

POLITECHNIKA ŚLĄSKA  
WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI

# PODSTAWY PROGRAMOWANIA KOMPUTERÓW

PROGRAM DO ANALIZY LINIOWYCH OBWODÓW PRĄDU STAŁEGO

Jacek Wieczorek

1 listopada 2020

---

autor	Jacek Wieczorek
prowadzący	Wojciech Łabaj
rok akademicki	2020/2021
kierunek	informatyka
rodzaj studiów	SSI
semestr	1
termin laboratorium	poniedziałek 11:00–12:15, środa 8:00–10:15
sekcja	12
termin oddania sprawozdania	2020-11-01
repozytorium	<a href="https://github.com/polsl-aei-ppk/ed7a9f7a-gr12-repo">https://github.com/polsl-aei-ppk/ed7a9f7a-gr12-repo</a>

---

## 1 TEMAT

---

Tematem zadania programistycznego było stworzenie programu umożliwiającego analizę liniowych obwodów prądu stałego złożonych z rezystorów, sił elektromotorycznych i prądomotorycznych. Program powinien wczytywać z pliku listę elementów i łączonych przez nie węzłów układu, a następnie dla każdego elementu wyznaczyć spadek napięcia, natężenie prądu i traconą na nim moc. Dodatkowo powinien zostać sporządzony bilans mocy układu. W przypadku niedostarczenia żadnych argumentów, program powinien wypisać krótką instrukcję obsługi.

## 2 ANALIZA PROBLEMU

---

Analiza liniowych obwodów elektrycznych sprowadza się (niejako z definicji) do rozwiązywania układów równań liniowych. Jednak przed rozwiązaniem układu równań należy go najpierw sformułować. W tym celu zastosowany został algorytm zmodyfikowanej analizy potencjałów węzłowych MNA[1] (ang. Modified Nodal Analysis).

### 2.1 ALGORYTM MNA

Algorytm MNA pozwala na sformułowanie układu równań liniowych opisujących działanie układu w postaci równania macierzowego. Algorytm uwzględnia układy składające się nie tylko z rezystancji, a także z elementów reaktywnych. Dodatkowo pozwala na analizę układów z idealnymi wzmacniaczami operacyjnymi<sup>1</sup>.

Wybrany algorytm pozwala na analizę układów znacznie bardziej skomplikowanych niż te określone w poleceniu. Pomimo tego zdecydowano się na wykorzystanie go ze względu na osobiste zainteresowania autora, który od dawna chciał stworzyć program do analizy odpowiedzi częstotliwościowej układu. Należy tutaj zaznaczyć, że ta decyzja nie wpłynęła znacząco na złożoność implementacji.

Dla układu o  $n$  węzłach i  $m$  siłach elektromotorycznych, wynikiem działania algorytmu jest układ równań w postaci macierzowej:

$$\mathbf{Ax} = \mathbf{z}$$

gdzie  $\mathbf{A}$  to macierz o wymiarach  $(n+m) \times (n+m)$ , a  $\mathbf{z}$  to wektor  $(n+m) \times 1$ . Uwzględnia się w nich admitancje międzywęzłowe oraz wartości SEM i SPM. Szukaną w tym równaniu jest wektor  $\mathbf{x}$   $((n+m) \times 1)$  zawierający potencjały węzłowe i prądy pobierane z sił elektromotorycznych. Tak zbudowane równanie zapisywane jest w postaci jednej macierzy jako:

$$[\mathbf{A} \quad \mathbf{z}]$$

a następnie jest rozwiązywane algorytmem eliminacji Gaussa, opisanym w następnej sekcji.

### 2.2 ALGORYTM ELIMINACJI GAUSSA

Algorytm eliminacji Gaussa pozwala na rozwiązywanie układów równań liniowych zapisanych w formie macierzy  $n \times (n+1)$ . Pierwsze  $n$  kolumn jest zbudowane ze współczynników układu równań. Ostatnia „doklejona” kolumna zawiera wyrazy wolne równań.

Algorytm opiera się na przekształceniach równoważnych równań — obustronnym mnożeniu przez stałą, dodawaniu stronami i zamianie ich kolejności w macierzy. Operacje te są wykonywane

---

<sup>1</sup>Przy założeniu, że pracują z ujemnym sprzężeniem zwrotnym

tak, by redukować kolejne współczynniki w równaniach i sprowadzić główną część macierzy wejściowej do postaci górnotrójkątnej.

