

ЕЛЕКТРОННІ КОЛА ТА СИГНАЛИ

УДК 621.372.061

Ю. І. Шаповалов, Б. А. Мандзій, Д. Р. Бачик
Національний університет “Львівська політехніка”

РОЗШИРЕННЯ МОЖЛИВОСТЕЙ СИСТЕМИ ПРОГРАМНИХ ФУНКЦІЙ МАОРСs СТОСОВНО ФОРМУВАННЯ ПЕРЕДАВАЛЬНИХ ХАРАКТЕРИСТИК ЛІНІЙНИХ ПАРАМЕТРИЧНИХ КІЛ

© Шаповалов Ю. І., Мандзій Б. А., Бачик Д. Р., 2016

Описано розширення можливостей системи МАОРСs стосовно обчислення передавальних функцій лінійних параметричних кіл, що являють собою відношення довільних напруг та струмів кола. Теоретичною основою обчислень для такого розширення вибрано табличний метод формування математичної моделі кола.

Ключові слова: спряжена та нормальна параметричні передавальні функції; частотний символічний метод, система програмних функцій МАОРСs.

Yu. Shapovalov, B. Mandziy, D. Bachyk
Lviv Polytechnic National University

THE EXPANSION OF THE POSSIBILITIES OF THE SYSTEM PROGRAM FUNCTIONS MAORCs FOR THE STUDY OF LINEAR PERIODICALLY-TIME-VARIABLE CIRCUITS

© Shapovalov Yu., Mandziy B., Bachyk D., 2016

The system program function MAORCs is designed to study multivariate analysis and optimization of state linear periodically-time-variable circuits. The architecture of the system MAORCs is based on the principles of software Matlab environment, which provides the existence of the independent the software functions and creating the user-defined functions, and because it is an open system, it allows you to supplement it with the new functions. These additions extend the functionality of the system MAORCs, make its application more efficient for analysis, research or design of the electronic devices, that are modeled by the linear periodically-time-variable circuits. The mathematical software of the system MAORCs based on the frequency symbolic method, which allows us to calculate the conjugate parametric transfer functions of linear periodically-time-variable circuits. Such transfer functions linking the input signals with the output signals in the form of approximating polynomials of Fourier in trigonometric or complex form. By the same polynomials of Fourier are approximated the normal parametric transfer functions, which are the basis of assessment of the asymptotic stability of the circuit.

The system MAORCs implements symbolic methods of calculation, because the conjugate and normal transfer functions usually contain a number of parameters of the elements, that are defined by characters. Substitution of the numerical values in place of the following characters is performed, usually, in the last stages of solving the relevant tasks of the device designing.

The limitation of the system MAORCs is that the current system functions can only determine the transfer functions, which are the ratio nodal voltage of the circuit acting on the circuit signals in the form of the current sources.

The aim of this work is to expand the functional capabilities of the system MAORCs with respect to the computation of the transfer functions, that are representing random variables of

the circuit, among which are the arbitrary node voltages of the circuit, and the arbitrary voltages and currents on the elements of this circuit. Theoretical calculations for such an expansion are using the tabular describing method of the mathematical model of the circuit.

Key words: conjugate and normal parametric transfer functions, frequency symbolic method, the system functions MAOPCs.

Вступ

У роботах [1, 2] описано систему програмних функцій MAOPC, яка призначена для проведення автоматизованого багатоваріантного аналізу та оптимізації лінійних параметричних кіл. Архітектура системи MAOPCs основана на принципах програмного середовища Matlab [3], що передбачає існування незалежних між собою програмних функцій, що містять user-defined functions (функції, що визначає користувач), а тому вона є відкритою системою, що дає змогу доповнювати її новими такими функціями. Ці доповнення розширюють функціональні можливості системи MAOPCs, роблять її застосування ефективнішим для аналізу, дослідження чи проектування радіотехнічних пристроїв, що моделюються лінійними параметричними колами. Математичне забезпечення системи MAOPCs основане на частотному символьному методі [4, 5, 6] (ЧС-метод), який дає змогу обчислювати спряжені параметричні передавальні функції лінійних параметричних кіл $W(s,t)$, що зв'язують вхідні сигнали з вихідними у вигляді апроксимуючих поліномів Фур'є у тригонометричній чи комплексній формі [4]. Такими ж поліномами Фур'є апроксимуються нормальні параметричні передавальні функції $W(s,\xi)$ [4], де ξ – момент подавання на коло δ -функції, які є основою оцінки асимптотичної стійкості кола. Для визначення поліномів $W(s,t)$ та $W(s,\xi)$ у системі MAOPCs передбачено відповідні програмні функції, поряд з якими наявні програмні функції визначення коефіцієнтів чутливості, розрахунку миттєвих значень вихідних змінних, проведення параметричної оптимізації та аналізу допусків й інші, які у сукупності незалежні, тому можуть бути використані незалежно одна від одної. Тут є повна аналогія, наприклад, з командами мови асемблера, які теж незалежні одна від одної, а їх послідовності створюють різноманітні програми перетворення вхідних даних на дані результату.

