

ÚVOD

Ve zprávě jsou shrnuty výsledky, kterých jsme dosáhli v oblasti vývoje softwarových nástrojů pro analýzu a testování obvodů s imitancemi vyšších řádů.

Dostupné obvodové simulátory typu SPICE, MicroCap a další vykazují řadu omezení týkajících se jak typů analyzovaných obvodů, tak i analyzačních režimů. Imitance vyšších řádů je například možné v programu MC4 modelovat pouze nepřímo pomocí řízených Laplaceových zdrojů, a to poměrně komplikovaným způsobem. Elektrické schéma je totiž třeba překreslit na schéma blokové typu „state variable diagram“ a bloky odpovídající imitancím vyšších řádů modelovat Laplaceovými zdroji. Výsledky simulace takovýchto struktur v kmitočtové oblasti jsou pak uspokojivé, což ovšem neplatí pro oblast časovou. Uvedené simulátory totiž postupují při časové analýze tak, že přenosovou funkci Laplaceova zdroje nejprve převedou pomocí IFFT (inverzní rychlé Fourierovy transformace) na impulsní odezvu a časové průběhy pak počítají konvolucí s budicím signálem. Výsledek pak odpovídá realitě jen v případech, kdy algoritmus IFFT vypočte všechny „reprezentativní vzorky“ impulsní charakteristiky. Zaniká-li impulsní odezva relativně pomalu, vzniká nepříjemný „okénkový efekt“ spojený s transformací IFFT a výsledky analýzy jsou zkresleny. Je-li analyzovaný systém derivačního charakteru, např. filtr typu horní propust, obsahuje teoretická impulsní odezva zobecněné impulsy v počátku časové osy, které nejsou při IFFT brány v úvahu a výsledky analýzy jsou pak zcela nepoužitelné.

K popisovaným situacím však často dochází při simulaci s imitancemi vyšších řádů, kdy modelovaný blok vykazuje vícenásobné nulové body nebo póly.

Dalším omezením, které je podstatné pro řešení našeho projektu, je neschopnost simulátorů analyzovat linearizované obvody v symbolické formě. Přitom symbolická analýza je potřebná v procesu syntézy jak k ověření vlastního principu navrhované struktury, tak i v její následné optimalizaci a dostavování. Kromě toho symbolická analýza ve spojení s numerickou analýzou může poskytovat velmi přesné řešení zatížené menšími numerickými chybami než metody založené na čistě numerických principech.

Dostupné simulátory neumožňují ani semisymbolickou analýzu linearizovaných obvodů, tj. nalezení numerických hodnot koeficientů obvodových funkcí. Taktéž nelze určovat nulové body a póly. O stabilitě lze rozhodnout pouze nepřímo prostřednictvím časové analýzy.

Výše uvedené důvody nás vedly k vývoji analyzačního programu se zaměřením na řešení obvodů s imitancemi vyšších řádů. Za základ jsme zvolili jednu z verzí programu COCO pro symbolické výpočty, kterou jsme vytvořili v počátečních etapách řešení projektu [3].

KONCEPCE ŘEŠENÍ

Zjednodušený blokový diagram programu je uveden na obr.1. Některé algoritmy stávajícího programu COCO pro symbolickou analýzu jsou začleněny do větve „Symbolické výpočty“. V počáteční fázi analýzy se může uživatel programu rozhodnout, zda analyzovat obvod symbolicky nebo semisymbolicky. Možnost této volby je výhodná zejména při analýze rozsáhlejších obvodů, kdy jednak výsledky symbolické analýzy jsou v nepřehledném a mnohdy nevyužitelném tvaru, jednak doba analýzy může být neúnosně dlouhá. Pak je možné přejít přímo na semisymbolickou analýzu, jejíž algoritmy jsou podstatně rychlejší, přesnost stanovení koeficientů obvodových funkcí je však obecně nižší. Na druhou stranu je však semisymbolická analýza založena na hledání nulových bodů a pólů pomocí algoritmu pro hledání vlastních čísel obvodových matic. Z nulových bodů a pólů jsou následně získávány koeficienty obvodové funkce jednoduchým roznásobováním kořenových činitelů. U symbolické analýzy je situace opačná - nulové body a póly jsou určovány z koeficientů hledáním kořenů, což je postup méně přesný než procedura pro hledání vlastních čísel obvodové matice.

