

Gerd Reszka

Ośrodek Elektronicznej Techniki Obliczeniowej
Politechniki Śląskiej

O PEWNYM SPOSOBIE ANALIZY STAŁOPRĄDOWEJ UKŁADÓW ELEKTRONICZNYCH
Z ZASTOSOWANIEM NIELINIOWYCH MODELI DIODY I TRANZYSTORA

Streszczenie. Praca zawiera opis metody analizy stałoprądowej pewnej klasy nieliniowych obwodów elektronicznych. Podany jest także schemat i tekst programu na EMC realizującego obliczenia opisaną metodą oraz, na przykładzie, pokazane przygotowanie prostego obwodu do postaci wymaganej przez program.

1. Modele diody i tranzystora

Z literatury [1], [2] wiadomo, że półprzewodnikowe złącze p-n można dość dokładnie opisać następującą charakterystyką napięciowo prądową

$$i_d = i_0 (e^{a_T u_d} - 1), \quad (1)$$

gdzie i_0 jest prądem termicznym diody, a_T jest odwrotnością tzw. potencjału termicznego, a i_d oraz u_d są odpowiednio prądem płynącym przez diodę oraz napięciem na niej. Jeśli uwzględnimy jeszcze również wykładniczą zależność prądu termicznego od temperatury złącza, to opisany model dość wiernie charakteryzuje diodę półprzewodnikową.

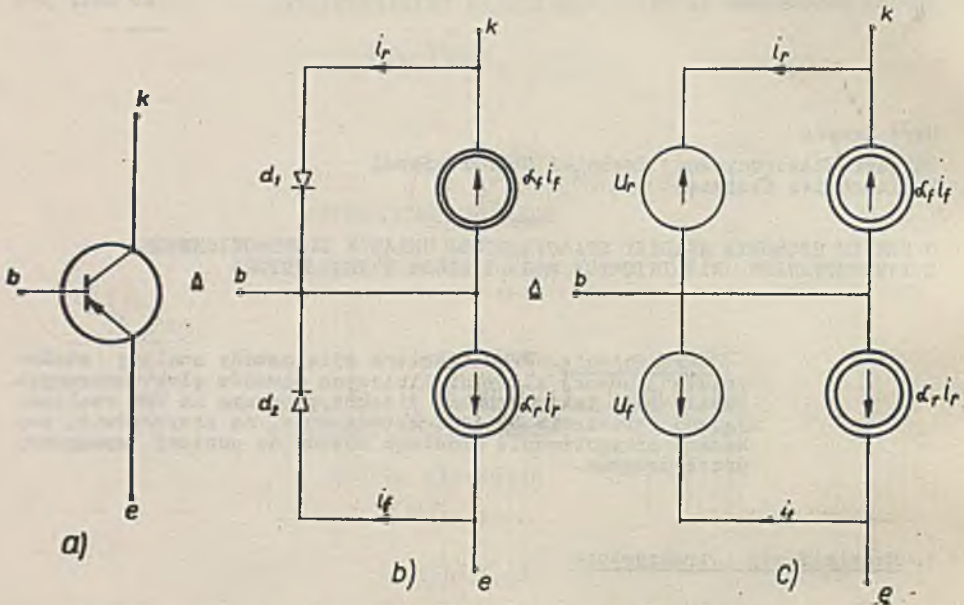
Z powyższego wynika, że diodę w obwodzie można, do celów analizy tego obwodu, zastąpić nieliniowym źródłem napięciowym (lub prądowym) o charakterystyce określonej zależnością (1).

Dla tranzystora można zdefiniować następujący model (stałoprądowy), zwany modelem Eberesa - Mollai i pokazany na rys. 1.

W wypadku tranzystora n-p-n wystarczy zmienić kierunki prądów sił prądomotorycznych i odwrotnie przyłączyć diody d_1 i d_2 .

W ten sposób zamieniając w analizowanym obwodzie tranzystory i diody ich schematami zastępczymi otrzymamy w konsekwencji obwód, który będzie zawierał:

- a) oporności (liniowe);
- b) źródła zasilania (prądowe i napięciowe);
- c) nieliniowe źródła napięciowe (diody i tranzystory);
- d) źródła prądowe sterowane prądowo.



Rys. 1

Nasze dalsze rozważania będą dotyczyły obwodów składających się wyłącznie z takich elementów oraz takich, w których nie występują oczka zawierające same SEM oraz węzły, w których schodzą się gałęzie zawierające same SEM. Prowadzi to bowiem do naruszenia pierwszego i drugiego prawa Kirchhoffa, a przez to do osobliwości macierzy opisujących dany układ.

2. Wyprowadzenie zależności podstawowych

Jeśli przez A (dużymi literami w tym paragrafie oznaczać będziemy macierze i wektory), oznaczymy macierz incydencji grafu będącego obrazem topologicznym rozpatrywanego obwodu, to pierwsze prawo Kirchhoffa możemy zapisać w postaci

$$A I = 0. \quad (2)$$

Można wykazać [1], że macierz incydencji da się przedstawić w postaci

$$\begin{array}{c} w \\ \left[\begin{array}{c|c} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{array} \right] \begin{array}{c} b-w \\ F \end{array} \end{array}, \quad (3)$$

gdzie w jest ilością niezależnych węzłów obwodu, a b ilością jego gałęzi.