Система MAOPCs реалізує символьні методи розрахунків, тому передавальні функції $W(s,t)$ та $W(s,\xi)$ часто містять параметри елементів кола, що задані символами. Підстановка числових значень на місце таких символів виконується, зазвичай, на останніх етапах проектування пристрою.

Обмеженням системи MAOPCs є те, що її програмні функції дозволяють визначати тільки передавальні функції, які є відношенням вузлових напруг кола до джерел струму, що діють на коло. Це зумовлено тим, що математична модель кола будується за методом вузлових напруг, у якому невідомими змінними є лише згадані вузлові напруги кола, а заданими вхідними сигналами – джерела струму. Ця обставина у деяких випадках спричиняє незручності в аналізі кіл, і тому актуальним є завдання усунення згаданого обмеження.

Метою цієї роботи є розширення функціональних можливостей системи MAOPCs стосовно обчислення передавальних функцій, що являють собою відношення довільних змінних кола, серед яких і довільні вузлові напруги кола, і довільні напруги та струми на елементах цього кола. Як теоретичну основу обчислень для такого розширення використано табличний метод опису математичної моделі кола [7].

Основна частина

Для досягнення поставленої мети у системі MAOPCs реалізуємо програмну функцію формування математичної моделі лінійного параметричного кола *FormOfMatModelByTabMethod()*, що основана на табличному методі опису кіл [7]:

$$\begin{bmatrix} 1 & 0 & -A^t \\ Y_r & Z_r & 0 \\ 0 & A & 0 \end{bmatrix} \cdot \begin{bmatrix} U_r \\ I_r \\ U_b \end{bmatrix} = \begin{bmatrix} 0 \\ W \\ 0 \end{bmatrix}, \quad (1)$$

де 1 – одинична матриця, A – матриця інцидентій, A^t – транспонована матриця інцидентій, Y_r – матриця провідностей гілок, Z_r – матриця опорів гілок, W – вектор струмів та напруг незалежних джерел, U_b – вектор вузлових напруг кола, U_r та I_r – вектори напруг та струмів гілок кола, відповідно.

Перед використанням програмної функції *FormOfMatModelByTabMethod()* користувач створює файл “input.m”, у якому вказує параметри та структуру заданого кола. Тільки за наявності файла “input.m” програмна функція *FormOfMatModelByTabMethod()* сформує математичну модель кола.

Формування файла “input.m” розглянемо для одноконтурного параметричного підсилювача з двома параметричними елементами (рис. 1).

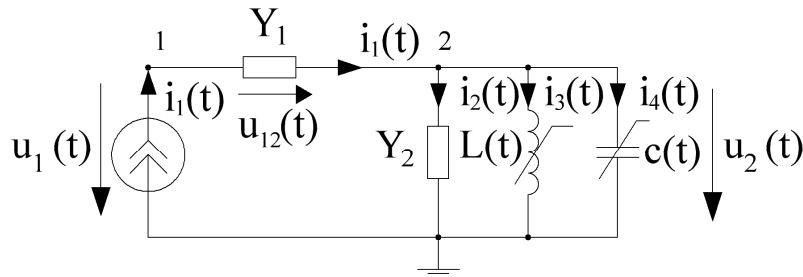


Рис. 1. Модель одноконтурного параметричного підсилювача з двома параметричними елементами
 $c(t) = c_0 \cdot (1 + m_c \cdot \cos(\Omega \cdot t + \varphi_c))$, $c_0 = 10 \cdot 10^{-12} \text{ Ф}$, $L(t) = L_0 \cdot (1 + m_L \cdot \cos(\Omega \cdot t + \varphi_L))$, $L_0 = 0.2533 \cdot 10^{-6} \text{ Гн}$,
 $Y_1 = 0.25 \text{ См}$, $Y_2 = 0.0004 \text{ См}$, $i(t) = I_m \cdot \cos(\omega_c \cdot t + \varphi)$, $I_m = 0.0001 \text{ А}$, $\varphi = -\pi/4$, $m_c = 0.01$, $m_L = 0.1$,
 $\Omega = 4 \cdot \pi \cdot 10^8 \text{ рад/с}$, $\omega_c = 2 \cdot \pi \cdot 10^8 \text{ рад/с}$

Цей файл складається з шести блоків. Розглянемо його структуру.

Блок 1. Параметри параметричних елементів кола.