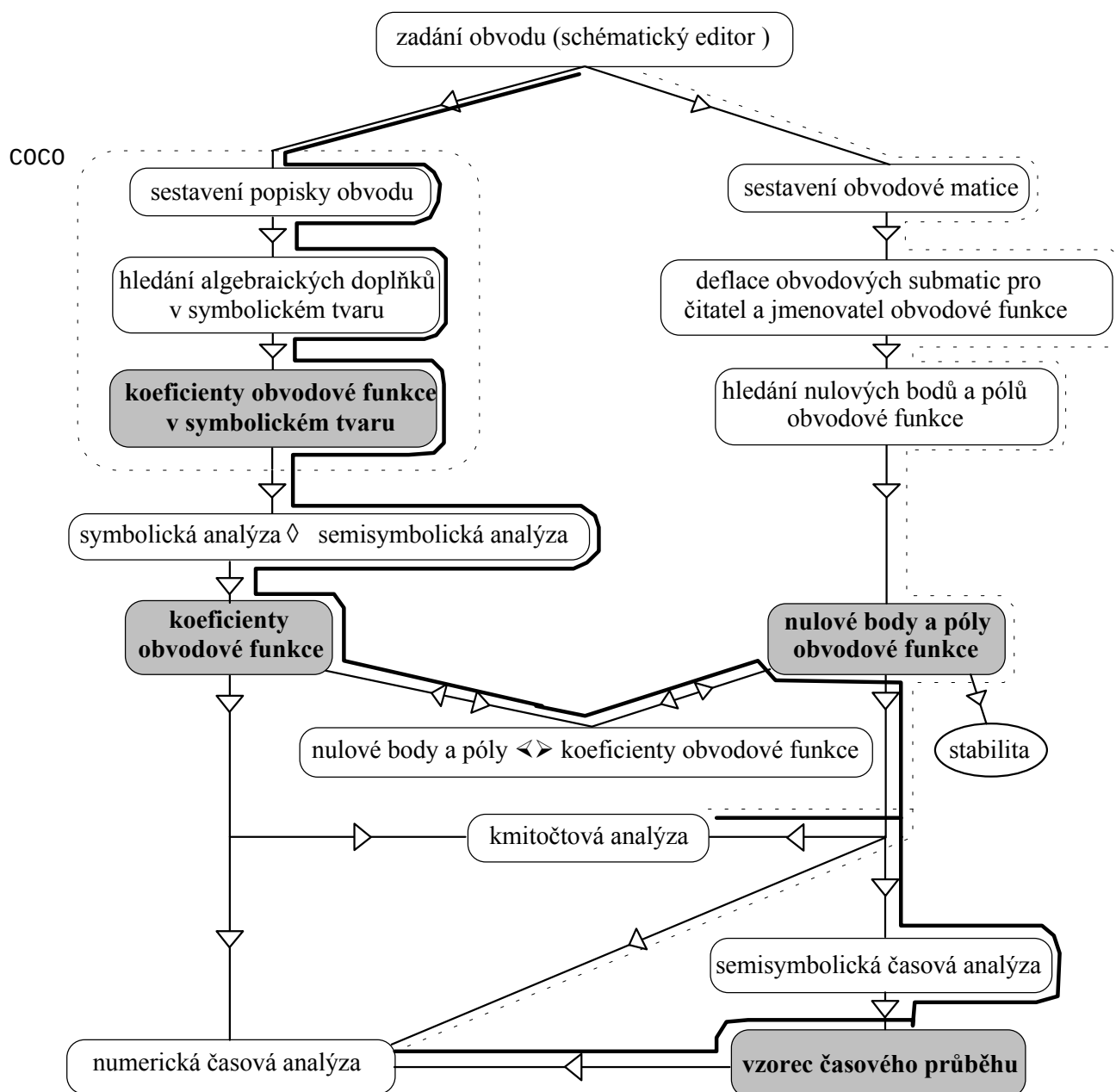
Kmitočtová analýza může vycházet buď z nulových bodů a pólů, nebo z koeficientů obvodové funkce. První přístup se ukázal jako výhodnější z několika důvodů. Poskytuje přesnější výsledky u obvodů vysokých řádů. Výpočet je dostatečně rychlý, protože se berou v úvahu jen skutečně existující póly a zejména nulové body. Z nulových bodů a pólů je možno efektivně počítat kmitočtový průběh skupinového zpoždění.

Časovou analýzu je možno realizovat dvěma odlišnými způsoby, semisymbolicky nebo numericky. Semisymbolická analýza je založena na automatickém rozkladu obvodové funkce na parciální zlomky. Využívá se zde znalosti nulových bodů a pólů. Póly musí být stanoveny s vysokou přesností, zejména v případě jejich násobnosti, neboť násobné póly vedou na odlišné rozkladové vzorce. Výsledkem semisymbolické analýzy je vzorec pro vynucenou odezvu obvodu na budící signál, tj. odezvu za předpokladu nulového výchozího energetického stavu. K dispozici je knihovna jednoduchých budících signálů (Diracova distribuce, skoková funkce, lichoběžníkový signál, harmonický signál).

Numerická časová analýza v jednodušším případě zabezpečuje výpočet časových průběhů z vzorce získaného semisymbolickou analýzou. Lze ji však provozovat v čistě numerickém nebo hybridním režimu. Čistě numerický režim je založen na původním algoritmu zpětné Laplaceovy transformace, který jsme vyvinuli pro časovou analýzu obvodů popsaných přenosovou funkcí [1], [2]. Metoda nejprve přiřazuje přenosové funkci stavový popis v 1. kanonické struktuře. Následuje numerické řešení příslušné stavové rovnice. Využitím stavového popisu se nám podařilo rozšířit možnosti analýzy o okamžité nalezení ustáleného stavu v případě periodického buzení. Hybridní režim využívá rozkladu obvodové funkce na parciální zlomky a modeluje Jordanovskou strukturu obvodu s následujícím numerickým řešením odpovídající komplexní stavové rovnice.

