

# SIMULACE ELEKTRICKÝCH A ELEKTRONICKÝCH OBVODŮ V JAZYKU MATLAB

*Juraj Valsa*

*Ústav teoretické a experimentální elektrotechniky,  
Fakulta elektrotechniky a informatiky VUT v Brně*

## Úvod

Pro simulaci elektrických a elektronických obvodů existuje dnes celá řada velmi dobrých univerzálních programů, které se vzájemně liší pouze v detailech. Jejich nevýhodou je, že jsou málo průhledné, takže uživatel obvykle musí věřit výsledkům a nemá mnoho možností si je ověřit. Často je užitečné vytvořit si alespoň pro jednodušší obvody vlastní simulační program, u kterého lze celý postup výpočtu detailně sledovat.

Matematický jazyk MATLAB se k tomuto účelu výborně hodí, protože usnadňuje celou řadu složitých matematických postupů, kterých se při simulaci obvodů používá. To, že pracuje s vektory a maticemi, jej činí v tomto ohledu neocenitelným.

V příspěvku budeme sledovat algoritmy použitelné v následujících oblastech:

- výpočet harmonického ustáleného stavu v lineárním (resp. linearizovaném) obvodu symbolickým (komplexním) počtem
- automatizovaná formulace rovnic linearizovaného obvodu v maticovém tvaru
- výpočet pólů a nulových bodů
- řešení odezev v kmitočtové i časové oblasti
- řešení přechodných dějů v nelineárních obvodech.

## Harmonický ustálený stav v lineárních obvodech

Simulace harmonického ustáleného stavu v lineárních obvodech vychází z předpokladu, že obvod je napájen harmonickým signálem s kmitočtem  $\omega$ . Sledujeme stav v obvodu po odeznění přechodných dějů. Všechna napětí a všechny proudy v obvodu jsou harmonické, mají stejný kmitočet  $\omega$  a vzájemně se liší pouze amplitudami a fázemi. Řešení je založeno na tzv. symbolickém počtu. Harmonické průběhy napětí a proudů jsou popsány pomocí komplexních čísel, tzv. fázorů.

Např. průběh okamžité hodnoty napětí

$$u(t) = U_m \sin(\omega t + \varphi_u)$$

je zobrazen komplexním číslem (tzv. fázorem absolutní hodnoty)

$$\hat{U}_m = U_m e^{j\varphi_u} = U_m \cos \varphi_u + j U_m \sin \varphi_u = U_m \angle \varphi_u = a + jb.$$

Při praktických výpočtech se často používá fázoru efektivní hodnoty

$$\hat{U} = \frac{1}{\sqrt{2}} \hat{U}_m = U e^{j\varphi_u}, \text{ kde } U = \frac{1}{\sqrt{2}} U_m \text{ je efektivní hodnota napětí.}$$

Polární tvar čísla

$$U e^{j\varphi_u}$$

nese informaci o amplitudě resp. efektivní hodnotě a fázi. Jednotlivé složky kartézského (složkového) tvaru  $a+jb$  jsou pak průmětem fázoru do reálné resp. imaginární osy v rovině komplexních čísel.

Analogickým způsobem je definován fázor maximální nebo efektivní hodnoty proudu.

Pro řešení obvodů využíváme obou Kirchhoffových zákonů a to pro součet proudů v uzlu

$$\sum \hat{I} = 0, \text{ pro součet napětí v uzavřené smyčce } \sum \hat{U} = 0$$

a Ohmova zákona, vyjadřujícího vztah mezi napětím a proudem větve ve tvaru

$$\hat{I} = \frac{\hat{U}}{\hat{Z}} = \hat{U}\hat{Y}.$$

Zde

$\hat{Z}$  je impedance,  $\hat{Y}$  je admitance větve.

Impedance i admitance jsou opět komplexní čísla, která mohou být vyjádřena jak v polárním tak i v kartézském tvaru.

Impedance rezistoru s odporem  $R$  je reálné číslo  $R$ , impedance kondenzátoru imaginární číslo

$$\frac{1}{j\omega C} = \frac{-j}{\omega C}, \text{ impedance cívky imaginární číslo } j\omega L.$$

Admitance rezistoru je reálná,  $G = \frac{1}{R}$ , admitance kondenzátoru a cívky imaginární  $j\omega C$

$$\text{resp. } \frac{1}{j\omega L} = \frac{-j}{\omega L}.$$

Pro spojování impedancí do série a paralelně platí analogické vzorce jako pro spojování rezistorů; matematické operace však musejí být provedeny s ohledem na komplexní tvar čísel.

Protože MATLAB pracuje výhradně s komplexními čísly v kartézském (složkovém) tvaru a při simulaci obvodu je často třeba převádět čísla z jednoho tvaru do druhého, použijeme připravených M-funkcí `cpolar.m` a `ccart.m`.

```
% ccart - kartézský tvar komplexní matice zadané v polárním tvaru
% argument argdeg se zadává ve stupních
function A = ccart(modul,argdeg)
arg=pi/180*argdeg;
A=modul.*(cos(arg)+i*sin(arg));
```

(1)

Příkazem

$$U=ccart(230,-30)$$

vytvoříme fázor efektivní hodnoty napětí 230V s počáteční fází  $-30$  stupňů, tj. komplexní číslo  $199,1858-115i$ .

```
% cpolar - vytvoř polární tvar komplexní matice A
% výstupem je struktura polarA
% její prvky jsou matice stejného rozměru jako A
% polarA.mod je matice modulů jednotlivých prvků matice A
% polarA.deg jsou argumenty ve stupních
% polarA.rad jsou argumenty v radiánech
function polarA = cpolar(A)
a=real(A); b=imag(A);
modul=sqrt(a.^2+b.^2);
argrad=atan2(b,a);
argdeg=180/pi*argrad;
polarA=struct('mod',modul,'deg',argdeg,'rad',argrad);
```

(2)

Je-li např.