У першому блоці перераховуються параметри кожного параметричного елемента у колі. Ці параметри зберігаються в структурі *Param{}*, яка містить три стовпці: у першому вказується назва параметра, у другому – його значення, у третьому – ознака підстановки (“0” – числове значення, “1” – параметр у символі). Для підсилювача з рис. 1 вигляд структури *Param{}* показаний на рис. 2.

```

%%% BLOCK 1_parameters of parametric elements %%%
%%% <змінна> <значення> <ознака підстановки>
Param{1,1}='c0'; Param{1,2}=10e-12; Param{1,3}=1; syms(Param{1,1});
Param{2,1}='mc'; Param{2,2}=0.127; Param{2,3}=1; syms(Param{2,1});
Param{3,1}='Omega' Param{3,2}=4*pi*1e8; Param{3,3}=1; syms(Param{3,1});
Param{4,1}='L0'; Param{4,2}=0.2533e-6; Param{4,3}=1; syms(Param{4,1});
Param{5,1}='mL'; Param{5,2}=0.1; Param{5,3}=1; syms(Param{5,1});
Param{6,1}='ph_c'; Param{6,2}=0; Param{6,3}=1; syms(Param{6,1});
Param{7,1}='ph_L'; Param{7,2}=pi; Param{7,3}=1; syms(Param{7,1});
save(OutputFilePath,'Param','-append');

```

Рис. 2. Вигляд структури *Param{}* для підсилювача з рис. 1

Блок 2. Функції, що описують параметричні елементи.

Параметричні елементи задатся у структурі *ParamFuncs{}*. Для підсилювача з рис. 1 вигляд структури *ParamFuncs{}* показано на рис. 3.

```

%%% BLOCK 2 _ Laws change of parameters of parametric elements %%%
ParamFuncs(1,1)=c0*(1+mc*cos(Omega*t+ph_c));
ParamFuncs(2,1)=L0*(1+mL*cos(Omega*t+ph_L));
save(OutputFilePath,'ParamFuncs','-append');
NumOfParametricElements=size(ParamFuncs,1);
save(OutputFilePath,'NumOfParametricElements','-append');

```

Рис. 3. Вигляд структури *ParamFuncs{}* для кола з рис. 1

Параметричний елемент записується в окремий елемент структури *ParamFuncs{}*.

Блок 3. Параметри джерел сигналів (напруги чи струму).

В цьому блоці перераховуються параметри кожного джерела сигналу, наявного у колі. Ці параметри зберігаються у структурі *SignalParam{}*, яка має вигляд, аналогічний структурі *Param{}* з рис. 2. Структура *SignalParam{}* для підсилювача з рис. 1 показана на рис. 4.

```

%%% BLOCK 3_Parameters of Input Signals %%%
%%% <змінна>          <значення>          <ознака підстановки>
SignalParam{1,1}='A0'; SignalParam{1,2}=0.0001; SignalParam{1,3}=0; syms(SignalParam{1,1});
SignalParam{2,1}='w'; SignalParam{2,2}=2*pi*1e8; SignalParam{2,3}=0; syms(SignalParam{2,1});
SignalParam{3,1}='ph'; SignalParam{3,2}=(pi/4); SignalParam{3,3}=0; syms(SignalParam{3,1});
save(OutputFilePath,'SignalParam','-append');

```

Рис. 4. Структура SignalParam{} для підсилювача з рис. 1

Блок 4. Функції, що описують зміну вхідних сигналів.

Четвертий блок задає наявні у колі вхідні сигнали. Для гармонічного джерела струму $i(t) = I_m \cdot \cos(\omega_c \cdot t + \varphi)$ підсилювача з рис. 1 вигляд структури InputSignal{} зображено на рис. 5.

```

%%% BLOCK 4_Functions of Input Signals%%%
InputSignal{1,1}=A0*exp(i*(w*t+ph)); InputSignal{1,2}=w;
save(OutputFilePath,'InputSignal','-append');

```

Рис. 5. Вигляд структури InputSignal{} для одноконтурного параметричного підсилювача з рис. 1

Як і в попередньому блоці, тут створюється окрема структура InputSignal{}, у кожен елемент якої записуються параметри та форма сигналу джерела.

Блок 5. Список елементів кола.

П'ятий блок у файлі "input.m" містить список всіх елементів досліджуваного кола, їх параметри та вузли з'єднання. Тут формується структура Vars{}, яка містить шість стовпців: у першому стовпці вказується назва елемента, у другому – значення його параметра, у третьому – ознака підстановки ("0" – числове значення, "1" – параметр у символі), у четвертому – пара вузлів, з якими з'єднаний відповідний елемент, у п'ятому – ознака параметричності елемента кола ("0" – параметр елемента постійний, "1" – елемент параметричний) і у шостому стовпці вказуємо, чи елемент кола є керованим джерелом (записуємо "номер керуючої гілки", якщо елемент є керованим джерелом, у іншому випадку записуємо "0"). На рис. 6 представлено вигляд структури Vars{} для підсилювача з рис. 1.

```

%%% BLOCK 5_Elements list in circuit %%%
% 1 – Назва елемента % 2 – Значення параметра елемента % 3 – ознака підстановки % 4 – вузли, між якими
розміщений елементи в колі % 5 – ознака параметричності елемента % 6 – ознака залежності джерел живлення
Vars{1,1}='I'; Vars{1,2}=InputSignal(1,1); Vars{1,3}=0; Vars{1,4}=[0 1]; Vars{1,5}=0; Vars{1,6}=0;
syms(Vars{1,1});
Vars{2,1}='Y1'; Vars{2,2}=0.25; Vars{2,3}=0; Vars{2,4}=[1 2]; Vars{2,5}=0; Vars{2,6}=0;
syms(Vars{2,1});
Vars{3,1}='Y2'; Vars{3,2}=0.0004; Vars{3,3}=0; Vars{3,4}=[2 0]; Vars{3,5}=0; Vars{3,6}=0;
syms(Vars{3,1});
Vars{4,1}='L'; Vars{4,2}=ParamFuncs(2,1); Vars{4,3}=0; Vars{4,4}=[2 0]; Vars{4,5}=1; Vars{4,6}=0;
syms(Vars{4,1});
Vars{5,1}='C'; Vars{5,2}=ParamFuncs(1,1); Vars{5,3}=0; Vars{5,4}=[2 0]; Vars{5,5}=1; Vars{5,6}=0;
syms(Vars{5,1});
save(OutputFilePath,'Vars','-append');
NumOfBranches= size(Vars,1); % кількість (гілок)
save(OutputFilePath,'n','-append');
NumOfNodes=2; % кількість вузлів у схемі
save(OutputFilePath,'NumOfNodes','-append');