Z uvedeného plyne, že uživatel programu má více možností, jak provést příslušnou analýzu. Výběr bude záviset mj. na tom, zda upřednostňuje časově nenáročný výpočet, numerickou přesnost atd. Přednastavili jsme dva standardní průchody programem, „symbolický“ a „hybridní“. Oba průchody jsou v obr.1 graficky vyznačeny. Kombinace

obou cest je možná a zkušenější uživatel tak může optimalizovat řešení svého konkrétního problému.



Obr.1. Struktura programu; — symbolický, ----- hybridní průchod programem.

Princip symbolických výpočtů, které jsou obsahem bloků souhrnně označených jako „COCO“ je popsán v [3] a [4]. V dalším textu popíšeme podstatu nově vyvinutých algoritmů v ostatních blocích.

SESTAVENÍ OBVODOVÝCH MATIC

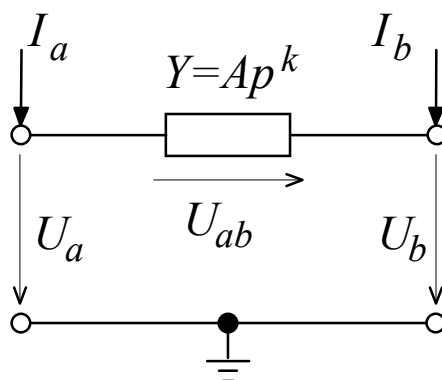
Pro maticový popis obvodů s imitancemi vyšších řádů byla vybrána modifikovaná metoda uzlových napětí. Vývojem efektivních metod popisu linearizovaných obvodů jsme se začali zabývat již v minulém roce řešení projektu. Ve výzkumné zprávě [5] jsme popsali způsoby začlenění modelů proudových konvektorů a transimpedančních operačních zesilovačů do maticových rovnic. V této kapitole navážeme na [5] a ukážeme jeden ze způsobů modelování imitancí vyšších řádů, který jsme vyvinuli pro potřeby semisymbolické analýzy obvodových funkcí.

Uvažujme plovoucí admitanci k -tého řádu

$$Y(p) = Ap^k, \quad (1)$$

která je zapojena do obvodu podle obr.2. Metoda uzlových napětí vede na rovnici

$$\begin{bmatrix} Ap^k & -Ap^k \\ -Ap^k & Ap^k \end{bmatrix} \begin{bmatrix} U_a \\ U_b \end{bmatrix} = \begin{bmatrix} I_a \\ I_b \end{bmatrix}. \quad (2)$$



Obr.2. Plovoucí admitance k -tého řádu, $k \geq 0$.

Maticová rovnice je sestavována za účelem semisymbolické analýzy obvodových funkcí. Jestliže je jako nástroj analýzy použit algoritmus zobecněné úlohy hledání vlastních čísel obvodové matice, musí se Laplaceův operátor vyskytovat v pseudoadmitanční matici pouze v nulté a první mocnině. Toho lze dosáhnout dodefinováním dalších proměnných a jejich začleněním do obvodové matice podle následujícího zápisu:

$$\begin{bmatrix} & & & & & Ap \\ & & & & & -Ap \\ p & -1 & & & & \\ & & p & -1 & & \\ \vdots & & & & & \vdots \\ & & & & p & -1 \end{bmatrix} \begin{bmatrix} U_a \\ U_b \\ U_{ab}^{(1)} \\ U_{ab}^{(2)} \\ \vdots \\ U_{ab}^{(k-1)} \end{bmatrix} = \begin{bmatrix} I_a \\ I_b \\ \\ \\ \\ \end{bmatrix}. \quad (3)$$

V rovnici je zavedeno $k-1$ nových proměnných

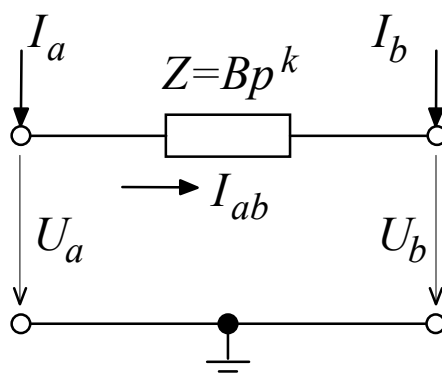
$$U_{ab}^{(i)} = p^i (U_a - U_b), i = 1, 2, \dots, k-1 \quad (4)$$

díky nimž jsme dosáhli potřebného snížení mocniny operátoru, ovšem za cenu rozšíření řádu obvodové matice o číslo $k-1$. Matice je však velmi řídká, čehož lze využít při následné deflaci a hledání vlastních čísel. Řádky pseudoadmitanční matice označené $\bullet$ jsou tzv. zakázané řádky [5], což znamená, že v procesu doplňování úplné pseudoadmitanční matice o submatice dalších obvodových prvků v obvodu již nesmíme modifikovat označené řádky, jinak by totiž došlo k nežádoucí modifikaci vazebních rovnic v nich zakódovaných.

O něco komplikovanější je modelování impedance k -tého řádu

$$Z(p) = Bp^k \quad (5)$$

podle obr.3. Jak známo, impedance je cizorodým prvkem v metodě uzlových napětí. To se projeví ve větším počtu přídavných rovnic:



Obr.3. Plovoucí impedance k -tého řádu, $k \geq 0$.