$$Ii=-5.6+0.34i,$$

pak příkazem

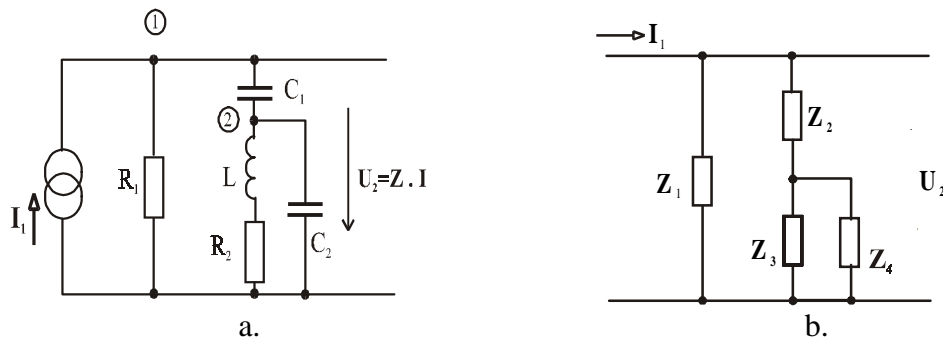
$$I=cpolar(Ii)$$

získáme

$$I.\text{mod}= 5.6103, I.\text{deg}= 176.5256, I.\text{rad}= 3.0810.$$

### Příklad 1

Přenosovou impedanci  $Z_t(j\omega) = \frac{\hat{U}_2}{\hat{I}_1}$  obvodu na obr.1 vypočítáme jako



Obr.1. Obvod řešený v příkladu 1. a. celkové schéma, b. impedance

$$\hat{Z}_t = \frac{\hat{Z}_1(\hat{Z}_2 + \frac{\hat{Z}_3\hat{Z}_4}{\hat{Z}_3 + \hat{Z}_4})}{\hat{Z}_1 + \hat{Z}_2 + \frac{\hat{Z}_3\hat{Z}_4}{\hat{Z}_3 + \hat{Z}_4}} = \frac{\hat{Z}_1\hat{Z}}{\hat{Z}_1 + \hat{Z}}$$

Obvod řešíme v rozsahu úhlových kmitočtů  $\omega_{\min}=0.01\text{s}^{-1}$ ,  $\omega_{\max}=10\text{s}^{-1}$  (3 dekády, celkem 181 bodů).

Program RLC1.m počítá a zobrazí kmitočtovou závislost modulu impedance  $Z_t$  v decibelech (tzv. Bodeho charakteristiku) a kmitočtovou závislost argumentu  $\varphi$  ve stupních.

Dále zobrazí hodograf této závislosti (imaginární část přenosu jako funkci reálné části v komplexní rovině).

Přibližnou numerickou derivací pak program počítá sklon modulové charakteristiky v decibelech na dekádu kmitočtu

$$s_a(\omega) = \frac{d(20\log_{10} K_u)}{d[\log_{10}(\omega)]} \quad (3)$$

a záporně vzaty sklon fázové charakteristiky, tzv. charakteristiku skupinového zpoždění

$$t_g(\omega) = -\frac{d\varphi(\omega)}{d\omega} \quad (4)$$

*% RLC1 - řešení ustáleného harmonického stavu v obvodu na obr.1* (5)

*% symbolickým počtem*

*om=logspace(log10(0.01),log10(10),181); % variace kmitočtu*  
*Z1=10; Z2=1./(2\*om\*i); Z3=2+4\*om\*i; Z4=1./(0.1\*om\*i); % přenosová impedance*  
*Z=Z2+Z3.\*Z4./(Z3+Z4);*

*Ku=Z./(Z1+Z); % přenosová impedance v kartézském tvaru*

*Kuu=cpolar(Ku); % v polárním tvaru*

*KudB=20\*log10(Kuu.mod); % modul v decibelech*

*dom=diff(om); % tabulka diferencí omega*

```

om1=om(1:length(om)-1);           % zkrácená tabulka omega
tg=-diff(Kuu.rad)./dom;           % skupinové zpoždění
sa=log(10)*om1.*diff(KudB)./dom;  % sklon modulové char. v dB/dek
%
figure(1)
subplot(2,1,1)
semilogx(om,KudB), grid on        % modulová charakteristika
title('modulova a fazova charakteristika') % v decibelech
xlabel('omega [s^-1]'), ylabel('Ku [dB]')
subplot(2,1,2)
semilogx(om,Kuu.deg), grid on     % argumentová char.
xlabel('omega [s^-1]'), ylabel('fi [deg]') % ve stupních
%
figure(2)
subplot(2,1,1)
semilogx(om1,sa), grid on         % sklon modulové char.
title('sklon modulove a fazove charakteristiky')
xlabel('omega [s^-1]'), ylabel('sa [dB/dek]')
subplot(2,1,2)
semilogx(om1,tg), grid on        % char. skupinového zpoždění
xlabel('omega [s^-1]'), ylabel('tg [s]')
%
figure(3)
plot(real(Ku),imag(Ku)), grid on  % hodograf
title('hodograf')
xlabel('Re(Ku)'), ylabel('Im(Ku)')

```

## **Automatizovaná formulace rovnic lineárního obvodu v maticovém tvaru a následné řešení odezev v kmitočtové i časové oblasti**

V této kapitole budou uvedeny příkazy jazyka Matlab, jimiž lze provést následující operace při simulaci lineárních obvodů:

- sestavení matic  $G$  a  $C$  na základě modifikované metody uzlových napětí
- výpočet komplexní hodnoty zvolené obvodové funkce v harmonickém ustáleném stavu
- výpočet citlivosti této funkce na změnu parametru některého obvodového prvku
- výpočet pólů resp. nulových bodů obvodové funkce (semisymbolická analýza)
- výpočet impulsové odezvy  $g(t)$  a přechodné odezvy  $h(t)$  numerickou inverzí Laplaceova obrazu
- výpočet odezvy obvodu na libovolný vstupní signál konvolucí časového průběhu na vstupu s impulsovou odezvou (váhovou funkcí)  $g(t)$ .