Drzewo, które zawiera wszystkie gałęzie z nieliniowymi źródłami napięciowymi, źródłami zasilania oraz (ewentualnie) gałęzie oporowe, nazwiemy drzewem głównym. Macierzowa postać pierwszego prawa Kirchoffa (2) przyjmie teraz postać

$$\begin{bmatrix} I_E \\ I_R \\ I_G \\ I_{Dr} \\ I_{Df} \end{bmatrix} [1 \mid F] = 0, \quad (4)$$

gdzie:

- I_E - prądy w gałęziach drzewa zawierających źródła napięciowe (niesterowane);
- I_R - prądy w oporowych gałęziach drzewa;
- I_G - prądy w oporowych gałęziach dopełnienia;
- I_{Dr} - prądy kolektorowych sterowanych źródeł prądowych modelu;
- I_{Df} - prądy emiterowych sterowanych źródeł prądowych modelu.

Z zależności (4) wynika równość

$$\begin{bmatrix} I_E \\ I_R \end{bmatrix} = -F \begin{bmatrix} I_G \\ I_{Dr} \\ I_{Df} \end{bmatrix} \quad (5)$$

oraz

$$\begin{bmatrix} V_G \\ V_{Dr} \\ V_{Df} \end{bmatrix} = F^T \begin{bmatrix} E \\ V_R \end{bmatrix}. \quad (6)$$

W naszym przypadku mamy następujące związki dodatkowe:

$$V_R = R I_R, \quad (6')$$

gdzie R jest diagonalną macierzą oporów gałęzi drzewa oraz

$$I_G = G V_G, \quad (7)$$

gdzie G jest diagonalną macierzą przewodności gałęzi dopełnienia.

Ponadto

$$I_{Dr} = \alpha_f I_f \quad (8)$$

$$I_{Df} = \alpha_r I_r \quad (9)$$

$$V_{Dr} = U_r \quad (10)$$

$$V_{Df} = U_f, \quad (11)$$

gdzie $\alpha_f, \alpha_r, u_r, u_f, i_r$ oraz i_f są parametrami modelu Ebersa-Molla z rys. 1.

Przepiszmy wzór (5) w postaci

$$\begin{bmatrix} I_f \\ I_r \\ I_{cc} \\ I_R \end{bmatrix} = - \begin{bmatrix} F_{fG} & F_{fDr} & F_{fDf} \\ F_{rG} & F_{rDr} & F_{rDf} \\ F_{cG} & F_{cDr} & F_{cDf} \\ F_{rG} & F_{rDr} & F_{rDf} \end{bmatrix} \cdot \begin{bmatrix} I_G \\ I_{Dr} \\ I_{Df} \end{bmatrix} \quad (12)$$

wtedy wzór (6) przyjmie postać

$$\begin{bmatrix} V_G \\ V_{Dr} \\ V_{Df} \end{bmatrix} = \begin{bmatrix} F_{fG}^T & F_{rG}^T & F_{cG}^T & F_{rG}^T \\ F_{fDr}^T & F_{rDr}^T & F_{cDr}^T & F_{rDr}^T \\ F_{fDf}^T & F_{rDf}^T & F_{cDf}^T & F_{rDf}^T \end{bmatrix} \begin{bmatrix} U_f \\ U_r \\ U_c \\ V_R \end{bmatrix} \quad (13)$$

U_f i U_r są wektorami napięć nieliniowych modelu tranzystora, I_f oraz I_r są wektorami przepływających przez nie prądów, U_c jest wektorem napięć wymuszających (zasilających), natomiast V_R jest wektorem napięć na oporowych gałęziach drzewa.

Wprowadźmy teraz następujące oznaczenia:

$$F_1 = \begin{bmatrix} F_{fG} & F_{fDr} & F_{fDf} \\ F_{rG} & F_{rDr} & F_{rDf} \end{bmatrix} \quad (14)$$

$$F_2 = \begin{bmatrix} F_{fG}^T & F_{rG}^T & F_{cG}^T & F_{rG}^T \\ F_{fDr}^T & F_{rDr}^T & F_{cDr}^T & F_{rDr}^T \end{bmatrix} \quad (15)$$

$$F_3 = \begin{bmatrix} F_{rG} & F_{rDr} & F_{rDf} \end{bmatrix} \quad (16)$$

oraz oznaczymy przez:

- g - ilość oporowych gałęzi dopełnienia,
- d - ilość gałęzi drzewa,
- l - ilość oporowych gałęzi drzewa,
- z - ilość źródeł zasilania.

Wielkości te spełniają dla drzewa głównego oczywisty związek

$$d-l = 2nt + z, \quad (17)$$

gdzie nt jest liczbą tranzystorów w rozważanym obwodzie.