Po redukcji macierzy do postaci trójkątnej możliwe jest łatwe wyznaczenie wartości szukanych zmiennych. Począwszy od dołu, każde kolejne równanie zawiera tylko jedną niewiadomą więcej. Rozwiązanie równania w danym wierszu pozwala na podstawienie zmiennych do równania w wierszu powyżej, a następnie jego rozwiązanie.

## 2.3 MACIERZ JAKO STRUKTURA DANYCH

Opisane powyżej algorytmy opierają się głównie o operacje na macierzach. W celu ułatwienia ich implementacji zdecydowano się na stworzenie własnej generycznej klasy macierzy. Konieczne jest by macierz mogła przyjąć dowolne wymiary, ponieważ zależą one od topologii analizowanego układu. Logika stojąca za tą strukturą danych zasadniczo ogranicza się do zamiany numerów wierszy i kolumn na indeksy pól w podlegającym jej wektorze.

## 3 SPECYFIKACJA ZEWNĘTRZNA

---

W związku z zainteresowaniami autora i chęcią implementacji programu pozwalającego także na symulację obwodów prądu zmiennego, program został stworzony w dwóch wersjach — podstawowej (legacy) i rozszerzonej (extended). Wyboru wersji dokonuje się przy kompilacji z użyciem CMake — odpowiada za to opcja `EXTENDED`, domyślnie ustawiona na `OFF`.

### 3.1 WERSJA PODSTAWOWA

Wersja podstawowa programu przyjmuje jeden lub dwa argumenty — odpowiednio: plik wejściowy i wyjściowy (opcjonalny). W przypadku niedostarczenia nazwy pliku wyjściowego, wyniki są wypisywane na standardowy strumień wyjścia. Jeśli nie został dostarczony żaden argument, program wypisuje krótką informację dot. sposobu użycia.

Format pliku wejściowego został zdefiniowany w treści zadania. Każda linia zawiera opis jednego elementu - typ (określony literą R, I lub E), węzeł początkowy, węzeł końcowy i wartość. Węzły numerowane są kolejnymi liczbami naturalnymi.

Niestety treść zadania nie przewiduje sposobu określania węzła odniesienia. Wybór takiego węzła jest jednak konieczny do przeprowadzenia analizy, dlatego za „masę” zawsze przyjmowany jest węzeł nr 1. *Oznacza to, że istnienie węzła nr 1 jest warunkiem koniecznym do przeprowadzenia poprawnej symulacji.*

Wynikiem działania programu jest lista potencjałów węzłowych oraz zestawienie elementów zawierające spadki napięć, prądy i straty mocy na komponentach. Na samym końcu wyświetlany jest bilans mocy układu (całkowita moc tracona na odbiornikach).

### 3.2 WERSJA ROZSZERZONA

Jako że wersja rozszerzona programu nie jest ściśle związana z tematem zadania, jej opis został pominięty w niniejszym sprawozdaniu. Zamiast tego, opis jest dostępny w dokumentacji wygenerowanej przez program Doxygen na stronie „Wersja rozszerzona”.

## 4 SPECYFIKACJA WEWNĘTRZNA

---

Szczegóły implementacji zostały udokumentowane przy pomocy narzędzia Doxygen i są opisane w komentarzach w plikach z kodem źródłowym. Do plików projektu został dołączony także dodatkowy opis w pliku `mypice.dox` i plik konfiguracyjny Doxygen `doxygen.conf`.

## 5 TESTOWANIE

---

Celem wykazania poprawności działania programu, przeprowadzone zostały w nim różne symulacje, a ich wyniki zostały porównane z tymi otrzymanymi z programów takich jak LTSpice[3] czy ngspice[4]. Analiza AC i DC opierają się na tych samych algorytmach i strukturach danych, zatem stosowna jest weryfikacja poprawności działania rdzennej implementacji na wynikach obu analiz. Porównanie wyników zostało zamieszczone w dodatku A z uwagi na ich dużą objętość.

