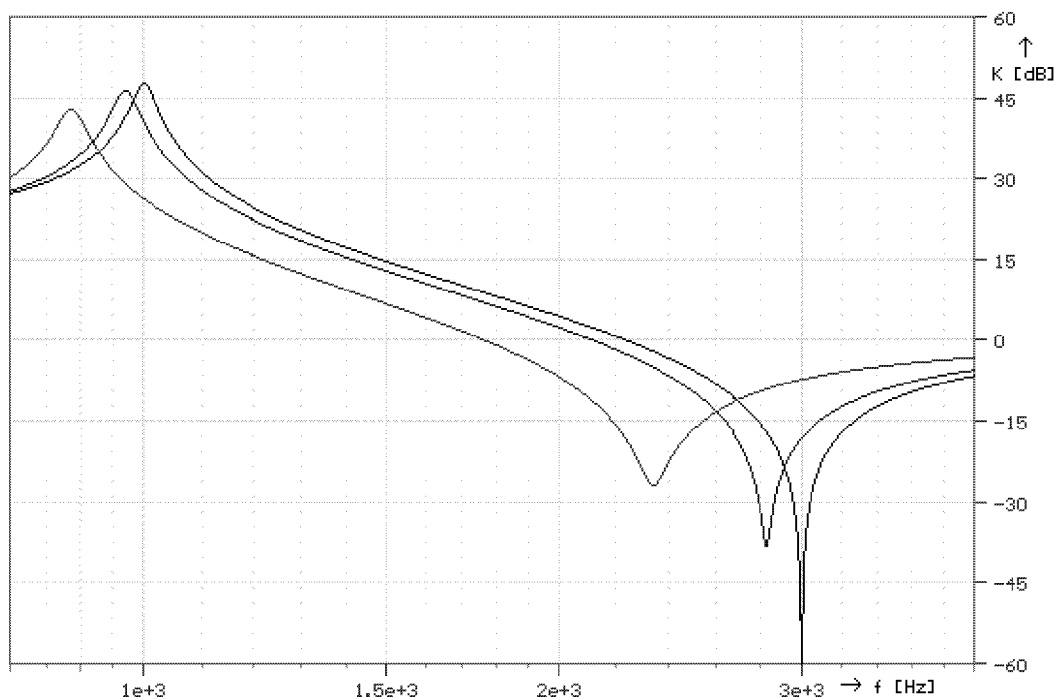
$$a_1 = \frac{C_4(C_6 + C_7)}{C_1 C_2} - 2, a_2 = 1 - \frac{C_4 C_6}{C_1 C_2}.$$

V Tab.3 jsou srovnány výsledky počítačové analýzy se správnými hodnotami. Vidíme shodu prakticky na všech dosažitelných desetinných místech.

koeficient	správná hodnota	program SCSK	testovaný program
a_1	-1.99396226415094339	-1.99396226417	-1.99396226415094
a_2	0.99790356394129979	0.997903563969	0.997903563941297

Tab.3. Výsledky testování přesnosti semisymbolické analýzy obvodu na obr.10.

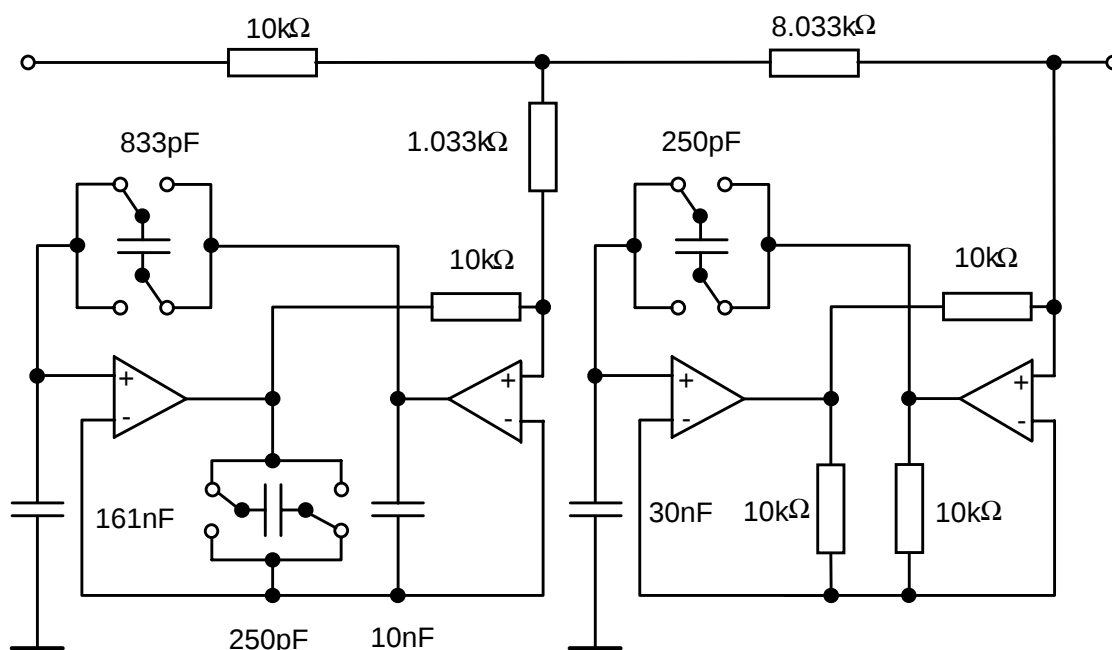
Na obr.11 jsou výsledky kmitočtové analýzy daného filtru. Simulace byla provedena pro vzorkovací kmitočet 100kHz. Analýza ukázala, že reálné vlastnosti operačních zesilovačů mají zanedbatelný vliv na průběh kmitočtové charakteristiky. Při analýze byl použit následující model operačního zesilovače: $A_0=200\,000$, $f_T=1\text{MHz}$, $R_o=0\Omega$. Podstatnou roli zde však hrají odpory spínačů v sepnutém stavu, což se dalo očekávat, uvažíme-li relace mezi vzorkovacím kmitočtem a poměrně velkými pracovními kapacitami. Z obrázku je zřejmé, že při použití polovodičových spínačů o typickém odporu v sepnutém stavu několik stovek ohmů bude skutečná charakteristika značně zdeformovaná.