Korzystając ze związków (12)-(16), możemy napisać

$$\begin{aligned} \begin{bmatrix} I_f \\ I_r \end{bmatrix} = -F_1 \begin{bmatrix} I_G \\ I_{Dr} \\ I_{Df} \end{bmatrix} &= -F_1 \begin{bmatrix} G & V_G \\ \alpha_f I_f \\ \alpha_r I_r \end{bmatrix} = -F_1 \begin{bmatrix} G & 0 & 0 \\ 0 & \alpha_f & 0 \\ 0 & 0 & \alpha_r \end{bmatrix} \begin{bmatrix} V_G \\ I_f \\ I_r \end{bmatrix} = -B \begin{bmatrix} V_G \\ I_f \\ I_r \end{bmatrix} = \\ &= -[B_1 \mid B_2] \begin{bmatrix} V_G \\ I_f \\ I_r \end{bmatrix} = -B_1 V_G - B_2 \begin{bmatrix} I_f \\ I_r \end{bmatrix}. \end{aligned} \quad (18)$$

Po uporządkowaniu otrzymujemy zależność

$$(1 + B_2) \begin{bmatrix} I_f \\ I_r \end{bmatrix} = -B_1 V_G. \quad (19)$$

Korzystając z (13) możemy napisać

$$\begin{aligned} V_G = P_2 \begin{bmatrix} U_f \\ U_r \\ U_c \\ V_R \end{bmatrix} &= [C_1 \mid C_2] \begin{bmatrix} U_f \\ U_r \\ U_c \\ V_R \end{bmatrix} = C_1 \begin{bmatrix} U_f \\ U_r \\ U_c \end{bmatrix} + C_2 V_R = \\ &= C_1 \begin{bmatrix} U_f \\ U_r \\ U_c \end{bmatrix} + C_2 R I_R. \end{aligned} \quad (20)$$

Wprowadzając teraz oznaczenie

$$P = \begin{bmatrix} G & 0 & 0 \\ 0 & \alpha_f & 0 \\ 0 & 0 & \alpha_r \end{bmatrix}$$

i korzystając z (12) otrzymujemy

$$\begin{aligned} I_R = -F_3 \begin{bmatrix} I_G \\ I_{Dr} \\ I_{Df} \end{bmatrix} &= -F_3 P \begin{bmatrix} V_G \\ I_f \\ I_r \end{bmatrix} = -D \begin{bmatrix} V_G \\ I_f \\ I_r \end{bmatrix} = - [D_1 \mid D_2] \begin{bmatrix} V_G \\ I_f \\ I_r \end{bmatrix} = \\ &= -D_1 V_G - D_2 \begin{bmatrix} I_f \\ I_r \end{bmatrix}. \end{aligned} \quad (21)$$

Podstawiając do (21) zależność (20) otrzymujemy związek pomiędzy prądami oporowych gałęzi drzewa a napięciami i prądami źródeł nieliniowych oraz napięciami zasilania w postaci

$$I_R = -(1 + D_1 C_2 R)^{-1} D_1 C_1 \begin{bmatrix} U_f \\ U_r \\ U_c \end{bmatrix} - (1 + D_1 C_2 R)^{-1} D_2 \begin{bmatrix} I_f \\ I_r \end{bmatrix}. \quad (22)$$

Robiąc podobne przekształcenia dla równań (20) i (22) otrzymujemy związek

$$\begin{aligned} V_G &= (1 - C_2 R (1 + D_1 C_2 R)^{-1} D_1) C_1 \begin{bmatrix} U_f \\ U_r \\ U_c \end{bmatrix} - \\ &= C_2 R (1 + D_1 C_2 R)^{-1} D_2 \begin{bmatrix} I_f \\ I_r \end{bmatrix}. \end{aligned} \quad (23)$$

Jeśli na koniec wstawimy zależność (23) do równania (19), to po uporządkowaniu i prostych przekształceniach otrzymamy zależność:

$$\begin{bmatrix} I_f \\ I_r \end{bmatrix} = - (1 + B_2 - B_1 C_2 R (1 + D_1 C_2 R)^{-1} D_2)^{-1} B_1 (1 - C_2 R (1 + D_1 C_2 R)^{-1} D_1) C_1 \begin{bmatrix} U_f \\ U_r \\ U_c \end{bmatrix}. \quad (24)$$

Jest to poszukiwany przez nas związek pomiędzy prądami nieliniowych źródeł napięcia oraz ich napięciami i napięciami zasilania (wymuszenia). Zależność tę można napisać także w postaci

$$\begin{bmatrix} I_f \\ I_r \end{bmatrix} = Y \begin{bmatrix} U_f \\ U_r \\ U_c \end{bmatrix} = Y_A \begin{bmatrix} U_f \\ U_r \end{bmatrix} + Y_B U_c. \quad (25)$$

Podstawiając teraz w wyrażeniu (25) za prądy I_f oraz I_r ich wartości (1) w funkcji napięcia, otrzymujemy nieliniowy układ równań, który ogólnie można zapisać jako

$$F_n \begin{pmatrix} U_f \\ U_r \end{pmatrix} - Y_A \begin{bmatrix} U_f \\ U_r \end{bmatrix} - Y_B U_C = 0, \quad (26)$$

którego jacobian definiowany jest przez wyrażenie

$$J_{i,k} = -y_{A_{i,k}} + a_{T_{i0}} e^{a_{T_{i1}} U_i} \delta_{i,k}, \quad (27)$$

gdzie $\delta_{i,k}$ jest deltą Kroneckera.