Sprawdzone zostało również zachowanie programu w przypadku dostarczenia niepoprawnych danych wejściowych. Zgodnie z założeniami, wczytanie wadliwego pliku powoduje wypisanie odpowiedniego komunikatu o błędzie. Podobnie jest w przypadku, gdy wejściowy układ nie posiada rozwiązań.

## 6 WNIOSKI

---

Stworzony program pozwala zarówno na analizę punktu pracy układu prądu stałego jak i analizę odpowiedzi częstotliwościowej układu prądu zmiennego. Wyniki dotychczas przeprowadzonych symulacji pokrywały się z wynikami otrzymywanymi z programów takich jak LTSpice czy ngspice.

Największym ograniczeniem programu jest brak wsparcia dla elementów nieliniowych — na przykład diod i tranzystorów. Doskwierający jest także brak analizy stanów przejściowych (transient). Być może problemy te zostaną zaadresowane w następnej wersji w odleglejszej przyszłości.

## LITERATURA

---

- [1] An Algorithm for Modified Nodal Analysis, <https://lpsa.swarthmore.edu/Systems/Electrical/mna/MNA3.html>, dostęp 26.10.2020r.
- [2] Wikipedia — Gaussian elimination, [https://en.wikipedia.org/wiki/Gaussian\\_elimination](https://en.wikipedia.org/wiki/Gaussian_elimination), dostęp 26.10.2020r.
- [3] Analog Devices, LTSpice, <https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html>, dostęp 26.10.2020r.
- [4] Program ngspice, <http://ngspice.sourceforge.net/>, dostęp 26.10.2020r.

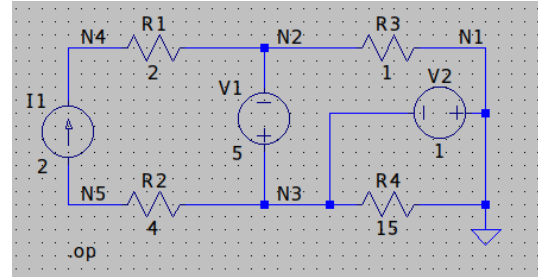
## A PORÓWNANIE WYNIKÓW

### A.1 ANALIZA DC — PRZYKŁAD Z ZADANIA

W treści zadania podany został następujący przykładowy układ:

```
1 R 4 2 2
2 I 5 4 2
3 R 5 3 4
4 E 3 2 -5
5 E 3 1 1
6 R 1 2 1
7 R 3 1 15
```

Listing 1: Netlista opisująca układ



Rysunek 1: Odpowiadający schemat ideowy

Wynik działania programu:

```
1 Potencjaly wezlowe:
2     V(1) = 0 V
3     V(2) = -6 V
4     V(3) = -1 V
5     V(4) = -2 V
6     V(5) = -9 V
7
8 E1 - [3, 2]:
9     V(E1) = -5 V
10    I(E1) = 8 A
11    P(E1) = -40 W
12
13 E2 - [3, 1]:
14     V(E2) = 1 V
15     I(E2) = -6.06667 A
16     P(E2) = -6.06667 W
17
18 I1 - [5, 4]:
19     V(I1) = 7 V
20     I(I1) = -2 A
21     P(I1) = -14 W
22
23 R1 - [4, 2]:
24     V(R1) = -4 V
25     I(R1) = -2 A
26     P(R1) = 8 W
27
28 R2 - [5, 3]:
29     V(R2) = 8 V
30     I(R2) = 2 A
31     P(R2) = 16 W
32
33 R3 - [1, 2]:
34     V(R3) = -6 V
35     I(R3) = -6 A
36     P(R3) = 36 W
37
38 R4 - [3, 1]:
```

```
39      V(R4) = 1 V
40      I(R4) = 0.0666667 A
41      P(R4) = 0.0666667 W
42
43 Moc calkowita: 60.0667 W.
```

Wynik działania programu LTSpice:

```
1      --- Operating Point ---
2
3 V(n2):  -6          voltage
4 V(n4):  -2          voltage
5 V(n3):  -1          voltage
6 V(n5):  -9          voltage
7 I(I1):   2          device_current
8 I(R4):  0.0666667   device_current
9 I(R3):   6          device_current
10 I(R2):   2          device_current
11 I(R1):  -2          device_current
12 I(V2): -6.06667     device_current
13 I(V1): -8          device_current
```

Jak można zauważyć, w obu przypadkach wyznaczone potencjały węzłowe i prądy płynące przez rezystory są takie same.