		1				
		-1				
-1	1					Bp
		p	-1			
			p	-1		
$\vdots$					$\vdots$	
					p	-1

 $\bullet$

U_a
U_b
I_{ab}
$I_{ab}^{(1)}$
$I_{ab}^{(2)}$
$\vdots$
$I_{ab}^{(k-1)}$

 $=$

I_a
I_b

 $\cdot$
(6)

V rovnici je tentokrát zavedeno k nových proměnných

$$I_{ab}, I_{ab}^{(i)} = p^i I_{ab}, i = 1, 2, \dots, k-1, \quad (7)$$

kde I_{ab} je proud tekoucí popisovanou impedancí (viz obr.3).

V programu je pseudoadmitanční matice deklarována jako matice komplexní, kde k imaginární matici je přiřazen multiplikativní operátor p .

DEFLACE OBVODOVÝCH SUBMATIC

Obvodová funkce je získána ve tvaru racionální lomené funkce, kde v čitateli a jmenovateli jsou příslušné algebraické doplňky obvodové matice. Při semisymbolické analýze postupujeme podle následujícího schématu:

1. Podle povahy obvodové funkce vynecháme příslušný řádek a sloupec pseudoadmitanční matice.
2. Na vzniklou submatici aplikujeme deflační algoritmus, tj. takovou posloupnost lineárních operací s řádky a sloupci matice, který převede úlohu výpočtu jejího determinantu na standardní úlohu hledání vlastních čísel.
3. Nalezneme vlastní čísla vzniklé struktury, což jsou nulové body obvodové funkce, jejich počet, což je řád čitatele, a násobící koeficient čitatele.
4. Body 1-3 opakujeme pro jmenovatel obvodové funkce, čímž nalezneme póly obvodové funkce, jejich počet, což je řád jmenovatele, a násobící koeficient jmenovatele. Násobící koeficienty čitatele a jmenovatele sloučíme do jediného koeficientu.
5. Zkontrolujeme, zda se neshodují některé nulové body a póly, a v kladném případě provedeme jejich krácení.

Na vývoji deflačních algoritmů pracujeme již delší dobu, viz např. [6]. Je tomu tak proto, že na způsobu deflace submatice mnohdy silně závisí rychlost a zejména přesnost nalezení všech vlastních čísel. Některé deflační algoritmy již v průběhu deflace separují některá vlastní čísla a výrazně tak zjednodušují následnou úlohu hledání vlastních čísel. Všechny nám známé algoritmy však mají jeden společný rys - jsou postaveny jako čistě matematická úloha nad obecnou čtvercovou komplexní maticí, kdy se nevyužívá apriorních znalostí struktury této matice. Vezmeme-li v úvahu specifickou strukturu poměrně řídké obvodové matice sestavené algoritmem modifikované metody uzlových napětí, můžeme proces deflace zefektivnit.

Práce na novém deflačním algoritmu jsou téměř ukončeny, pokud jde o obvody s regulárními prvky a o jednoduché neregulární prvky typu induktor a operační zesilovač. Matice popisující imitance vyšších řádů vykazují zvláštní strukturu (viz rovnice 3 a 6) a tento případ není zatím do našeho algoritmu zahrnut. Z těchto důvodů jsme do programu prozatímne použili spolehlivý deflační algoritmus, který se osvědčil ve výpočetních procedurách simulátoru C.I.A.

Výpis zdrojového programu vyvinutého deflačního algoritmu je uveden v Příloze 1. V Příloze 2 je výpis procedur použitého deflačního algoritmu spolu s algoritmem pro hledání vlastních čísel obvodové matice.

HLEDÁNÍ NULOVÝCH BODŮ A PÓLŮ OBVODOVÉ FUNKCE

Hledání nulových bodů, resp. pólů je řešeno ve třech klasických krocích:

1. Numerické vyvažování reálné čtvercové matice, získané deflací obvodové submatice.
2. Transformace vyvážené matice na Hessenbergův tvar.
3. Aplikace algoritmu QR s úplnou pivotací na Hessenbergovu matici.

Naprogramování těchto algoritmů jsme věnovali velkou pozornost, protože se jedná o nejdůležitější část celého programu z hlediska numerické přesnosti analýzy. Přestože uvedený postup je dnes klasický a všeobecně známý, úroveň jeho praktického naprogramování je překvapivě různá. Například v [7] je uváděn soubor procedur, které

vykazují numerickou nestabilitu již při hledání dvojnásobného vlastního čísla. Nakonec jsme za základ našeho řešení zvolili opět fortranovské procedury ze simulátoru C.I.A., které po malých modifikacích poskytují velmi přesné výsledky i při analýze obvodů s imitancemi vysokých řádů.

Výpis jednotlivých procedur je uveden v Příloze 2.

SYMBOLICKÁ ANALÝZA ◊ SEMISYMBOLICKÁ ANALÝZA

Tato procedura je snadná - za symbolické výrazy v koeficientech obvodových funkcí dosadíme jejich numerické ekvivalenty. Následuje úprava vzniklé racionální lomené funkce na unifikovaný tvar vytknutím koeficientů u nejvyšších mocnin v čitateli a jmenovateli a jejich sdružení do jediné multiplikativní konstanty.