Rozwiązując powyższy układ równań metodą Newtona wyznaczmy wektory napięć U_f oraz U_r , następnie z zależności (1) obliczamy wartości prądów w wektorach I_f oraz I_r . Korzystając teraz z zależności (22) i (23), możemy wyznaczyć napięcia na oporowych gałęziach dopełnienia V_G oraz prądy w oporowych gałęziach drzewa I_R , a w konsekwencji prądy I_G oraz napięcia V_R . Wprowadzone (do obliczeń maszynowych) poszerzenie problemu o prądy diod oraz SPM wymuszenia nie zmienia zasadniczo powyższych wywodów, gdyż elementy te stanowią tylko dalsze składniki lewej strony wyrażenia (25) (prądy diod) oraz rozszerzenie wektora wymuszeń U_C .

3. Opis programu

Program analizujący obwody spełniające opisane wyżej warunki został napisany dla maszyny cyfrowej Odra 1204 w języku ALGOL 1204 (patrz Dodatek A). Opiszemy teraz poszczególne stadia przygotowania analizowanego obwodu i danych w postaci wymaganej przez program.

3.1. Przygotowanie danych

Pierwszą czynnością jest zastąpienie wszystkich diod występujących w obwodzie, nieliniowymi źródłami napięciowymi i wszystkich tranzystorów schematami zastępczymi jak na rys. 2c. Można oczywiście stosować także bardziej rozbudowany model tranzystora, poszerzony o opory r_b , r_c , r_e odpowiednio w bazie kolektorze i emiterze, gdyż zmieni to tylko topologię schematu zastępczego analizowanego obwodu. Na tak przygotowanym schemacie nanosimy strzałki napięć i prądów, z tym, że napięcie strzałkujemy zawsze przeciwnie do kierunku prądu (w SEM - również!).

Oznaczamy teraz węzły obwodu, przy czym węzłem jest połączenie dowolnych dwu elementów obwodu, gdyż każdy element w programie traktowany jest jako osobna gałąź.

Następnie wybieramy drzewo grafu obwodu, pamiętając o tym, że musi to być drzewo, w skład którego wchodzi wszystkie gałęzie zawierające nieliniowe SEM diod i tranzystorów oraz napięciowe źródła zasilania (wymuszenia). W dalszym ciągu będziemy takie drzewo nazywali drzewem głównym. W wypadku niemożności utworzenia takiego drzewa można do obwodu wprowadzić dodatkowe gałęzie zawierające odpowiednio (wielkością) dobrane opory. Teraz numerujemy gałęzie obwodu poczynając od numeru 1, przestrzegając przy tym następującej kolejności:

- nieliniowe SEM tranzystorów	u_r i u_f
- nieliniowe SEM diod	u_d
- oporowe gałęzie drzewa	R
- oporowe gałęzie dopełnienia	G
- sterowane SPM o współczynniku przenoszenia	α_r
- sterowane SPM o współczynniku przenoszenia	α_f .

Jeżeli w obwodzie znajduje się więcej niż jeden tranzystor, to najlepiej ponumerować je także, gdyż kolejność numeracji gałęzi schematu zastępczego zawsze musi być ta sama, tzn. numer j -tego parametru s -tego tranzystora różni się zawsze o $s-1$ od j -tego parametru tranzystora pierwszego. Kolejność numeracji tranzystorów jest oczywiście dowolna.

Jeśli oznaczymy teraz przez

- w - ilość niezależnych węzłów obwodu,
- nt - ilość tranzystorów w obwodzie,
- g - ilość oporowych gałęzi dopełnienia,
- z - ilość niesterowanych SEM,
- ncs - ilość niesterowanych SPM,

to możemy przygotować dane w następującej kolejności:

$w, nt, g, z, ncs.$

Następną grupę danych stanowi odpowiednio zakodowana macierz incydencji grafu obwodu. Koduje się ją w ten sposób, że dla każdego węzła (wiersza macierzy incydencji), podaje się numery schodzących się w tym węźle gałęzi, przy czym numer opatrujemy znakiem plus, gdy prąd rozważanej gałęzi wypływa z węzła i znakiem minus, gdy prąd do węzła dopływa. Należy tu wszakże przestrzegać następującej reguły: jako i -ty wiersz macierzy incydencji wybieramy taki, w którym występuje gałąź o numerze 1. Ograniczenie to wynika stąd, że program tworząc macierz incydencji w postaci (3) nie przedstawia kolumn tej macierzy. Jeśli jednak tylko drzewo główne istnieje, to zawsze można dane w tej kolejności napisać. Każdy wiersz kończymy znakiem *. Przed napisaniem tego znaku można jeszcze dany wiersz poprawić jeśli popełniliśmy błąd w zapisie (zły kierunek prądu). Pomyłek związanych z napisaniem złego numeru gałęzi tak korygować się nie da. I tak, jeśli np. czwarty wiersz macierzy incydencji zakodowany wygląda tak

5,-4,7,12,-12*,

to oznacza to, że poprawiono błąd w kodowaniu kierunku prądu w gałęzi 12 (ważny jest ostatni zapis, tzn. prąd w gałęzi 12 płynie do węzła IV).