```

Рис. 6. Вигляд структури Vars{} для одноконтурного параметричного підсилювача з рис. 1

Блок 6. Глобальні змінні.

Шостий блок містить глобальні параметри для проведення дослідження. На рис. 7 подано вигляд шостого блока для підсилювача з рис. 1.

Розглянувши структуру файла “input.m”, подаємо алгоритм побудови математичної моделі кола (рис. 8) та фрагменти його програмної реалізації (рис. 9–11) у програмній функції *FormOfMatModelByTabMethod()*.

```

%%% BLOCK 6_Global parameters %%%
Approximation = 0; % 0-exponential ; 1-trigonometrical
save(OutputFilePath,'Approximation','-append');
Solver_Type = 2; % 0-Matlab ; 1-Maple ; 2- Kramer;
save(OutputFilePath,'Solver_Type','-append');
DigitsInVpa = 12; % вибір точності розрахунків,
save(OutputFilePath,'DigitsInVpa','-append');
NumOfHarm=4; % кількість гармонік;
save(OutputFilePath,'NumOfHarm','-append');

```

Рис. 7. Глобальні параметри з файла “input.m” для одноконтурного параметричного підсилювача з рис. 1

Після завантаження даних з файла “input.m”, програмна функція *FormOfMatModelByTabMethod()* формує матриці A та A^t . Для цього з четвертого стовпця структури *Vars{}* зчитується інформація про підключення елементів до вузлів кола. На рис. 9 представлено фрагмент програмного коду, що відповідає такій процедурі.

Далі програмна функція *FormOfMatModelByTabMethod()* формує матриці Y_r, Z_r та вектор W . При цьому програмна функція аналізує кожний елемент кола, зчитуючи його ім'я з першого стовпця структури *Vars{}*. Після визначення типу елемента програмна функція у відповідні позиції матриць Y_r, Z_r чи вектора W записує параметр цього елемента. На рис. 10 подано фрагмент програмного коду, що відповідає такій процедурі.

Під час формування цих матриць у програмній функції *FormOfMatModelByTabMethod()* передбачена також перевірка кожного елемента на параметричність та належність цього елемента керованому джерелу. Якщо елемент є параметричним, то під час формування матриць Y_r і Z_r враховується ще похідна його параметра за часом. Якщо у схемі є кероване джерело, то у шостому стовпці структури *Vars{}* вказано порядковий номер гілки, що керує цим джерелом. Далі визначається тип керованого джерела, що дає змогу програмній функції *FormOfMatModelByTabMethod()* коректно сформувати матриці Y_r, Z_r та вектор W . Далі на основі матриць $Y_r, Z_r, A, A^t, 1$ та вектора W програмна функція *FormOfMatModelByTabMethod()* формує математичну модель кола (1). На рис. 11 представлено фрагмент програмного коду, що відповідає формуванню матриць Y_r, Z_r , вектора W та математичної моделі кола (1).