### **Sestavení matic obvodu**

Modifikovaná metoda uzlových napětí vede na obvodové rovnice ve tvaru

$$(\mathbf{G} + s\mathbf{C})\mathbf{X} = \mathbf{B}, \quad (6)$$

kde  $\mathbf{G}$  a  $\mathbf{C}$  jsou čtvercové matice  $n \times n$ ,  $s$  je obecně komplexní proměnná a  $\mathbf{B}$  je  $n$ -vektor pravých stran.

Vektor  $\mathbf{X}$  obsahuje neznámá uzlová napětí a některé přidané proudy.

Při simulaci vycházíme z popisu obvodu, tzv. *netlistu*. Ten je sestaven uživatelem nebo je výstupem programu, umožňujícího zadání ve formě elektrického schématu. Každý obvodový prvek je v netlistu popsán jedním řádkem ve tvaru

*typ, n1, n2, n3, n4, par.*

U dvojpólů

*n3=n1, n4=n2,*

u ideálního operačního zesilovače

*par=0.*

Celkový počet nezávislých uzlů je *nu*, počet přidaných proudů *numi*, celkový počet neznámých  $n=nu+numi$ .

Pro sestavení matice **G** připravíme pomocné vektory *radg*, *slpg* a *parg*. Vektor *radg* obsahuje indexy řádků, vektor *slpg* indexy sloupců a vektor *parg* numerické hodnoty, které budou do odpovídajících prvků matice **G** přičteny. Analogicky připravíme vektory *radc*, *slpc*, *parc*, které použijeme při formulaci matice **C**.

Jednotlivé lineární prvky přispějí do příslušných vektorů takto:

- rezistor mezi uzly n1 a n2 s odporem R:

*radg=[n1 n1 n2 n2]; slpg=[n1 n2 n1 n2]; parg=[1/R -1/R -1/R 1/R];*

- zdroj proudu řízený napětím se vstupními svorkami n1-n2, výstupními svorkami n3-n4 a strmostí gm:

*radg=[n3 n3 n4 n4]; slpg=[n1 n2 n1 n2]; parg=[gm -gm -gm gm];*

- zdroj napětí řízený napětím se vstupními svorkami n1-n2, výstupními svorkami n3-n4 a zesílením A (index výstupního proudu zdroje i2):

*radg=[n3 n4 i2 i2 i2 i2]; slpg=[i2 i2 n1 n2 n3 n4]; parg=[1 -1 -A A 1 -1];*

- zdroj proudu řízený proudem se vstupními svorkami n1-n2, výstupními svorkami n3-n4 a zesílením B (index řídicího proudu zdroje i1):

*radg=[n1 n2 n3 n4 i1 i1]; slpg=[i1 i1 i1 i1 n1 n2]; parg=[1 -1 B -B 1 -1];*

- zdroj napětí řízený proudem se vstupními svorkami n1-n2, výstupními svorkami n3-n4 a přenosovým odporem W (index řídicího proudu i1, index výstupního proudu zdroje i2):

*radg=[n1 n2 n3 n4 i1 i1 i2 i2 i2]; slpg=[i1 i1 i2 i2 n1 n2 n3 n4 i1];*

*parg=[1 -1 1 -1 1 -1 1 -1 -W];*

- ideální operační zesilovač se vstupními svorkami n1-n2, výstupními svorkami n3-n4 a s nekonečně velikým zesílením (index výstupního proudu zesilovače i2):

*radg=[n3 n4 i2 i2]; slpg=[i2 i2 n1 n2]; parg=[1 -1 1 -1];*

Podobně připravíme tabulky *radc*, *slpc*, *parc*, do kterých uložíme údaje o zapojení a parametrech kondenzátorů a cívek.

- kondenzátor mezi uzly n1 a n2 s kapacitou C:

*radc=[n1 n1 n2 n2]; slpc=[n1 n2 n1 n2]; parc=[C -C -C C];*

- cívka mezi uzly n1 a n2 s indukčností L (index proudu cívkou i1):

*radg=[n1 n2 i1 i1]; slpg=[i1 i1 n1 n2]; parg=[1 -1 1 -1];*

*radc=[i1]; slpc=[i1]; parc=[-L];*

je-li s cívkou zapojen do série rezistor a nevyžadujeme výpočet napětí společného uzlu, budou mít ukládací tabulky tvar

*radg=[n1 n2 i1 i1 i1]; slpg=[i1 i1 n1 n2 i1]; parg=[1 -1 1 -1 -R];*

*radc=[i1]; slpc=[i1]; parc=[-L];*

- reálný operační zesilovač se vstupními svorkami n1-n2, výstupními svorkami n3-n4 a konečnou hodnotou součinu šířky pásma a zesílení GB (index výstupního proudu zesilovače i2):

*radg=[n3 n4 i1 i1]; slpg=[i1 i1 n1 n2]; parg=[1 -1 -GB GB];*

*radc=[i1 i1]; slpc=[n3 n4]; parc=[1 -1];*

Každou z matic **G** a **C** pak vytvoříme jediným příkazem

```
G=sparse(radg, slpg, parg);
C=sparse(radc, slpc, parc);
```

Matice jsou uloženy jako řídké (sparse) matice. Na jedné straně to znamená menší nároky na operační paměť při simulaci rozsáhlých obvodů, na druhé straně však je třeba při provádění některých operací brát tuto skutečnost v úvahu.

Podle připojení nezávislých zdrojů k obvodu resp. podle volby vstupní brány obvodu sestavíme ještě sloupcový vektor **B** pravých stran.

## Příklad 2

Sestavení matice obvodu na obr.1.