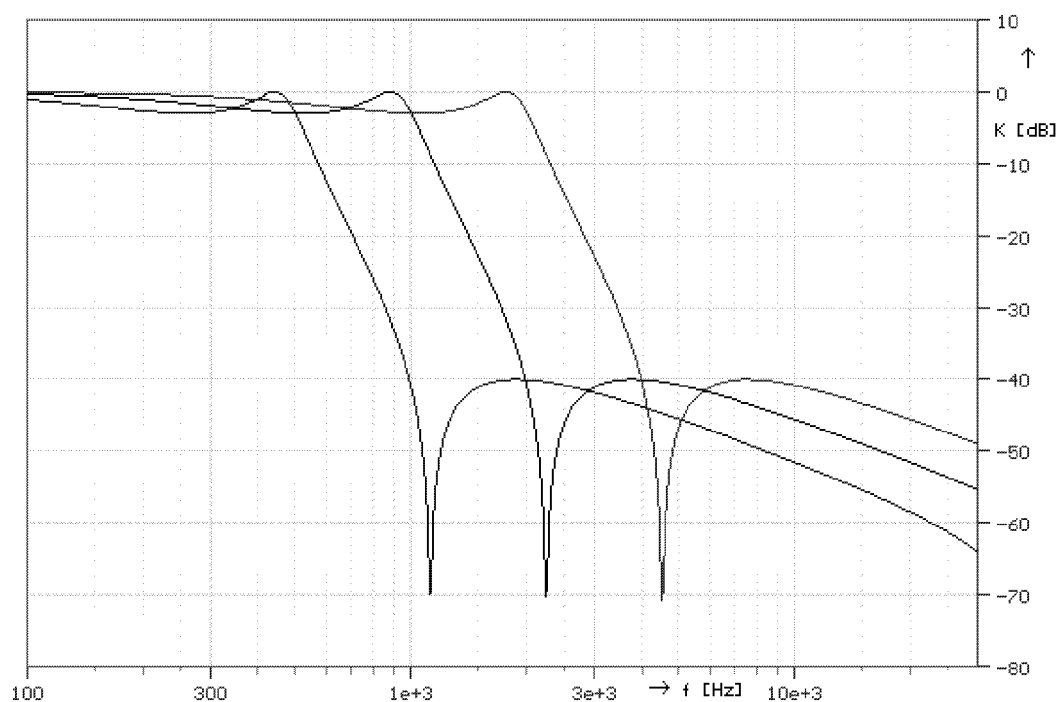
Obr.11. Kmitočtová analýza obvodu na obr.10. Vzorkovací kmitočet 100kHz. Odpor spínačů v sepnutém stavu: 1. 0Ω , 2. 50Ω , 3. 100Ω .

Analýza reálných spínaných obvodů.

V [17] a [22] je publikováno zapojení filtru typu ARSC, tj. aktivního filtru s rezistory a spínanými kapacitami. Rezistory jsou zde pracovními a nikoliv parazitními elementy. Schéma je na obr.12. Filtr byl navržen jako cauerovská dolní propust s mezním kmitočtem 1kHz při spínacím kmitočtu 100kHz. Na obr.13 jsou výsledky simulace, které jsou v soulase s navrhovanými parametry filtru. Obrázek ukazuje na možnost přeladování změnou vzorkovacího kmitočtu.



Obr.12. Simulovaný spínaný filtr [22].



Obr.13. Výsledky simulace - amplitudová kmitočtová charakteristika filtru z obr.12 pro různé vzorkovací kmitočty: **1.** 50kHz, **2.** 100kHz, **3.** 200kHz.
 Parametry operačních zesilovačů: $A_0=200\,000$, $f_T=10\text{MHz}$, $R_o=0\Omega$.
 $R_{spinače,ON} = 400\Omega$.

ZÁVĚR

V předložené práci je popsána základní teorie a algoritmy počítačové analýzy reálných spínaných obvodů. Praktickým výsledkem je software realizující vyvinuté algoritmy, umožňující již v tomto stadiu simulaci složitých struktur.

V nadcházejícím období bude potřebné:

- 1) Zdokonalit algoritmy kmitočtové analýzy a rozšířit výpočtové možnosti o případ buzení vstupním signálem bez SH obvodů. To si vyžádá kombinovanou semisymbolickou analýzu v rovině z i s .
- 2) Rozšířit knihovnu obvodových prvků o operační zesilovače s dvěma kmitočty lomu a výstupním odporem, o indukory a obvody s vzájemnou indukčností, o transkonduktanční zesilovače a další prvky.
- 3) Rozšířit výpočetní možnosti programu o časovou analýzu.
- 4) Po dobudování algoritmů přejít k úpravě uživatelského prostředí programu. Bude potřebné zejména vyřešit způsob zadávání obvodu formou schématického nebo tabulkového editoru a vytvořit standardní komunikační prostředí mezi programem a uživatelem.