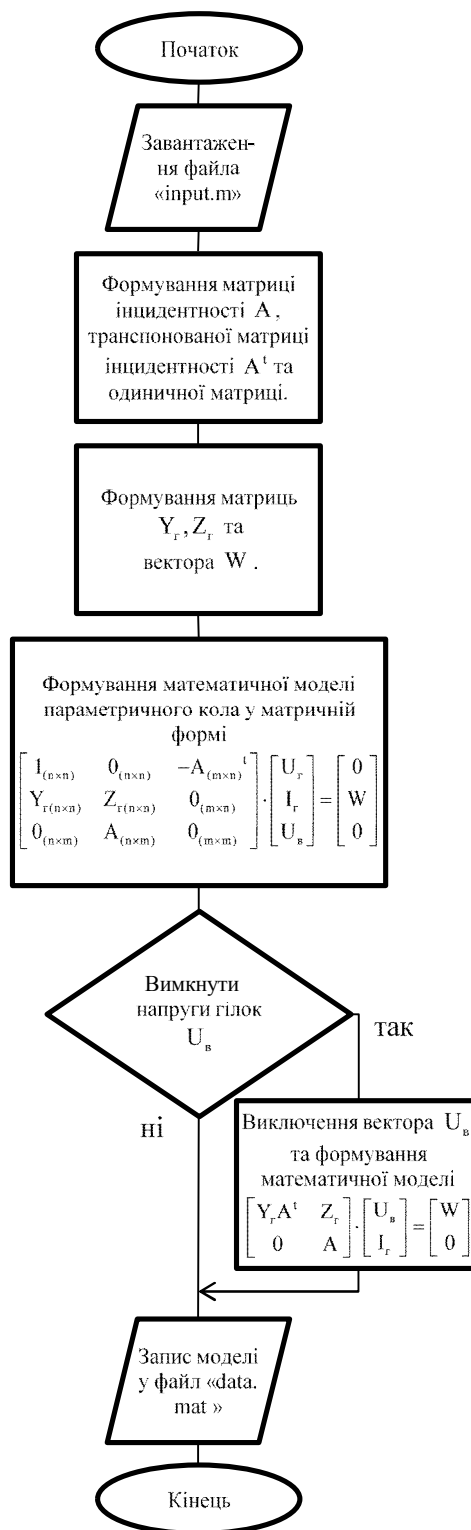


Рис. 8. Алгоритм побудови математичної моделі лінійного параметричного кола

```

for i=1:n
    for j=1:2 % читати інформацію про
        розміщення елемента з Vars{ }
            X(i,j)=Vars{i,4}(1,j);
        end;
    end;
    m = max(max(X))+1; % кількість вузлів у колі
    X(X==0) = m; % замінити всі '0' в матриці
    A = zeros(m,n); % створити матрицю розміром (m,n)
    for i=1:n
        A(X(i,1),i) = 1;% гілка "i" починається з вузла
        "X(i,1)"
        A(X(i,2),i) = -1; % i закінчується на вузлі "X(i,2)"
    end;
    A(m,:)=[] % матриця інцидентій
    At = A' % транспонована матриця
    save(OutputFilePath,'A','-append');
    save(OutputFilePath,'At','-append');

```

Рис. 9. Фрагмент програмного коду програмної функції *FormOfMatModelByTabMethod()* у середовищі *Matlab*

```

for i=1:n % для всіх елементів
    NAME = Vars{i,1}; % прочитати назву елемента
    switch NAME(1) % тип елемента
        case 'L' % індуктивність
            Yv(i,i) = 1;
            Wv(i,1) = 0;
            if Vars{i,5} == 1% перевірка, чи елемент
                параметричний% Zv(i,i) = -s*L - dL/dt
                    Zv(i,i) = -s*component(i,1) - component(i,2);
                else
                    Zv(i,i) = -s*component(i,1);% Zv(i,i) = -s*L
                end
            end
        case 'U' % джерело напруги
            Yv(i,i) = 1;
            Zv(i,i) = 0;
            Wv(i,1) = component(i,1); % Wv(i,i) = U
        case 'I' % джерело струму
            Yv(i,i) = 0;
            Zv(i,i) = 1;
            Wv(i,1) = component(i,1); % Wv(i,i) = I
        otherwise
            disp('Error!!! No such element!')
    end;
end;
end;

```

Рис. 10. Фрагмент програмного коду програмної функції *FormOfMatModelByTabMethod()* у середовищі *Matlab*

У деяких випадках у моделі кола не потрібно мати всі змінні з (1), тому в програмній функції *FormOfMatModelByTabMethod()* передбачена можливість вилучення з системи (1) невідомих напруг гілок. Для цього аргументом програмної функції *FormOfMatModelByTabMethod()* є параметр *excludeVars*. Якщо він дорівнює "1", то з системи рівнянь вилучається вектор напруг гілок U_r , який визначається з першого рівняння (1) і підставляється у друге рівняння (1) цієї системи. Система рівнянь, що описує коло, набуває вигляду:

$$\begin{bmatrix} Z_r & Y_r A^t \\ A & 0 \end{bmatrix} \cdot \begin{bmatrix} I_r \\ U_B \end{bmatrix} = \begin{bmatrix} W \\ 0 \end{bmatrix}. \quad (2)$$

Загалом для підсилювача з рис. 1 програмна функція *FormOfMatModelByTabMethod()* формує математичну модель кола у вигляді (1), де

$$A = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 1 \end{bmatrix}, Y_r = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & Y_1 & 0 & 0 & 0 \\ 0 & 0 & Y_2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & sc(t) + c'(t) \end{bmatrix}, Z_r = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -(sL(t) + L'(t)) & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix},$$

$$U_r = \begin{bmatrix} U_{r1} \\ U_{r2} \\ U_{r3} \\ U_{r4} \\ U_{r5} \end{bmatrix}, I_r = \begin{bmatrix} I_{r1} \\ I_{r2} \\ I_{r3} \\ I_{r4} \\ I_{r5} \end{bmatrix}, U_B = \begin{bmatrix} U_{B1} \\ U_{B2} \end{bmatrix}, W = \begin{bmatrix} -I(s) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (3)$$