## A.2 ANALIZA AC — FILTR SALLÉN-KEY

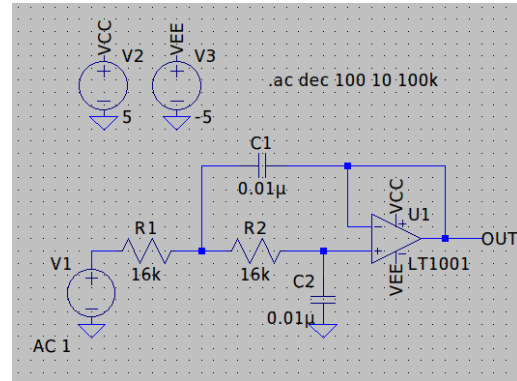
Analiza AC filtru aktywnego II rzędu w topologii Salen-Key. Filtr został zaprojektowany[?], by częstotliwość odcięcia  $f_c$  wynosiła w przybliżeniu 1kHz.

```

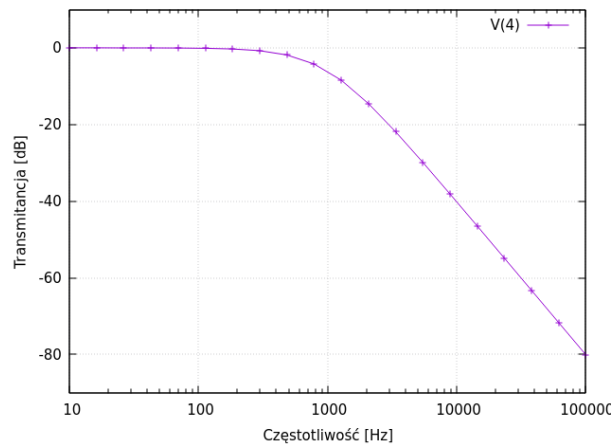
1 Sallen-Key lowpass filter
2
3 OPA1 4 3 4
4 R1 1 2 16k
5 R2 2 3 16k
6 C1 2 4 0.01u
7 C2 3 0 0.01u
8 V1 1 0 0 ac 1
9
10 .ac dec 5 10 100k
11 .print ac V(4)

```

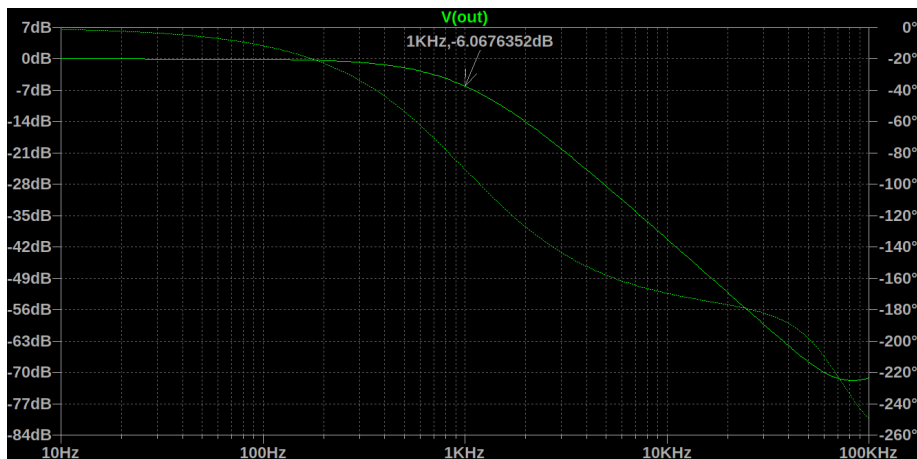
Listing 2: Netlista opisująca filtr



Rysunek 2: Schemat ideowy filtra



(a) Wynik symulacji w myspice



(b) Wynik symulacji w LTSpice

Rysunek 3: Porównanie wyników symulacji z programem LTSpice

Porównując charakterystyki można zauważyć rozbieżność w wynikach jedynie dla wysokich częstotliwości ok. 100kHz. Jest to spowodowane zastosowaniem nieidealnego wzmacniacza LT1001 w symulacji LTSpice.