Po zakodowaniu w ten sposób wszystkich w niezależnych węzłów obwodu piszemy następną daną, którą jest temperatura otoczenia. Chodzi tu o uwzględnienie wpływu temperatury na prądy zerowe diod i tranzystorów. Program automatycznie aktualizuje te wartości, przyjmując, że podane w danych prądy zerowe odnoszą się do temperatury 20 deg. Po temperaturze piszemy następującą sekwencję danych:

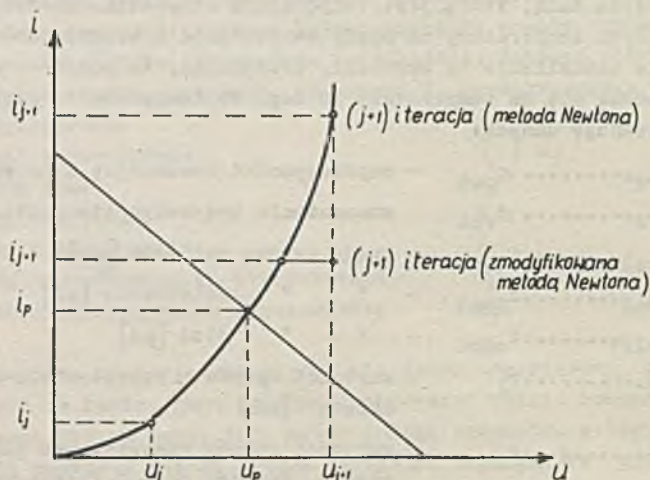
$\alpha_{r1}, \alpha_{r2}, \dots, \alpha_{rnt}$	- współczynniki rewersyjny i normalny
$\alpha_{f1}, \alpha_{f2}, \dots, \alpha_{fnt}$	wzmocnienia prądowego tranzystorów
$i_{0e1}, i_{0e2}, \dots, i_{0ent}$	- prądy zerowe emiterów [μA]
$i_{0c1}, i_{0c2}, \dots, i_{0cnt}$	- " " kolektorów [μA]
$i_{0d1}, i_{0d2}, \dots, i_{0dnd}$	- " " diod [μA]
$0, r_1, r_2, \dots, r_1$	- wartości oporów w rezystywnych gałęziach drzewa [$k\Omega$]
$0, g_1, g_2, \dots, g_g$	- wartości oporów rezystywnych gałęzi dopełnienia. Wartość $g_1 = 0$ traktowana jest przez program jako oznaczenie gałęzi bezprądowej,
$u_{r1}, u_{r2}, \dots, u_{rnt}$	- przybliżenia początkowe wyznaczanych iteracyjnie wartości nieliniowych SEM [V]
$u_{f1}, u_{f2}, \dots, u_{fnt}$	
$u_{d1}, u_{d2}, \dots, u_{dnd}$	
$u_{c1}, u_{c2}, \dots, u_{cz}$	- wartości SEM zasilania (wymuszenia)
i_1, i_2, \dots, i_{ncs}	- " SPM " "

3.2. Działanie programu

Po wprowadzeniu tak przygotowanych danych program tworzy nieliniowy układ równań wg algorytmu podanego w p. 2, który to układ rozwiązuje następnie metodą Newtona. Metoda ta została nieco zmodyfikowana dla potrzeb opisywanego programu.

Po pierwsze - układ zlinearyzowanych równań jest rozwiązywany metodą ortogonalizacji. Polega ona na przekształceniu macierzy współczynników w macierz ortogonalną wierszami [3]. Rozwiązanie uzyskuje się przez pomnożenie transponowanej macierzy ortogonalnej przez kolumnę wyrazów wolnych (które też są przekształcane w procesie ortogonalizacji macierzy). Sposób ten ma tę zaletę, że w przypadku układów sprzecznych wyznacza zbiór równań niezależnych, dla których rozwiązanie uzyskuje się identycznie. Utrzymująca się przez kilka iteracji sprzeczność układu jest sygnałem, że rozwiązania należy szukać z nowego przybliżenia początkowego.

Po drugie - został zmodyfikowany proces obliczania przyrostu prądu. Istota tej modyfikacji [1], w przypadku jednego elementu nieliniowego wyjaśniona jest na rys. 2.