Математична модель (3) дає змогу в програмному середовищі системи *MAOPCs* для підсилювача з рис. 1 формувати довільні передавальні функції $W(s, t)$ та $W(s, \xi)$, які визначаються відношенням змінних з векторів U_r, I_r, U_r до змінних з вектора W . На рис. 12 наведено фрагмент програмного коду, що відповідає процедурі визначення передавальних функцій $W_{U_i}(s, t) = U_i(s, t)/I_1(s)$,

$W_{Y_2}(s, t) = I_2(s, t)/I_1(s)$, $W_L(s, t) = I_3(s, t)/I_1(s)$, $W_C(s, t) = I_4(s, t)/I_1(s)$, $W_{U_2}(s, t) = U_2(s, t)/I_1(s)$ та миттєвих потужностей $P(t)_{i(t)}$, $P(t)_{Y_1}$, $P(t)_{Y_2}$, $P(t)_L$, $P(t)_C$ підсилювача з рис. 1.

```

for i=1:n % для всіх елементів
    if Vars{i,6} ~= 0 % перевірка на залежність джерел
        j = Vars{i,6}; % порядковий номер керуючого джерела
        MASTER = Vars{Vars{i,6},1}; % назва керуючого джерела
        SLAVE = Vars{i,1}; % назва керованого джерела
        if (MASTER(1) == 'U') && (SLAVE(1) == 'I') % I ← U
            Yv(j,j) = 0; Yv(j,i) = 0; Yv(i,j) = Vars{i,2}; Yv(i,i) = 0;
            Zv(j,j) = 1; Zv(j,i) = 0; Zv(i,j) = 0; Zv(i,i) = -1;
            Wv(j,1) = 0; Wv(i,1) = 0;
        elseif (MASTER(1) == 'U') && (SLAVE(1) == 'U') % U ← U
        save(OutputFilePath,'Yv','append');
        save(OutputFilePath,'Zv','append');
        save(OutputFilePath,'Wv','append');
        %%%%%%%%%%%%%%%
        MatY = sym('MatY',[2*n+m-1 2*n+m-1]);
        W = sym('W',[2*n+m-1 1]);
        MatY(1:n,1:n) = diag(ones(n,1));
        MatY(1:n,(n+1):2*n) = zeros(n,n);
        MatY((2*n+1):(2*n+m-1),1:n) = zeros(m-1,n);
        MatY((n+1):2*n,(2*n+1):(2*n+m-1)) = zeros(n,m-1);
        MatY((2*n+1):(2*n+m-1),(2*n+1):(2*n+m-1)) = zeros(m-1,m-1);
        MatY((n+1):2*n,1:n) = Yv;
        MatY((n+1):2*n,(n+1):2*n) = Zv;
        MatY((2*n+1):(2*n+m-1),(n+1):2*n) = A;
        MatY(1:n,(2*n+1):(2*n+m-1)) = -At;
        W(1:n) = zeros(n,1);
        W((n+1):2*n) = Wv;
        W((2*n+1):(2*n+m-1)) = zeros(m-1,1);
        save(OutputFilePath,'MatY','append');
        save(OutputFilePath,'W','append');

```

Рис. 11. Фрагмент програмного коду програмної функції *FormOfMatModelByTabMethod()* у середовищі *Matlab*

```

TrFunc(1,'s','t') % передавальна функція  $i_1(t)$  у  $u_1(t)$ ;
TrFunc(8,'s','t') % передавальна функція  $i_1(t)$  у  $i_2(t)$ ;
TrFunc(9,'s','t') % передавальна функція  $i_1(t)$  у  $i_3(t)$ ;
TrFunc(10,'s','t') % передавальна функція  $i_1(t)$  у  $i_4(t)$ ;
TrFunc(12,'s','t') % передавальна функція  $i_1(t)$  у  $u_2(t)$ ;.....
OutVar('classic',2,TF{1,4},i*2*pi*1e8,'t') % визначення  $U_1(s,t)$ ;
OutVar('classic',2,TF{8,4},i*2*pi*1e8,'t') % визначення  $I_2(s,t)$ ;
OutVar('classic',2,TF{9,4},i*2*pi*1e8,'t') % визначення  $I_3(s,t)$ ;
OutVar('classic',2,TF{10,4},i*2*pi*1e8,'t') % визначення  $I_4(s,t)$ ;
OutVar('classic',2,TF{12,4},i*2*pi*1e8,'t') % визначення  $U_2(s,t)$ ;.....
Re_OutVariables{1,1}=real(OutVariables{1,1}{4,1}); %  $u_1(t)$ ;
Re_OutVariables{1,8}=real(OutVariables{1,8}{4,1}); %  $i_2(t)$ ;
Re_OutVariables{1,9}=real(OutVariables{1,9}{4,1}); %  $i_3(t)$ ;
Re_OutVariables{1,10}=real(OutVariables{1,10}{4,1}); %  $i_4(t)$ ;
Re_OutVariables{1,12}=real(OutVariables{1,12}{4,1}); %  $u_2(t)$ 
% Блок визначення миттєвої потужності у гілках кола;
PowerInTheBranch(1,[Re_OutVariables{1,1},InputSignal{1,1}]);
PowerInTheBranch(1,[Re_OutVariables{1,2},InputSignal{1,1}]);
PowerInTheBranch(1,[Re_OutVariables{1,8},Re_OutVariables{1,12}]);
PowerInTheBranch(1,[Re_OutVariables{1,9},Re_OutVariables{1,12}]);
PowerInTheBranch(1,[Re_OutVariables{1,10},Re_OutVariables{1,12}]);
TransferCoefficientOfPower=(Re_OutVariables{1,1},InputSignal{1,1},Re_OutVariables{1,8},Re_OutVariables{1,12})%
визначення коефіцієнта передачі по потужності