Obvod má 2 nezávislé uzly a dále obsahuje sériovou větev L-R2, jejíž proud zvolíme jako třetí neznámou obvodovou veličinu. Pak sestavíme matice **G** a **C** takto:

```
% ukládací vektory
radg=[]; slpg=[]; parg=[];          % nuluj
radc=[]; slpc=[]; parc=[];
% rezistor R1
radg=[radg 1]; slpg=[slpg 1]; parg=[parg 1/R1];
% induktor L v sérii s rezistorem R2
radg=[radg [2 3 3]]; slpg=[slpg [3 2 3]]; parg=[parg [1 1 -R2]];
radc=[radc 3]; slpc=[slpc 3]; parc=[parc -L];
% kondenzátory C1 a C2
radc=[radc [1 1 2 2]]; slpc=[slpc [1 2 1 2]];
parc=[parc [C1 -C1 -C1 C1+C2]];
% sestavení matic G a C
G=sparse(radg,slpg,parg); C=sparse(radc,slpc,parc);
```

(7)

## Kmitočtové charakteristiky v harmonickém ustáleném stavu

Definujeme kmitočty, pro které budeme počítat kmitočtovou závislost vektoru **X**

```
freq=logspace(log10(fmin), log10(fmax), nnf);
```

a v cyklu pro jednotlivé kmitočty počítáme

```
for k=1:nnf
    x(:,k)=(G+j*2*pi*omega(k)*C)\B;
end;
```

(8)

(Všimněte si operátoru „backslash“ \ !)

Modulovou a argumentovou kmitočtovou charakteristiku zobrazíme např. pomocí příkazů

```
subplot(2,1,1)          % horní graf (modul v dB)
semilogx(freq, 20*log10(abs(x(out,:))))
subplot(2,1,2)          % dolní graf (argument ve stupních)
semilogx(freq, 180/pi*angle(x(out,:)))
```

Index *out* označuje pořadové číslo veličiny ve vektoru výstupních hodnot, jejíž kmitočtovou závislost sledujeme.

## Výpočet citlivosti přenosové funkce na změnu parametru

Absolutní citlivost obvodové funkce *F* na změnu parametru obvodového prvku *q*

$$S(F, q) = \partial F / \partial q \quad (9)$$

můžeme vždy vyjádřit jako záporně vzatý součin dvou přenosových funkcí daného obvodu.

Nejjednodušší případ nastane, je-li  $F$  přenosová impedance  $Z_{2,1}$  a proměnný parametr je admitance  $Y_q$ . Pak

$$S(F, Y_q) = -Z_{3,1}Z_{2,3}, \quad (10)$$

kde  $Z_{3,1}$  je přenosová impedance ze vstupu na svorky admitance  $Y_q$  a  $Z_{2,3}$  přenosová impedance z těchto svorek na výstup.

Při výpočtu první přenosové impedance použijeme opět rovnice (8). Vektor pravých stran  $B_1$  obsahuje nuly a jedničku v řádku, odpovídajícím vstupnímu uzlu obvodu. Z vektoru výsledků určíme napětí na proměnné admitanci. Pro výpočet druhé přenosové impedance pak připravíme nový vektor pravých stran  $B_2$ , rovnici (8) znovu vyřešíme a odečteme napětí na výstupu.

V případě, že se zajímáme o citlivost jedné přenosové impedance na všechny admitance v obvodu, je lépe použít pro výpočet všech impedancí  $Z_{3,2}$  v jediném kroku tzv. *adjungovaného obvodu*. Ten má admitanční matici komplexně sdruženou k původní admitanční matici obvodu. Nejúčinnější postup je ten, který využívá LU dekompozice. V Matlabu jej lze realizovat např. takto:

$$\begin{aligned} [L, U] &= LU(G+j*2*pi*omega(k)*C); && \% LU dekompozice matice obvodu \\ x1(:,k) &= L \setminus (U \setminus B1); && \% \text{výpočet } Z_{1,3} \end{aligned}$$

Pro adjungovaný obvod pak

$$x2(:,k) = U' \setminus (L' \setminus B3); \quad \% \text{výpočet } Z_{3,2}$$

Vektor  $B_3$  má jedničku v řádku, odpovídajícím výstupnímu uzlu.

Hledaná citlivost na admitanci zapojenou mezi uzly  $m_1$  a  $m_2$  je pak na  $k$ -tém kmitočtu rovna

$$S = -(x1(m_1, k) - x1(m_2, k)) * (x2(m_1, k) - x2(m_2, k)); \quad (11)$$

Adjungovaný obvod můžeme použít i při výpočtu šumových poměrů v obvodu.

U jednodušších obvodů, popsaných maticemi nižších řádů, lze citlivosti počítat přímo z prvků inverzní matice bez nutnosti provádět LU faktorizaci a řešit soustavu rovnic se dvěma různými pravými stranami. Jsou-li vstupní uzly obvodu  $a, b$  a výstupní uzly  $c, d$ , pak citlivost přenosové impedance na strmost zdroje proudu řízeného napětím, jehož řídicí uzly jsou  $e, f$  a svorky řízeného zdroje  $g, h$ , vypočítáme jako

$$\begin{aligned} R &= \text{inv}(G+j*2*pi*omega(k)*C); && (12) \\ S &= -(R(c, g) - R(c, h) - R(d, g) + R(d, h)) * (R(e, a) - R(e, b) - R(f, a) + R(f, b)); \end{aligned}$$

V případě, že se jedná o citlivost na admitanci dvojpólu mezi uzly  $e, f$ , platí stejný vztah, pouze namísto indexu  $g$  píšeme index  $e$  a namísto  $h$  píšeme  $f$ .

Pomocí sumární citlivosti přenosové funkce na všechny kapacity a indukčnosti v obvodu můžeme dále nalézt přesné hodnoty sklonu modulové i fázové charakteristiky a vyhneme se tak chybám, způsobeným numerickým derivováním.

### Výpočet pólů obvodové funkce

Pro jednoduchost uvažujeme obvod, jehož matice  $C$  je nesingulární. Přitom kondenzátory nevytvářejí (ani s ideálními zdroji napětí) uzavřené smyčky a proudy cívbek jsou vzájemně nezávislé.