Rys. 2

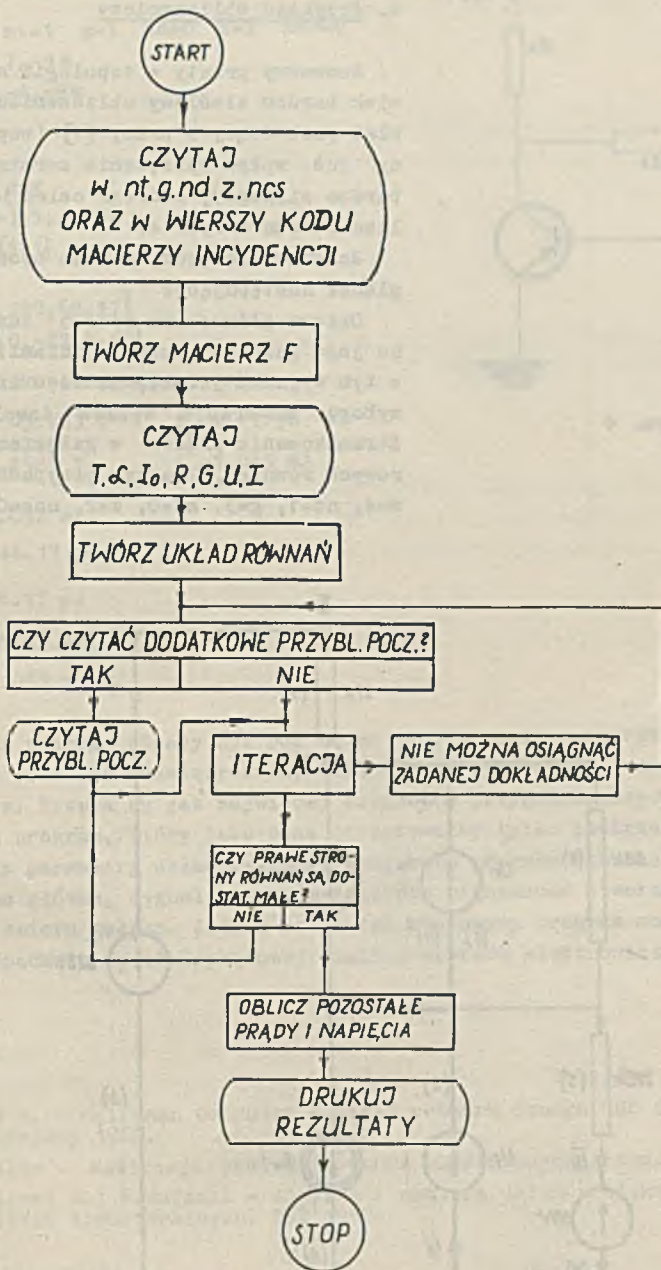
Po trzecie - zostały wprowadzone pewne ograniczenia na obliczany przyrost napięcia, aby uniknąć nadmiaru zmiennoprzecinkowego w maszynie (obliczanie wartości e^x dla dużych wykładników). Ograniczenia te są związane z dużą stromością wykładniczej charakterystyki diody w zakresie przewodzenia.

Korzyść z omówionych modyfikacji uwidacznia się szczególnie przy analizie obwodów z silnymi sprzężeniami zwrotnymi. Występują tam wtedy bardzo silne tendencje do oscylacji iterowanych napięć i rozwiązanie takiego problemu metodą Newtona bez rzeczonych modyfikacji albo jest w ogóle niemożliwe (nadmiar), albo trwa dłużej (wolniejsza zbieżność do rozwiązania).

Oczywistą jest również rzeczą, że rozwiązanie uzyskuje się tym pewniej i łatwiej, im lepsze jest przybliżenie początkowe, ale znalezienie dobrego przybliżenia początkowego nie zawsze jest łatwe.

Celem kontroli procesu iteracyjnego, w programie istnieje możliwość wydruku parametrów charakteryzujących zbieżność procesu, a mianowicie pierwiastka sumy kwadratów wartości prawych stron równań nieliniowych oraz pierwiastka sumy kwadratów przyrostów iterowanych zmiennych.

Po uzyskaniu odpowiednio małej (zadanej) sumy kwadratów prawych stron, program przechodzi do wyznaczania pozostałych wielkości, po czym następuje wydruk wszystkich napięć [V] na elementach nieliniowych, prądów płynących przez te elementy, prądów oporowych gałęzi dopełnienia [mA] i napięć oporowych gałęzi drzewa [V]. Uproszczony schemat blokowy programu przedstawiony jest na rys. 3.



Ryb. 3

Dane do programu mogą więc wyglądać np. następująco:

w=4 nt=1 g=3 nd=0 z=2 ncs=0

1,-6,7,-9*

2,-3,-4,-8*

3,7*

4,5*

T=20 deg

alfa=[.5, .99]

i0= [1,1]

R=0

G=[0,220,68,17]

U=[0,0,-22.5,10].

W wyniku otrzymujemy:

$u_r = -3.783 \text{ V}$ $i_r = -1.98 \text{ } \mu\text{A}$

$u_f = 0.1583 \text{ V}$ $i_f = 1.045 \text{ mA}$

$i_3 = 1.092 \text{ mA}$

$i_4 = -46.17 \text{ } \mu\text{A}$

$i_5 = 46.17 \text{ } \mu\text{A}$

$i_6 = 55.63 \text{ } \mu\text{A}$

$i_7 = 1.092 \text{ mA}$.

Ponieważ program pisany był pod kątem wypróbowania algorytmu układania równań oraz ich rozwiązywania, czynności przygotowania danych są tu dość żmudne. Trzeba by jak najwięcej czynności przygotowawczych przerzucić na sam program, który jako dane otrzymywałby tylko powiązania topologiczne oraz parametry układu. Przede wszystkim program sam musiałby tworzyć drzewo główne, sygnalizując ewentualnie niemożność utworzenia odpowiedniego zbioru gałęzi. Jednak nawet tak zbudowany program może służyć w wielu przypadkach do stałoprądowej analizy obwodów elektronicznych.

LITERATURA

1. Donald A. - Callahan Computer - aided network design. Mc Graw - Hill Book Company 1968.
2. W.N. Ilin - Maszynowe projektowanie elektronicznych schem.
3. S. Bellert H., Woźniacki - Analiza i synteza układów elektrycznych metodą liczb strukturalnych. WNT 1968.