```

Рис. 12. Фрагмент програмного коду визначення передавальних функцій, миттєвих напруг та потужностей у СФ МАОРС

Для визначення передавальних функцій у середовищі СФ МАОРСs необхідно записати програмну функцію $TrFunc()$, у якій вказати номер передавальної функції, що потрібно визначити.

У табл.1 та на рис. 13–15 наведено результати застосування програмних функцій $TrFunc()$, $OutVar()$, $Re_OutVariables\{\}$, $PowerInTheBranch()$ під час дослідження підсилювача з рис. 1.

Таблиця 1

№	t, c	0.800e-6	0.801e-6	0.802e-6	0.803e-6	0.804e-6
1	$i_2(t), A$	0.00033	0.00046	0.00042	0.00020	-8.67121e-5
2	$i_3(t), A$	0.00509	0.00074	-0.00361	-0.00652	-0.00724
3	$i_4(t), A$	-0.00534	-0.00110	0.00328	0.00636	0.00731
4	$u_1(t), B$	-0.82034	-1.14495	-1.05402	-0.51003	0.21684
5	$u_2(t), B$	0.82006	1.14457	1.05366	0.50986	-0.21678
6	$P(t)_{i(t)}, Bm$	0.20000e-7	0.39021e-7	0.31756e-7	0.82443e-8	0.97887e-9
7	$P(t)_{j_1}, Bm$	0.26900e-3	0.52401e-3	0.44408e-3	0.10398e-3	0.18797e-4
8	$P(t)_{j_2}, Bm$	0.41721e-2	0.85249e-3	-0.38040e-2	-0.33239e-2	0.15689e-2
9	$P(t)_L, Bm$	-0.43832e-2	-0.12635e-2	0.34538e-2	0.32431e-2	-0.15844e-2
10	$P(t)_C, Bm$	-0.57920e-4	-0.11304e-3	-0.93912e-4	-0.23188e-4	-0.32980e-5

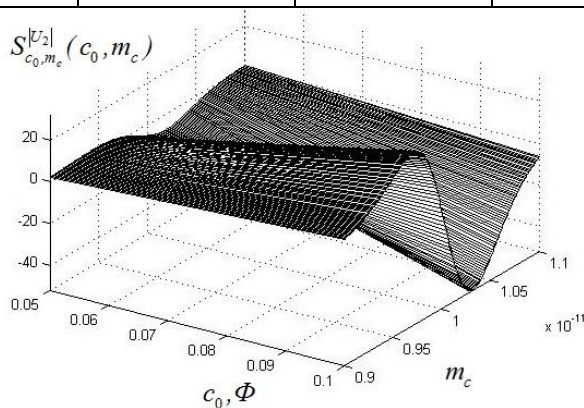


Рис. 13. Залежність чутливості $S_{c_0, m_c}^{[U_2]}(c_0, m_c)$ модуля вихідної напруги $u_2(t)$ від c_0 та m_c , якщо $\omega = 2 \cdot \pi \cdot 10^8$ рад / с та $t = 5 \cdot 10^{-9}$

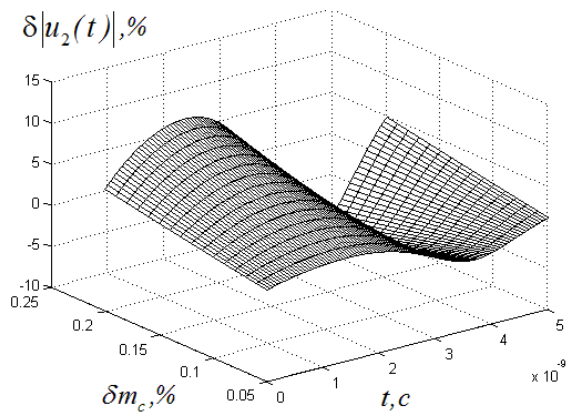


Рис. 14. Залежність відносного відхилення $\delta|u_2(t)|$ вихідної напруги $u_2(t)$ від часу та відхилення δm_c у межах 5–25 %