Pak lze vypočítat polohu pólů řešením zobecněné úlohy vlastních čísel jako

$$pol = -\text{eig}(\text{full}(G), \text{full}(C)) \quad (13)$$

Pro obvod na obr.1 získáme polohu pólů a nulových bodů pomocí příkazů

```
% poly a nulove body
pol=-eig(full(G),full(C))
zer=-eig(full(G(2:3,2:3)),full(C(2:3,2:3)))
```

(14)

Výsledné hodnoty jsou

```
pol =
-0.75389480074638 - 1.54693187615470i
-0.04221039850724 + 0.000000000000000i
-0.75389480074638 + 1.54693187615470i
zer =
-0.250000000000000 - 0.23779743280284i
-0.250000000000000 + 0.23779743280284i
```

Obvykle však matice **C** nemá plnou hodnotu a vektor *pol* obsahuje vedle správných pólů (jejichž počet je roven hodnotě *rank(C)* matice **C**) také hodnoty, které se od nich liší o mnoho dekadických řádů. Zvláště u tuhých soustav bývá pak obtížné rozhodnout, co skutečně platí. Proto je vhodné matici **C** (a současně i matici **G**) nejprve transformovat tak, abychom získali kanonickou soustavu s diagonální maticí **CC**, která má *rank(C)* nenulových prvků v levém horním rohu, a jinak samé nuly. Tato transformace je snadná, pokud napětí na kondenzátorech jsou vzájemně nezávislá. Pak příkaz

```
pol=-eig(full(GG),full(CC));
```

dá výsledný vektor *pol*, který na místech původně podezřelých hodnot obsahuje *Inf* a *NaN*. Těch se zbavíme jednoduchým příkazem

```
p=pol(~isnan(pol))
```

Jestliže kondenzátory vytvářejí v obvodu uzavřené smyčky a napětí na kondenzátorech jsou vzájemně závislá (případně proudy cívek jsou vzájemně závislé), musíme nejprve provést tzv. deflací matic. Protože příslušná funkce v Matlabu neexistuje, byla zvlášť vytvořena jako M-soubor. S touto funkcí je pak možno počítat nejen polohu pólů a nulových bodů, ale i velikost koeficientu  $a_n$  při nejvyšší mocnině proměnné  $s$  v čitateli i jmenovateli. Hodnotu tohoto koeficientu potřebujeme znát např. tehdy, když používáme nulové body a póly pro analýzu modulových kmitočtových charakteristik.

Ze získaných hodnot nulových bodů a pólů lze opět vypočítat přesné hodnoty sklonu modulové i fázové charakteristiky přenosu. Na kmitočtu  $\omega(k)$  pak vypočítáme skupinové zpoždění a sklon modulové charakteristiky jako

$$t_g(\omega) = \sum_{i=1}^m \frac{\operatorname{Re}(z_i)}{\operatorname{Re}^2(z_i) + [\operatorname{Im}(z_i) - \omega]^2} - \sum_{j=1}^n \frac{\operatorname{Re}(p_j)}{\operatorname{Re}^2(p_j) + [\operatorname{Im}(p_j) - \omega]^2}, \quad (15)$$

$$s_a(\omega) = -20\omega \left\{ \sum_{i=1}^m \frac{\operatorname{Im}(z_i) - \omega}{\operatorname{Re}^2(z_i) + [\operatorname{Im}(z_i) - \omega]^2} - \sum_{j=1}^n \frac{\operatorname{Im}(p_j) - \omega}{\operatorname{Re}^2(p_j) + [\operatorname{Im}(p_j) - \omega]^2} \right\}. \quad (16)$$

V Matlabu pak

```
tg=zeros(1,pocetf); % skupinové zpoždění
sa=zeros(1,pocetf); % sklon modulové charakteristiky
for mm=1:length(zer)
den=(real(zer(mm))^2+(imag(zer(mm))-om).^2);
tg=tg+real(zer(mm))./den;
```

(17)



```

    sa=sa+(imag(zer(mm))-om)./den;
end
for mm=1:length(pol)
    den=(real(pol(mm))^2+(imag(pol(mm))-om).^2);
    tg=tg-real(pol(mm))./den;
    sa=sa-(imag(pol(mm))-om)./den;
end
sa=-20*om.*sa;

```

## Výpočet impulsové odezvy $g(t)$ a odezvy na jednotkový skok $h(t)$ numerickou inverzí Laplaceova obrazu

Opakovanou aplikací algoritmu pro výpočet harmonického ustáleného stavu můžeme přibližně vypočítat časový průběh odezvy obvodu na jednotkový impuls (impulsovou odezvu  $g(t)$ ). Použijeme k tomu např. vzorce

$$f(t) = \frac{1}{t} \sum_{k=1}^m \operatorname{Re}[\beta_k F(\frac{\alpha_k}{t})] \quad (18)$$

Odezva je tak vyjádřena jako suma reálných částí součinu komplexních konstant  $\beta_k$  a hodnot obvodové funkce  $F$  určených na komplexních kmitočtech  $s = \alpha_k/t$ . Konstanty  $\beta$  a  $\alpha$  určíme podle toho, jak byla aproximována exponenciální funkce  $\exp(st)$  v nevlastním integrálu při inverzi Laplaceova obrazu. Použijeme-li např. Padého aproximace zlomkem s polynomem  $N$ -tého stupně v čitateli a  $M$ -tého stupně ve jmenovateli podle [2], pak máme  $m=5$  a při výpočtu odezvy pro jednu hodnotu času  $t$  musíme vyhodnotit obvodovou funkci  $F$  na 5 komplexních kmitočtech. Násobíme-li obraz obvodové funkce obrazem jednotkového skoku  $1/s$ , dostaneme obraz odezvy na skok. Příslušný program v Matlabu pak může vypadat např. takto:

```

% výpočet odezev g(t) a h(t)
alfa=[11.83009373916819+1.593753005885813j ...
    11.22085377939519+4.792964167565670j ...
    9.933383722175002+8.033106334266296j ...
    7.781146264464616+11.36889164904993j ...
    4.234522494797000+14.95704378128156j];
beta=[-16286.62368050479+139074.7115516051j ...
    +28178.11171305163-74357.58237274176j ...
    -14629.74025233142+19181.80818501836j ...
    +2870.418161032078-1674.109484084304j ...
    -132.1659412474876-17.47674798877164j];
tpoc=0; tkon=10; nnt=201; % interval řešení v časové oblasti
radt=linspace(tpoc,tkon,nnt);
for kk=2:nnt % cyklus pro čas t
    t=radt(kk);
    gt(kk)=0; ht(kk)=0;
    for mm=1:5 % komplexní kmitočet s
        s=alfa(mm)/t;
        mat=G+s*C;
        F=mat\B;
        H=F./s; % H(s)=F(s)/s
        gt(kk)=gt(kk)+real(beta(mm)*F(1))/t;
    end
end