Připravovaná verze programu bude umožňovat i tzv. polysymbolickou analýzu, kdy jen některé symbolické výrazy mohou být nahrazeny čísly.

NULOVÉ BODY A PÓLY → KOEFICIENTY OBVODOVÉ FUNKCE

Transformace nulových bodů, resp. pólů v koeficienty obvodové funkce je snadno realizovatelná postupným roznásobováním dílčích kořenových činitelů. Opačná úloha je však numericky velmi náročná.

KOEFICIENTY OBVODOVÉ FUNKCE → NULOVÉ BODY A PÓLY

Jde o matematickou úlohu hledání všech kořenů polynomu n -tého řádu. Naše aplikace vyžaduje značné nároky na numerickou přesnost, protože použití imitancí vyšších řádů může vést k násobným nulovým bodům, resp. pólům obvodových funkcí (zejména imitančních). Naprogramování procedury pro bezchybné vyhledání n -násobného kořenu pro $n > 5$ je nesnadnou úlohou. Selžou i mnohé metody převádějící problém hledání kořenů na problém hledání vlastních čísel matice přidružené k zadanému polynomu. Testovacímu zadání

$$\sum_{i=0}^{20} a_i p^i = 0,$$

$$a_0 = a_{20} = 1$$

$$a_1 = a_{19} = 20$$

$$a_2 = a_{18} = 190$$

$$a_3 = a_{17} = 1140$$

$$a_4 = a_{16} = 4845$$

$$a_5 = a_{15} = 15504$$

$$a_6 = a_{14} = 38760$$

$$a_7 = a_{13} = 77520$$

$$a_8 = a_{12} = 125970$$

$$a_9 = a_{11} = 167960$$

$$a_{10} = 184756$$

s dvacetinásobným kořenem -1 nakonec vyhověla procedura, kterou jsme naprogramovali na základě tzv. Laguerrovy metody hledání kořenů [7] doplněné technikou „root polishing“. Všechny 20 kořenů je nalezeno naprosto přesně.

Nicméně se v budoucnu pokusíme o hledání ještě spolehlivějších metod, neboť popisovaná procedura je důležitou součástí tzv. symbolického průchodu programem (viz obr.1) a na její přesnosti pak závisí výsledky jak kmitočtové, tak i časové analýzy.

Výpis vyvinutých procedur spolu s testovacím programem je v Příloze 3.

KMITOČTOVÁ ANALÝZA

V obou standardních průchodech programem se předpokládá kmitočtová analýza založená na reprezentaci obvodové funkce pomocí m nulových bodů a n pólů

$$K(p) = K_0 \frac{\prod_{i=1}^m (p - n_i)}{\prod_{i=1}^n (p - p_i)} \quad (8)$$

Modulová a argumentová kmitočtová charakteristika jsou určovány skládáním dílčích charakteristik jednotlivých kořenových činitelů. Známým způsobem lze z nulových bodů a pólů stanovit i průběh skupinového zpoždění.

V programu je možné zvolit i výpočet kmitočtových charakteristik z koeficientů obvodové funkce, která je nyní ve tvaru

$$K(p) = K_0 \frac{\sum_{i=0}^m a_i p^i}{\sum_{i=0}^n b_i p^i} \quad (9)$$

Tento postup můžeme zvolit v případě, nemáme-li důvěru v numerickou přesnost nulových bodů a pólů a chceme-li využít přesných hodnot koeficientů získaných ze symbolické analýzy. Výpočet skupinového zpoždění je však nyní prováděn numerickou derivací fázové charakteristiky a je tudíž méně přesný než v případě metody vycházející z nulových bodů a pólů.

Nabídka výpočtu skupinového zpoždění je aktivní pouze u obvodové funkce typu přenos napětí nebo proudu. Při výpočtu imitancních poměrů lze určovat i reálnou a imaginární část obvodové funkce.

SEMISYMBOLICKÁ ČASOVÁ ANALÝZA

Vyvinutý algoritmus semisymbolické analýzy časových průběhů je použitelný pouze k analýze vynucených odezev, tj. odezev při nulových počátečních podmínkách. Vychází ze znalosti nulových bodů a pólů obvodové funkce a probíhá v těchto krocích:

1. Obvodová funkce se vynásobí Laplaceovým obrazem budicího signálu. Získáme Laplaceův obraz vynucené odezvy.
2. Proveďte se rozklad Laplaceova obrazu vynucené odezvy na parciální zlomky.
3. Každému parciálnímu zlomku se přiřadí příslušný originál odezvy.
4. Proveďte se konečná úprava vzorce na reálnou funkci času.

Programátorsky nejnáročnější je krok 2. Za základ našeho algoritmu jsme zvolili řešení uvedené v [8]. V našem řešení jsme se vyvarovali chyb z původního pramene a výsledkem je rychlý a spolehlivý algoritmus semisymbolické analýzy, který je schopen identifikovat i zobecněné impulsy v časových průbězích. Prozatímni knihovnu budicích signálů, která je omezena na jednotkový impuls, jednotkový skok a obecný harmonický signál, v budoucnu rozšíříme na úroveň knihovny používané u numerické časové analýzy.