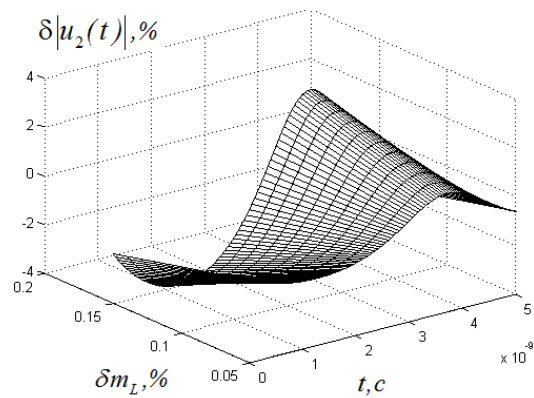


Рис. 15. Залежність відносного відхилення $\delta|u_2(t)|$ вихідної напруги $u_2(t)$ від часу та відхилення δm_L у межах 5–20 %

Висновки

Застосування у системі MAOPCs табличного методу формування рівнянь та реалізація його програмною функцією *FormOfMatModelByTabMethod()* дозволяє під час дослідження лінійних параметричних кіл обчислювати та контролювати їх довільні передавальні функції $W(s,t)$ та $W(s,\xi)$, що робить таке дослідження змістовнішим та різнобічнішим.

1. Shapovalov Yu., Mandziy B. and Bachyk D. The system functions MAOPCs for analysis and optimization of linear periodically time-variable circuits based on the frequency symbolic method // *Przegląd Elektrotechniczny*: 2014. Vol. 91. no 7. P. 39–42. URL: <http://pe.org.pl/articles/2015/7/13.pdf>.
2. Шаповалов Ю. І., Мандзій Б. А., Бачик Д. Р. Багатоваріантний аналіз та оптимізація лінійних параметричних кіл у програмному середовищі MAOPCs // *Вісн. НТУУ “КПІ”. Радіотехніка. Радіоапаратобудування*. 2015. № 60. С. 34–44. URL: <http://radap.kpi.ua/radiotechnique/article/viewFile/940/988-radap.pdf>.
3. Дьяконов. В. П. MATLAB 7.*/R2006/R2007: Самоучитель. М.: ДМК Пресс, 2008. 768 с.
4. Шаповалов Ю. І. Символьний аналіз лінійних електричних кіл у частотній області. Постійні та змінні параметри. Львів: Видавництво Нац. ун-ту “Львівська політехніка”, 2014. 324 с.
5. Shapovalov Yu., Mandziy B. and Mankovsky S. The peculiarities of analysis of linear parametric circuit performed by frequency-symbolic method. *Przegląd Elektrotechniczny*. 2010. Vol. 86. no 1. P. 158–160.
6. Shapovalov Yu., Mandziy B. Frequency symbolic analysis of linear periodically time-varying circuits with many parametric elements // *Przegląd Elektrotechniczny*. 2014. Vol. 90., no 5. P. 64–66. <http://pe.org.pl/articles/2014/5/13.pdf>.
7. Петренко А. В., Тимченко А. П., Власов А. И. Табличные методы анализа электронных схем. К.: Высшая школа. 1977. 192 с.

References

1. Yu. Shapovalov, B. Mandziy and D. Bachyk “The system functions MAOPCs for analysis and optimization of linear periodically time-variable circuits based on the frequency symbolic method”, *Przegląd Elektrotechniczny*, vol. 91, no 7, pp. 39–42, 2014, <http://pe.org.pl/articles/2015/7/13.pdf>.
2. Yu. Shapovalov, B. Mandziy and D. Bachyk “Multivariate analysis and optimization of linear periodically time-variable circuits at the environment of MAOPCs”, *Bulletin of National Technical University of Ukraine. Series Radiotechnique. Radioapparatus building*. – 2015. – № 60. – 34–44, <http://radap.kpi.ua/radiotechnique/article/viewFile/940/988-radap.pdf>.
3. Dyakonov. V.P. MATLAB 7.*/R2006/R2007: Tutorial / V. P. Dyakonov. – М.: ДМК Press, 2008. – 768 p.
4. Shapovalov Yu. Symbolic analysis of linear electrical circuits in the frequency domain. Fixed and variable parameters. Lviv, Lviv Polytechnic National University publication, 2014. – 324 p.
5. Yu. Shapovalov, B. Mandziy, and S. Mankovsky, “The peculiarities of analysis of linear parametric circuit performed by frequency-symbolic

method”, *Przegląd Elektrotechniczny*, vol. 86, no 1, pp. 158–160, 2010. 6. Yu. Shapovalov, B. Mandziy “Frequency symbolic analysis of linear periodically time-varying circuits with many parametric elements”, *Przegląd Elektrotechniczny*, vol. 90, no 5, pp. 64–66, 2014, <http://pe.org.pl/articles/2014/5/13.pdf>. 7. Petrenko A.V. *Tabular methods of analysis of electronic circuits* / A. V. Petrenko, A. P. Timchenko, A. I. Vlasov. – Kyiv, Ukraine: publication “Higher School”, 1977. – 192 p.