```

```

    ht(kk)=ht(kk)+real(beta(mm)*H(1))/t;
end;
end
gt(1)=10.5; ht(1)=0;           % počáteční hodnoty

```

### Výpočet odezvy na libovolný vstupní průběh

Vycházíme z tabulky okamžitých hodnot vstupního signálu a impulsové odezvy  $g(t)$ . Vstupní signál musí být zadán v ekvidistantních bodech shodných s okamžiky, pro které jsme počítali  $g(t)$ . Pak získáme vektor okamžitých hodnot signálu na výstupu konvolucí.

Příklad:

Vstupní signál je jednorázový impuls  $\sin^2(\pi t/timp)$  o délce  $timp$

```

% tabulku času a impulsové odezvy g(t) přebírá                                     (20)
% z předchozích výsledků (19)
% vstupní signál (proud i1(t))
for ii=1:numt
    t=radt(ii);
    i1(ii)=(sin(pi*t/timp))^2*(t<=timp);
end;
u2=conv(i1,ft)*((tmax-tmin)/(numt-1));
plot(radt,u2(1:numt)), grid on, zoom on

```

Příkazem *conv* se vytvoří vektor dvojnásobné délky, avšak pouze prvních *numt* hodnot je významných. Proto musí být pro vykreslení grafu délka vektoru *u2* omezena.

Odezvu na libovolný vstupní průběh můžeme rovněž vypočítat integrací diferenciální rovnice obvodu. Je-li matice **C** nesingulární (tj. všechny rovnice jsou diferenciální), sestavíme příslušný ODE soubor a použijeme vhodné integrační funkce. Pro výpočet odezvy obvodu na obr.1 na jednotkový skok proudu na vstupu např. deklaruujeme matice **G** a **C** a vektor pravých stran **B** jako *global* a sestavíme

```

function dx=RLC1ode(t,x)
global G C B
dx=C\ (B-G*x);

```

(21)

Potom příkazem

```
[t,x]=ode45('RLC1ode',[0 10],[0 0 0]);
```

(22)

vypočítáme časovou závislost vektoru obvodových veličin  $x$ . Položka  $x(:,1)$  odpovídá napětí v uzlu č.1, tj. odezvě obvodu na jednotkový skok proudu.

### Simulace nelineárních elektronických obvodů

Poslední verze 5.3 Matlabu s doplňkem 11 umožňuje řešit i soustavy algebro-diferenciálních rovnic, s jakými se běžně setkáváme při simulaci elektronických obvodů. Příprava řešení je však náročnější než v případě lineárního obvodu s nesingulární maticí parametrů setrvačných prvků **C** (v materiálech programu Matlab je tato matice nazývána „mass matrix“).

Uvažujme obvod složený z lineárních prvků  $R, C, L$  a nelineárních nesetrvačných prvků (diod, tranzistorů,...). Diferenciální rovnice obvodu lze psát ve tvaru

$$\mathbf{G}x + \mathbf{C}x' + n(x) = b(t) \quad (23)$$

(apostrofem označujeme derivaci  $x$  podle času).

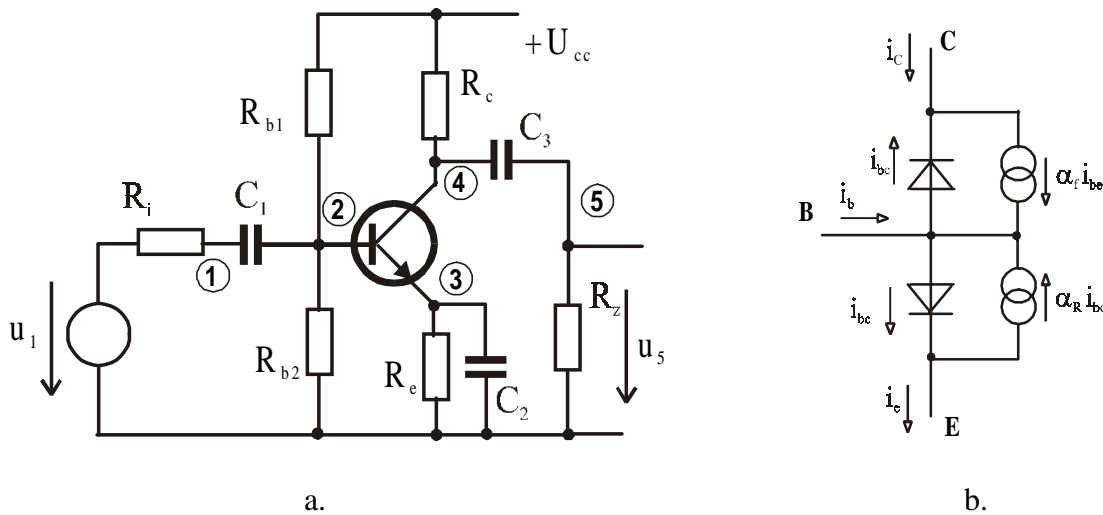
Vektor  $n(x)$  představuje nelineární funkci vektoru  $x$ . Úlohu převedeme na tvar

$$Cx' = b(t) - Gx - n(x), \quad (24)$$

vhodný ke zpracování pomocí solverů *ode15s* nebo *ode23t*.