V současné době pracujeme na algoritmu semisymbolické časové analýzy periodických ustálených dějů.

NUMERICKÁ ČASOVÁ ANALÝZA

Jak ukazuje obr.1, numerická analýza může probíhat ve dvou různých režimech:

1. Výpočet vynucených odezev na omezenou škálu typizovaných budicích signálů s využitím semisymbolické analýzy (tj. výpočet bodů křivky z vzorce).
2. Analýza na čistě numerickém principu s těmito rozšířenými možnostmi:
 - množina budicích signálů je rozšířena o zdroj lichoběžníkových impulsů; volbou parametrů signálu je možné modelovat i obdélníkové a pilovité kmity; dále je možné budit analyzovaný obvod uživatelským signálem, jehož vzorky jsou uloženy v datovém souboru o předepsaném formátu.
 - kromě vynucené odezvy je možno přímo určit i periodický ustálený stav za předpokladu, že zadaný segment vstupního signálu je jednou periodou jeho periodického pokračování.

Numerickými metodami časové analýzy lineárních systémů jsme se zabývali již v předchozích letech [1], [9], [10]. V těchto pracech jsme popsali různá zdokonalení Vlachovy - Singhalovy (V - S) metody numerické zpětné Laplaceovy transformace [11]. K zvýšení přesnosti analýzy pro delší simulační časy jsme použili tzv. „*resetting principle*“ [11]. Metoda, kterou nyní popíšeme, je založena na totéž principu, algoritmus výpočtu je však jednodušší než u V - S metody. Tím jsme dosáhli urychlení výpočtů při zachování vysoké přesnosti s minimálními nároky na paměť. Metoda poskytuje správné výsledky nezávisle na rozložení pólů Laplaceova obrazu.

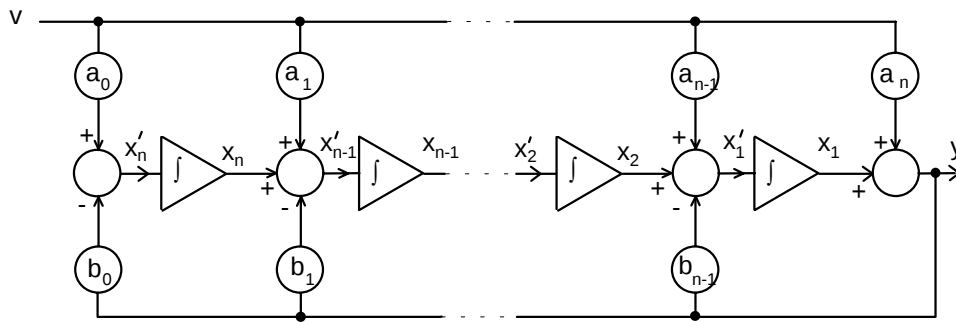
Předpokládejme, že Laplaceův obraz signálu je současně přenosovou funkcí určitého lineárního systému. Pak časový originál příslušející tomuto obrazu je impulsní charakteristikou systému. Označme vstup a výstup systému po řadě symboly $v(t)$ a $y(t)$ a jejich Laplaceovy obrazy $V(p)$ a $Y(p)$. Uvažujme přenosovou funkci ve tvaru

$$K(p) = \frac{Y(p)}{V(p)} = \frac{\sum_{i=0}^n a_i p^i}{p^n + \sum_{i=0}^{n-1} b_i p^i}, n \geq 1. \quad (10)$$

Pokud by řád čitatele byl větší než u jmenovatele, je možno jednoduchou úpravou přenosovou funkci rozdělit na dvě části, z nichž první bude obsahovat Laplaceův obraz zobecněných impulsů a druhá část bude ve standardním tvaru (10).

Známe koeficienty přenosové funkce a hledáme časový průběh $y(t)$ za předpokladu, že systém je ze svého nulového výchozího stavu vybuzen Diracovým impulsem.

K přenosové funkci (10) lze přiřadit nekonečně mnoho systémů n -tého řádu s různou vnitřní strukturou. Pro účely našeho řešení vybereme klasickou 1. kanonickou strukturu podle obr.4.



Obr. 4. 1. kanonická struktura dynamického systému n -tého řádu s přenosovou funkcí (10).

Zvolíme-li výstupní signály integrátorů za stavové veličiny, stavové rovnice 1. kanonické struktury budou mít známý tvar

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \underbrace{\begin{bmatrix} -b_{n-1} & 1 & 0 & \cdots & 0 \\ -b_{n-2} & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -b_1 & 0 & 0 & \cdots & 1 \\ -b_0 & 0 & 0 & \cdots & 0 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \underbrace{\begin{bmatrix} a_{n-1} - a_n b_{n-1} \\ a_{n-2} - a_n b_{n-2} \\ \vdots \\ a_1 - a_n b_1 \\ a_0 - a_n b_0 \end{bmatrix}}_{\mathbf{B}} v, y = x_1 + a_n v. \quad (11)$$

Podstata výpočtu časové odezvy z Laplaceova obrazu je následující:

- 1) Problém inverze Laplaceova obrazu (10) je převeden na problém nalezení impulsní charakteristiky systému z obr.4.
- 2) Impulsní charakteristika se určí řešením stavových rovnic (11).