ОБ ОДНОМ МЕТОДЕ АНАЛИЗА НА ПОСТОЯННОМ ТОКЕ ЭЛЕКТРОННЫХ ЦЕПЕЙ
С ПРИМЕНЕНИЕМ НЕЛИНЕЙНЫХ МОДЕЛЕЙ ДИОДА И ТРАНЗИСТОРА

Резюме

В статье подается метод анализа на постоянном токе одного класса электронных цепей. К работе принадлежат также программа на ЕВМ, работающая по поданной методике.

DC - ANALYSIS METHOD OF ELECTRONIC CIRCUITS BY USE THE
NONLINEAR DIODE AND TRANSISTOR MODELS

Summary

The paper deals with DC - analysis of nonlinear electronic circuits. Realization of this method on computer and the preparation of input data for computer program are described.

DODATEK A

```

begin
  comment program analizy ukladow tranzystorowo-diodowo-oporowych z nieli-
    niowym modelem ebbersa-molla;
  boolean bol;
  integer d,n,g,nd,z,l,ncs,n1,i,j,k,n2,p1;
  bol:=true;
  if bol
    then read(d,n,g,nd,z,ncs);
  n1:=n+n;
  l:=d-(n1+nd+z);
  p1:=d+g+n1+ncs;
  n2:=n1+nd;
  begin
    array f [1:d,1:p1+1],m [1:d+g,1:d+g],m1 [1:d+g,1:d-l+ncs+1],
      alfa [0:n1], alte1 [1:n2],
      g1 [0:g],r [0:l],io [1:n2],ei [1:n2+z+ncs],ip [1:d+g];
    real ae,t,s1,s2,ae1,e1,eps,delta;
    procedure specmult(n,m,a,i1,j1,b,c,l1,k1);
      value n,m,i1,j1,k1,l1;
      integer n,m,i1,j1,k1,l1;
      array a,b,c;
      begin
        integer j2,i2,k2;
        real s;
        for i:= i1 step 1 until n+i1-1 do
          begin
            i2:=i-i1+1;
            k2:=i2-1+l1;
            for j:=j1 step 1 until m+j1-1 do
              begin
                j2:=j-j1+k1;
                s:=a[i,j];
                c[k2,j2]:=if s gt 0 then b[i2]
                  else if s=0
                    then 0
                    else -b[i2]
              end j
            end i
          end specmult;
    procedure gausred(n,l,m,a,b,ergaus);
      value n,l,m;

```

```

integer n,l,m;
array a,b;
label ergaus;
tape;
procedure matinv(n,a,determ,sing);
value n;
integer n;
real determ;
array a;
label sing;
tape;
procedure matrixmult(m,n,p,a,b,c);
value m,n,p,b;
integer m,n,p;
array a,b,c;
begin
real s;
for i:=1 step 1 until m do
for k:=1 step 1 until p do
begin
s:=0;
for j:=1 step 1 until n do s:=s+a[i,j]×b[j,k];
c[i,k]:=s
end k,i
end matrixmult;
procedure funkcje;
ep6:begin
for i:=1 step 1 until n2 do
begin
s1:=0;
for j:=1 step 1 until n2+z+ncs do s1:=s1+m1[i,j]×ei[j];
ei1:=ei[i];
if ei1 gt 2
then begin
for k:=1 step 1 until n2 do ei[k]:=altei[k]+.5×(ei[k]-
- altei[k]);
go to ep6
end k;
ae1:=if ei1 gt 0 then ae else .5xae;
ip[i]:=io[i]×(exp(ae1×ei1)-1;

```



```

    m[i,n2+1]:=ip[i]+s1
  end i
end funkcje;
procedure jakobian;
begin
  for i:=1 step 1 until n2 do
    begin
      e1:=ei[i];
      copy(n2,m1[i,1],m[i,1]);
      ae1:=if e1 gt 0 then -ae else .5xae;
      m[i,i]:=m1[i,i]-ae1x(ip[i]+io[i])
    end i
  end jakobian;
procedure newton;
begin
  real r,q,altq;
  boolean bol1;
  bol1:=true;
ep1:funkcje;
  q:=0;
  for k:=1 step 1 until n2 do q:=q+m[k,n2+1]xm[k,n2+1];
  if bol1
    then go to ep3;
  r:=0;
  for k:=1 step 1 until n2 do r:=r+(ei[k]-altei[k])x(ei[k]-alte
i[k]);
  if q gt altq
    then begin
      for k:=1 step 1 until n2 do ei[k]:=altei[k]+.5x(ei[k]-
altei
[k]);
      r:=.25xr;
      if r gt delta
        then go to ep1
      end q gt altq;
    format(' -1.234_10^-12');
    if key(7)
      then print('      ',sqrt(r));
    if r lt delta
      then go to ep5;
ep2:format('?-1.234_10^-12');