### Příklad 3

Hledáme průběhy napětí v jednotlivých uzlech jednostupňového tranzistorového zesilovače na obr.3. Parametry obvodu jsou  $R_i = 1000\Omega$ ,  $R_{b1} = R_{b2} = 10k\Omega$ ,  $R_c = 5k\Omega$ ,  $R_e = 2k\Omega$ ,  $C_1 = 1\mu F$ ,  $C_2 = 20\mu F$ ,  $C_3 = 0.1\mu F$ ,  $U_{cc} = 6V$ .



Obr.2. Tranzistorový zesilovací stupeň z příkladu 3.  
a. Schéma, b. Ebers-Mollův model tranzistoru

Vstupní signál zesilovače je harmonické napětí s amplitudou 0.4V. Tranzistor jako nelineární prvek je popsán Ebers-Mollovým modelem, jehož diody mají charakteristiky dané funkcí  $i_{diode}$ . Proudoví zesilovací činitelé jsou  $\alpha_F = 0.99$ ,  $\alpha_R = 0.5$ . Schéma modelu je na obr.2b.

```
function varargout = amp4dae(t,u,flag)
% přechodný děj v jednostupňovém tranzistorovém zesilovači
% pro tranzistor je použit úplný Ebers-Mollův model
```

```
switch flag
```

```
case '' % Return du/dt = f(t,u).
```

```
varargout{1} = f(t,u);
```

```
case 'init' % Return default [tspan,u0,options].
```

```
[varargout{1:3}] = init;
```

```
case 'mass' % Return mass matrix M.
```

```
varargout{1} = mass(t,u);
```

```
end
```

```
% -----
```

```
function dudt = f(t,u)
```

```
Ucc = 6;
```

```
Rb1=10e3; Rb2=10e3; Rc=5e3; Re=2e3; Rz=10e3;
```

```
Uin = 0.4*sin(200*pi*t);
```

```
Ri = 1000;
```

```
% matice lineárních vodivostí
```

```
% napájecí napětí
```

```
% odpory v zesilovači
```

```
% vstupní signál
```

```
% vnitřní odpor zdroje signálu
```

```

G= [1/Ri 0 0 0 0
    0 1/Rb1+1/Rb2 0 0 0
    0 0 1/Re 0 0
    0 0 0 1/Rc 0
    0 0 0 0 1/Rz];
% vektor nelineárních řízených zdrojů proudu
af = 0.99; % parametry Ebers-Mollova modelu
ar = 0.5;
ibe=idie(u(2)-u(3),0.05); % proudy diod
ibc=idie(u(2)-u(4),0.05);
Nj=[0; (1-af)*ibe+(1-ar)*ibc; -ibe+ar*ibc; af*ibe-ibc; 0];
% vektor nezávislých zdrojů proudu
jj=[Uin/Ri; Ucc/Rb1; 0; Ucc/Rc; 0];
% vektor derivací
dudt=jj-G*u-Nj;
% -----
function [tspan,u0,options] = init
Ucc = 6;
tspan = [0 0.01];
u0 = [0; 0.4*Ucc; 0.4*Ucc; Ucc; 0];
%u0=zeros(5,1);
options = odeset('Mass','M','MassSingular','yes');
% -----
function C = mass(t,u) % matice setrvačných prvků
C1=1e-6; C2=20e-6; C3=0.1e-6;
C= [ C1 -C1 0 0 0
    -C1 C1 0 0 0
    0 0 C2 0 0
    0 0 0 C3 -C3
    0 0 0 -C3 C3];
% -----
% idie - výpočet proudu PN přechodem
% exponenciální charakteristika s lineárním prodloužením pro id>Ip
% u je napětí anoda-katoda, Ip je prahový proud
% id je anodový proud, Gd diferenciální vodivost,
% I0 proud zdroje v linearizovaném modelu
function [id,Gd,I0] = idie(u,Ip)
Is=1e-10; Ut=0.026;
Up=Ut*log(1+Ip/Is);
Gp=(Ip+Is)/Ut;
if u<=Up id=Is*(exp(u/Ut)-1); Gd=(id+Is)/Ut;
else
id=Ip+Gp*(u-Up); Gd=Gp;
end
I0=id-Gd*u;

```

Řešení vyvoláme příkazem

```
[t,u] = ode23t('amp4dae');
```

(26)

Vektor  $u$  obsahuje časové průběhy všech uzlových napětí v obvodu.

## Závěr

Jazykem Matlab lze velmi jednoduše realizovat celou řadu operací potřebných při simulaci elektrických obvodů. V článku byly popsány některé algoritmy použitelné pro lineární obvody. Aplikace v simulaci nelineárních obvodů jsou podstatně náročnější, protože sortiment nelineárních obvodových prvků je značně bohatý. Efektivní využití integračních funkcí jazyka Matlab bude vyžadovat formulaci Jacobiho matic pro každý takový prvek.