Tohoto postupu je využito i v metodě V - S . Náplň obou uvedených kroků však bude v našem případě jiná. V metodě V - S se pracuje se systémem v 2. kanonické struktuře, protože je to výhodné pro realizaci algoritmu inverzní Laplaceovy transformace v bodu 2. My však vyřešíme stavové rovnice jiným způsobem, pro nějž bude výhodnější tvar stavových rovnic představovat 1. kanonická struktura.

Stavové rovnice (11) mají klasické řešení

$$\mathbf{x}(t) = e^{\mathbf{A}t} \mathbf{B} * v(t) + e^{\mathbf{A}t} \mathbf{x}(0), y(t) = x_1(t) + a_n v(t), \quad (12)$$

kde symbol $*$ značí konvoluční součin.

Uvažujeme-li výpočet impulsní charakteristiky, tj. $v(t) = \delta(t)$, $\mathbf{x}(0) = \mathbf{0}$, pak s využitím „resetting principle“ lze vektor $\mathbf{x}(t)$ vyčíslit v jednotlivých bodech časové osy $t_1 < t_2 < \dots < t_k < \dots$ s ekvidistantním krokem Δt takto:

$$\begin{aligned} \mathbf{x}(t_1) &= e^{\mathbf{A}\Delta t} \mathbf{B} && \text{; počátek časové osy v nule; } v(t) = \delta(t); \mathbf{x}(0) = \mathbf{0} \\ \mathbf{x}(t_k) &= e^{\mathbf{A}\Delta t} \mathbf{x}(t_{k-1}), k > 1 && \text{; počátek časové osy v } t_{k-1}; v(t) = 0; \mathbf{x}(0) \rightarrow \mathbf{x}(t_{k-1}) \end{aligned}$$

Pak tedy stačí jednou provždy vypočítat matici $e^{\mathbf{A}\Delta t}$ a stavový vektor v k -tém kroku počítat rekurentně z vektoru v předešlém kroku. Hledaná odezva v daném kroku je pak přímo 1. souřadnice stavového vektoru (viz rovnice (11)):

$$y(t_k) = x_1(t_k), k > 0.$$

Zbývá určit odezvu v počátku $y(0)$. Z obr.4 vyplývá, že v případě $a_n \neq 0$ pronikne vstupní Diracův impuls na výstup s vahou a_n . Dále nás zajímá limitní hodnota $y(0^+)$. Jednoduchý výpočet z přenosové funkce vede k výsledku

$$y(0^+) = a_{n-1} - a_n b_{n-1}$$

Algoritmus výpočtu originálu k Laplaceovu obrazu (10) v ekvidistantních bodech lze tedy zapsat následovně:

```

begin
ZADAVANI;           { zadávání vektorů koeficientů CIT a JME čitatele a
                      jmenovatele přenosové funkce a řádu N; zadávání kroku
                      DELTA a počtu kroků KR}

TVORBA_MATIC;       { sestavení matice A a vektoru B z koeficientů přenosové
                      funkce}

EXPMAT(A,DELTA,);    { výpočet matice exp(A.DELTA), výsledek uložen do A}
if (cit[N] >> 0) then
    writeln('v počátku je Diracův impuls o mohutnosti',cit[N]);
Y[0]:= cit[N-1]-cit[N]*jme[N-1];
X:=B;                { inicializace vektoru X}
for k:= 1 to KR do
    begin
        MATVEKMULTI(A,X);      { násobení matice A vektorem X;

```

výsledek se uloží do vektoru $\mathbf{X}$

$Y[k] := X[1];$

end;

end.

Algoritmus tvorby matice $\mathbf{A}$ a vektoru $\mathbf{B}$ je zřejmý z rovnice (11). K časově nejnáročnější proceduře patří výpočet matice $e^{\mathbf{A}\Delta t}$. V případě konstantního kroku je tato procedura volána jen jednou. K osvědčeným způsobům výpočtu patří sčítání konečného počtu členů Taylorovy řady exponenciální maticové funkce:

$$e^{\mathbf{A}\Delta t} \approx \sum_{k=0}^M \frac{(\mathbf{A}\Delta t)^k}{k!} \quad (13)$$

Sčítání jednotlivých členů je naprogramováno v cyklu. Po každém sečtení je prováděn test splnění chybového kritéria: relativní změna norem výsledných matic před a po sečtení musí poklesnout pod předem stanovenou mez.