```

```

if key(7)
  then print(sqrt(q));
  if q lt eps
    then go to ep4;
  for k:=1 step 1 until n2 do altei[k]:=ei[k];
  altq:=q;
  jakobian;
  for k:=1 step 1 until n2 do
    begin
      for i:=1 step 1 until k-1 do
        begin
          q:=0;
          for j:=1 step 1 until n2 do q:=q+m[i,j]xm[k,j];
          for j:=1 step 1 until n2+1 do m[k,j]:=m[k,j]-qxm[i,j]
        end i;
      q:=0;
      for i:=1 step 1 until n2 do q:=q+m[k,i]xm[k,i];
      if q gt eps
        then begin
          q:=1/sqrt(q);
          for i:=1 step 1 until n2+1 do m[k,i]:=qxm[k,i];
        end
      else if key(7)
        then begin
          format(' ?m 12,12 =1.23410-12');
          print(k,n2+1,m[k,n2+1])
        end
    end
  end ortogonalizacji zlinearyzowanego ukladu rownan;
  for k:=1 step 1 until n2 do
    begin
      q:=ei[k];
      for i:=1 step 1 until n2 do q:=q-m[i,k]xm[i,n2+1];
      ei[k]:=if altei[k] ge 0 and q-altei[k] gt .5
        then 1/aexln((ip[k]+io[k])x(1+aex(q-altei[k]))/io[k]
1)
        else q
    end
  end obliczania nowego przyblizenia ei;
  go to ep1;
ep3:bol1:=false;
  go to ep2;
ep4:print('?num.i residua rownan?');
  format (' ?11 -1.23410-12');

```



```

    for k:=1 step 1 until n2 do print(k,m[k,n2+1])
  end newton;
  for i:=1 step 1 until d do
    for j:=1 step 1 until p1 do f[i,j]:=0;
  for i:=1 step 1 until d do
    begin
  l1:f[i,abs(ininteger)]:=sign(lastinteger);
    if lastchar ne 26
      then go to l1
    end i;
  for i:=1 step 1 until d do
    begin
      if f[i,i] lt 0
        then for j:=1 step 1 until p1 do f[i,j]:=-f[i,j];
      for j:=1 step 1 until d do
        if j ne 1 and f[j,i] ne 0
          then begin
              ae:=sign(f[j,i]);
              for k:=1 step 1 until p1 do f[j,k]:=f[i,k]-aexf[j,k]
            end;
          end i;
  for i:=1 step 1 until d do
    if f[i,i] lt 0
      then for k:=1 step 1 until p1 do f[i,k]:=-f[i,k];
      for i:=1 step 1 until d+g do
        begin
          for j:=1 step 1 until d+g do m[i,j]:=if i=j then 1 else 0;
          for j:=1 step 1 until d-1+ncs do m1[i,j]:=0
        end i;
      if bol
        then read(t,alfa,io,r,g1,ei);
        ae:=16020/(1.38x(273.16+t));
        s2:=exp(.07x(t-20))x.001;
  for i:=1 step 1 until n+nd do
    if i le n
      then begin
          io[i]:=io[i]x s2/(1-alfa[i]xalfa[i+n]);
          io[i+n]:=io[i+n]xs2/(1-alfa[i]xalfa[i+n])
        end i le n
      else io[i+n]:=io[i+n]xs2;
    for i:=1 step 1 until g do g1[i]:=if g1[i]=0 then 0 else 1/g1[i];

```

```

  for i:=1 step 1 until n1 do
  begin
    for j:=1 step 1 until n1 do m[i,j]:=m[i,j]+sign(f[i,j+d+g])*x
                                     alfa[j];

    copy(g,f[i,d+1],m[i,d-1+1]);
    copy(ncs,f[i,d+g+n1+1],m1[i,d-1+1])
  end i;
  for i:=1+n1 step 1 until nd+z+n1 do
  begin
    copy(n1,f[i,d+g+1],m[i,1]);
    copy(g,f[i,d+1],m[i,d-1+1]);
    copy(ncs,f[i,d+g+n1+1],m1[i,d-1+1])
  end i;
  specmult(l,n1,f,d-1+1.d+g+1,r,m,d-1+g+1,1);
  specmult(l,g,f,d-1+1,d+1,r,m,d-1+g+1,d-1+1);
  specmult(l,ncs,f,d-1+1,d+g+n1+1,r,m1,d-1+g+1,d-1+1);
  for i:=d+1 step 1 until d+g do
  begin
    for j:=d-1+1 step 1 until d do f[i-d,j-d+1]:=-f[j,i];
  for j:=1 step 1 until n2+z do f[i-d,j+1]:=-f[j,i]
  end i;
  specmult(g,l,f,1,1,g1,m,d-1+1,d-1+g+1);
  specmult(g,m2+z,f,1,1+1,g1,m1,n2+1+z,1);
  gauered(d+g,d-1+ncs,n2,m,m1,ergaus);
  matinv(n2,m,s1,sing);
  matrixmult(n2,n2,d-1+ncs,m,m1,m1);
  setinput(0);
  setoutput(0);
  ep5:format('?.23410-12');
  if key(8)
    then for i:=1 step 1 until n2 do print(ei[i]);
  print(' ?eps=');
  read(eps);
  print('delta=');
  read(delta);
  if key(9)
  then for i:=1 step 1 until n2 do read(ei[i]);
  newton;
  for i:=n2+1 step 1 until d+g do

```



```
begin
  s1:=0;
  for j:=1 step 1 until d-1+ncs do s1:=s1+m1[i,j]xei[j];
  for j:=1 step 1 until i-1 do s1:=s1+m[i,j]xip[j];
  ip i :=-s1
  end i;
  format('?-1.234,10-12');
  for i:=1 step 1 until n2 do print ('?wektor ei',ei[i]);
  for i:=1 step 1 until d+g do print(ip[i]);
sing:end;
ergaus:end?
```