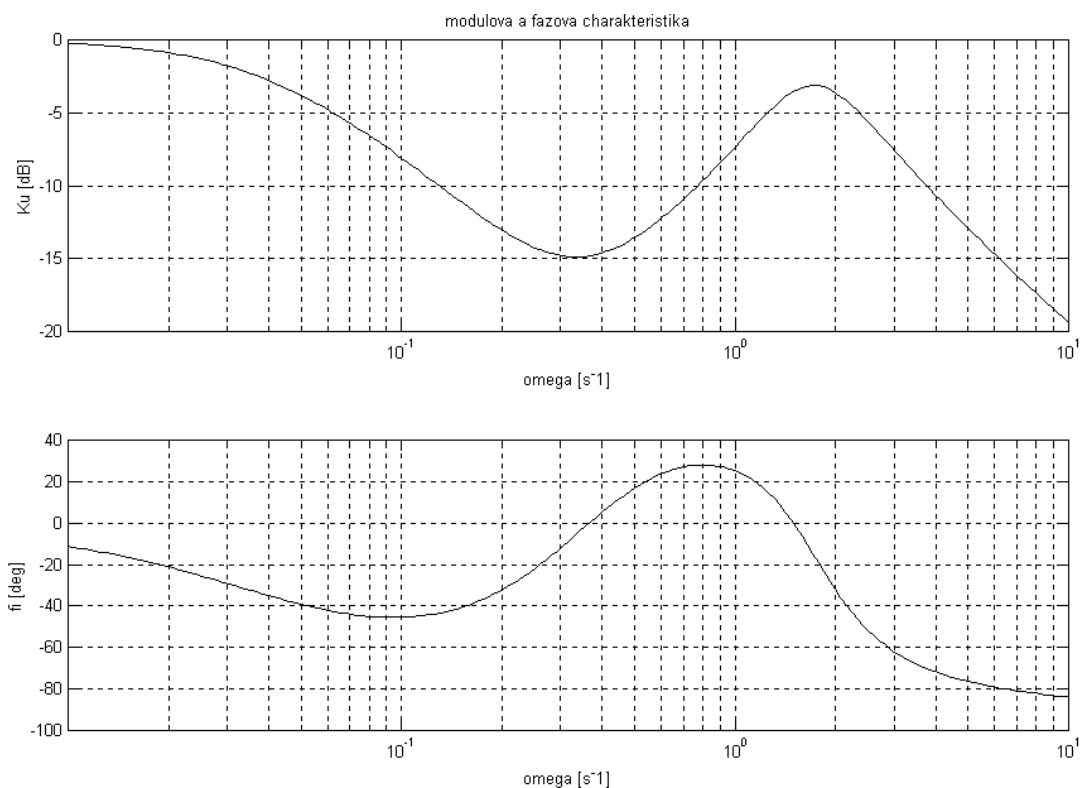
Je zajímavé, že matematické operace v Matlabu jsou relativně málo náročné na čas CPU. Na druhé straně jakékoli operace s řetězci (vyhledávání a identifikace znaků, převod čísel apod.) jsou neuvěřitelně pomalé. Proto bude pravděpodobně nutno pro čtení netlistu, pro vytváření tabulek *radg*, *slpg*, *parg*, *radc*, *slpc*, *parc* a pro některé další operace napsat specializované m-funkce v jazyku C.

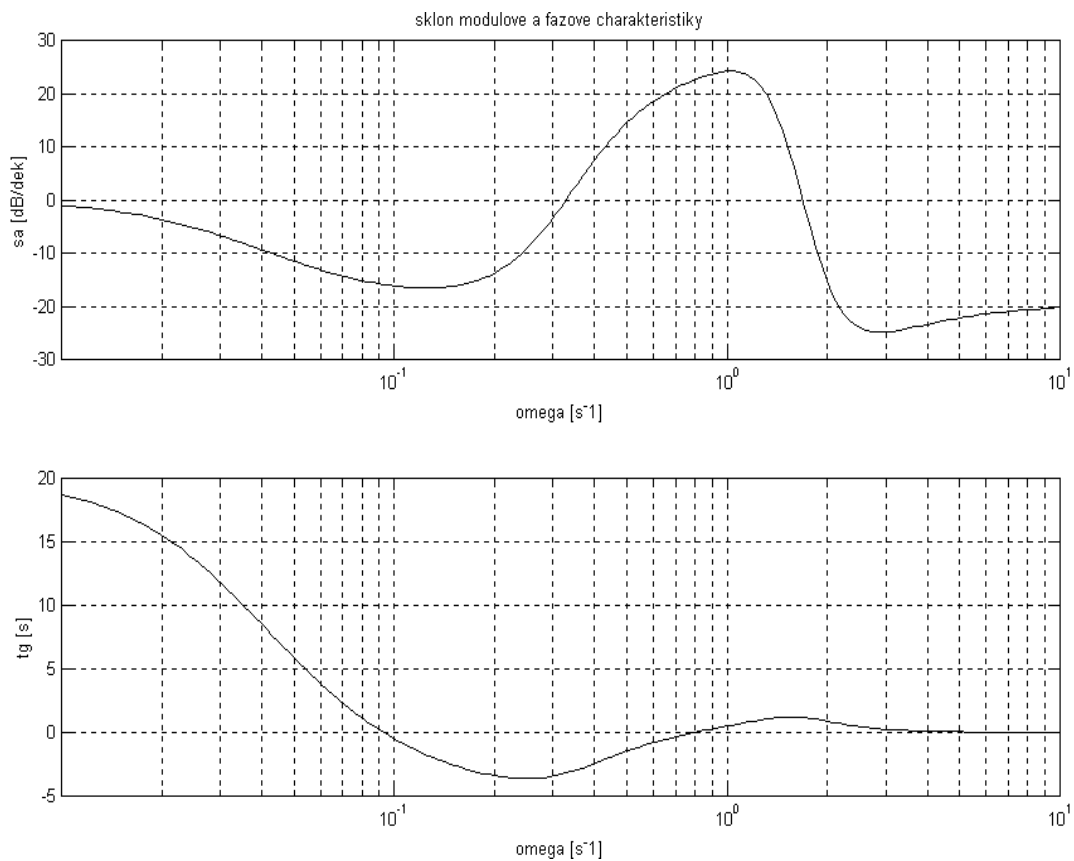
Práce na tomto projektu byly podporovány granty GAČR č. 102/98/0782 a 102/98/0130.

## Literatura

[1] MATLAB, The Language of Technical Computing, Version 5.2, The Math Works, Inc., 1998

[2] Vlach, J.: Computer Methods for Circuit Analysis and Design, 2<sup>nd</sup> Edition, Van Nostrand Reinhold, 1994






---

Prof. Ing. Juraj Valsa, CSc., Ústav teoretické a experimentální elektrotechniky  
 Fakulta elektrotechniky a informatiky VUT v Brně, Purkyňova 118, 612 00 Brno  
 tel (05) 4114 9514, e-mail [valsa@utee.fee.vutbr.cz](mailto:valsa@utee.fee.vutbr.cz)