Využijeme-li speciálního tvaru matice $\mathbf{A}$, který je dán použitou 1. kanonickou formou, můžeme výpočet maticové funkce (13) podstatně urychlit. Matici $\mathbf{A}$ rozložíme na součet jiných matic $\mathbf{A}_1$ a $\mathbf{A}_2$:

$$\begin{array}{c} \begin{array}{ccccc} -b_{n-1} & 1 & 0 & \cdots & 0 \\ -b_{n-2} & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -b_1 & 0 & 0 & \cdots & 1 \\ -b_0 & 0 & 0 & \cdots & 0 \end{array} \\ \mathbf{A} \end{array} = \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{array} \\ \mathbf{A}_1 \end{array} + \begin{array}{c} \begin{array}{ccccc} -b_{n-1} & 0 & 0 & \cdots & 0 \\ -b_{n-2} & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -b_1 & 0 & 0 & \cdots & 0 \\ -b_0 & 0 & 0 & \cdots & 0 \end{array} \\ \mathbf{A}_2 \end{array}. \quad (14)$$

K výpočtu k -té mocniny matice $\mathbf{A}\Delta t$ využijeme toho, že $k-1$. mocnina je vypočtena v předchozím cyklu a může tedy být k dispozici v paměti:

$$\frac{(\mathbf{A}\Delta t)^k}{k!} = \frac{(\mathbf{A}\Delta t)^{k-1}}{(k-1)!} \frac{\mathbf{A}\Delta t}{k} = \mathbf{M}_{k-1} \frac{\mathbf{A}\Delta t}{k}, \quad (15)$$

kde symbol $\mathbf{M}_{k-1}$ označuje matici uloženou v paměti. S využitím (14) můžeme psát

$$\frac{(\mathbf{A}\Delta t)^k}{k!} = \mathbf{M}_{k-1} (\mathbf{A}_1 + \mathbf{A}_2) \frac{\Delta t}{k} = (\mathbf{M}_{k-1} \mathbf{A}_1 + \mathbf{M}_{k-1} \mathbf{A}_2) \frac{\Delta t}{k}. \quad (16)$$

Vynásobením matic $\mathbf{M}_{k-1}$ a $\mathbf{A}_1$ vznikne matice, jejíž sloupce jsou sloupci původní matice $\mathbf{M}_{k-1}$, posunuté o jednu pozici doprava. První sloupec výsledné matice bude obsahovat samé nuly.

Vynásobením matic $\mathbf{M}_{k-1}$ a $\mathbf{A}_2$ vznikne matice, jejíž všechny sloupce kromě prvního jsou nulové. Člen v prvním sloupci a m -tém řádku je dán součtem

$$\sum_{i=1}^n \mathbf{M}_{k-1}[m, i] \mathbf{A}_2[i, 1] = - \sum_{i=1}^n \mathbf{M}_{k-1}[m, i] b_{N-i}. \quad (17)$$

Nyní můžeme formulovat algoritmus výpočtu k -tého členu Taylorova rozvoje (13):

Posunutí sloupců č.1 až $n-1$ matice $\mathbf{M}_{k-1}$ o jednu pozici doprava.

Tvorba 1. sloupce podle (17).

Násobení všech členů výrazem $\Delta t / k$.

Klasické násobení dvou matic lze uskutečnit pomocí n^3 násobení a n^3 sčítání. Naše procedura vyžaduje přesun $n-1$ sloupců a n^2 násobení a n^2 sčítání. Jestliže deklarujeme matici jako pole sloupců, pak posun sloupců lze realizovat jejich prostým přeadresováním.

Algoritmus byl testován s cílem srovnání jeho přesnosti a rychlosti s metodou V - S . Testování proběhlo na příkladech obrazů s vícenásobnými póly a na obrazech periodických, konkrétně harmonických signálů. Výsledky testování ukazuje Tab.1. Kromě popisované metody byly testovány tři varianty metody V - S , lišící se počtem pólů *Padéovy aproximační funkce* (viz [11]). Při testu byla prováděna náročná inverze obrazu

$$\frac{p}{p^2 + (2\pi \cdot 1000)^2} \mp \cos(2\pi \cdot 1000t),$$

kteřá probíhala s krokem 0,1ms celkem v 10000 krocích. To představovalo výpočet originálu přes jeho 1000 period. Z tabulky je zřejmé, že nový algoritmus tuto úlohu zvládl s řádově menší chybou než u nejpřesnější metody V - S , a to sedmkrát rychleji. Nejrychlejší V - S metoda s dvěma póly *Padéovy aproximační funkce* byla třikrát pomalejší, ovšem výsledky výpočtu jsou nepoužitelné v důsledku neúnosných numerických chyb.

metoda	průměrná relativní chyba v % správného údaje	čas výpočtu [s] na PC AT 486DX2/66
prezentovaná	10^{-7}	0,5
V-S, 10 pólů	10^{-6}	3,73
V-S, 4 póly	1	2,14
V-S, 2 póly	100 *)	1,59

*) Nepoužitelné výsledky v důsledku numerického selhání metody

Tab.1. Výsledky testování.

V programu je vyřešen i problém okamžitého nalezení periodické odezvy a výpočet odezvy na uživatelský signál. Popis těchto metod jsme uvedli v [1].

ZÁVĚR

V zprávě jsou stručně popsány algoritmy, které jsme vyvinuli pro počítačovou simulaci obvodů obsahujících imitance vyšších řádů. Tyto algoritmy jsou již úspěšně odladěny a v současné době dochází k jejich připojování k dříve realizovaným symbolickým algoritmům programu COCO. Zejména algoritmy semisymbolické analýzy již byly využity k ověřování principu nově syntetizovaných filtrů s imitancemi vyšších řádů.