Karol Pichurski
321663
ISSP 2

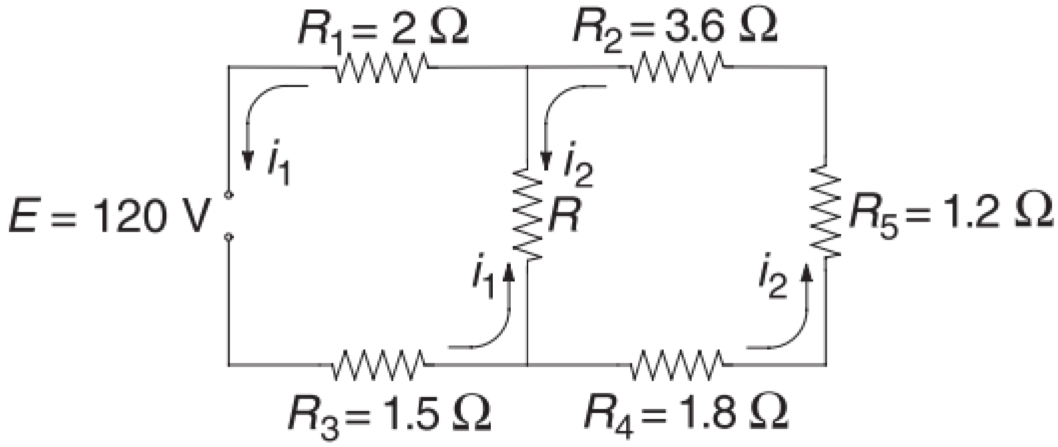
31.01.2022

Metody Numeryczne I

Raport Zadanie 7

TREŚĆ ZADANIA

Dla jakiej wartości oporu R moc wydzielana na nim jest największa?



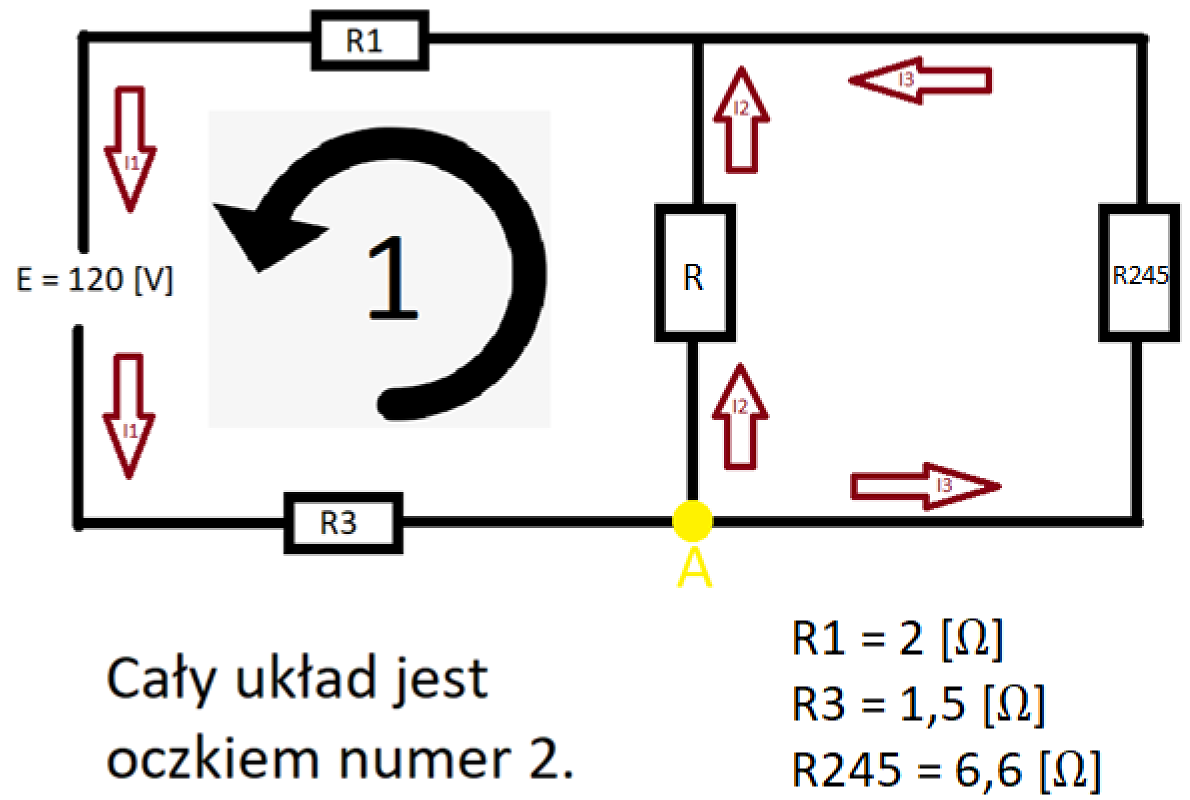
METODOLOGIA ROZWIĄZANIA ZADANIA

W celu rozwiązania zadania będę korzystał z następujących praw i wzorów fizycznych:

- - Pierwsze prawo Kirchhoffa: suma natężeń prądów wpływających jest równa sumie natężeń prądów wypływających z węzła,
- - Drugie prawo Kirchhoffa: suma spadków napięć w oczku jest równa sumie sił elektromotorycznych,
- - Prawo Ohma: proporcjonalność natężenia prądu płynącego przez przewodnik do napięcia panującego między końcami przewodnika można wyrazić za pomocą wzoru U = I * R,
- - Moc elektryczna: praca jaką wykonuje energia elektryczna w jednostce czasu. Jednostką mocy w układzie SI jest wat. Dla prawa Ohma można wyrazić moc elektryczną za pomocą wzoru $P = I^2 * R$,
- Łączenie oporników szeregowo wyrażamy wzorem $R_z = R_1 + R_2 + ... + R_n$;

Na samym początku łączę oporniki R_2, R_4 oraz R_5 w jeden zastępczy opornik R_{245}, który jest sumą wcześniej wymienionych oporników poprzez łączenie szeregowe i jego opór wynosi 6,6 Ω.

Po zastąpieniu oporników przedstawiam poniżej nowy układ z zaznaczonymi węzłami oraz oczkami potrzebnymi do wykorzystania praw Kirchoffa.



Wykorzystując nowy układ z węzłami i oczkami układam następujące równania.

- Dla węzła A: $I_1 - I_2 - I_3 = 0$
- Dla oczka numer 1: $U_{13} + U_R = E$
- Dla oczka numer 2: $U_{13} + U_{245} = E$

Dla tak wyprowadzonych równań tworzę układ równań, równocześnie zmieniając napięcia za iloczyn natężenia i oporu z prawa Ohma.

$$\begin{cases} I_1 - I_2 - I_3 = 0 \\ I_1 R_{13} + I_2 R = E \\ I_1 R_{13} + I_3 R_{245} = E \end{cases}$$

Teraz do układu równań podstawiam znane wartości.

$$\begin{cases} I_1 - I_2 - I_3 = 0 \\ I_1 3{,}5 + I_2 R = 120 \\ I_1 3{,}5 + I_3 6{,}6 = 120 \end{cases}$$

Układ możemy przedstawić za pomocą iloczynu dwóch macierzy i macierzy wynikowej.

$$\begin{bmatrix} 1 & -1 & -1 \\ 3{,}5 & R & 0 \\ 3{,}5 & 0 & 6{,}6 \end{bmatrix} * \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 120 \\ 120 \end{bmatrix}$$

Taką postać równania możemy rozwiązać już numerycznie za pomocą Pythona.

ROZWIĄZANIE ZADANIA

Na samym początku importuję biblioteki potrzebne do wykonania zadania:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.linalg as la
```

- numpy – biblioteka dostarczająca wydajne tablice do obliczeń numerycznych
- matplotlib – biblioteka przeznaczona do rysowania wykresów
- scipy – biblioteka dostarczająca moduł do algebry liniowej

Następnie deklaruje zmienne z danymi potrzebnymi do wykonania zadania:

```
R = np.linspace(0, 5, 10000)
E= np.array([0, 120, 120])
I = []
```

- R – jest tablicą 10 000 punktów równo oddzielonych od siebie w przedziale <0 ; 5> pośród, których będziemy szukać tego wydzielającego maksymalną moc
- E – jest macierzą wynikową z poprzedniego podpunktu raportu
- I – jest tablicą do, której będziemy zapisywać natężenia I zależne od oporu R

Następnie wykonuje obliczenia:

```
for Ri in R:
    matrix = np.array([[1, -1, -1],
                       [3.5, Ri, 0],
                       [3.5, 0, 6.5]])

    I.append(la.solve(matrix, E)[1])

P = [I[i]**2 * R[i] for i in range(len(I))]
```

- pętla for, która przejdzie po wszystkich badanych oporach z naszego zakresu <0 ; 5>, wewnątrz pętli deklaruje macierz z poprzedniego podpunktu raportu i wewnątrz niej wpisuje Ri, które jest naszym badanym oporem
- następnie do naszej tablicy I dodaję wynik, który jest rozwiązaniem układu macierzy, rozwiązanego funkcją solve z biblioteki scipy.linalg
- po wyjściu z pętli tworzę tablicę mocy opornika zależną od danego natężenia i oporu

Następnie szukam maksymalnej mocy i jej oporu:

```
P_max = max(P)
R_max = R[P.index(max(P))]
```

Następnie rysuje wykres zależności mocy od oporu:

```
plt.plot(R, P, R_max, P_max, 'ro')
plt.legend(['P(R)', '$P_{max}$'])
plt.grid()
plt.title('Wykres zależności mocy od oporu P(R)')
plt.xlabel('R [Ω]')
plt.ylabel('P [W]')
plt.show()
```

Na samym końcu wyświetlam wyniki końcowe:

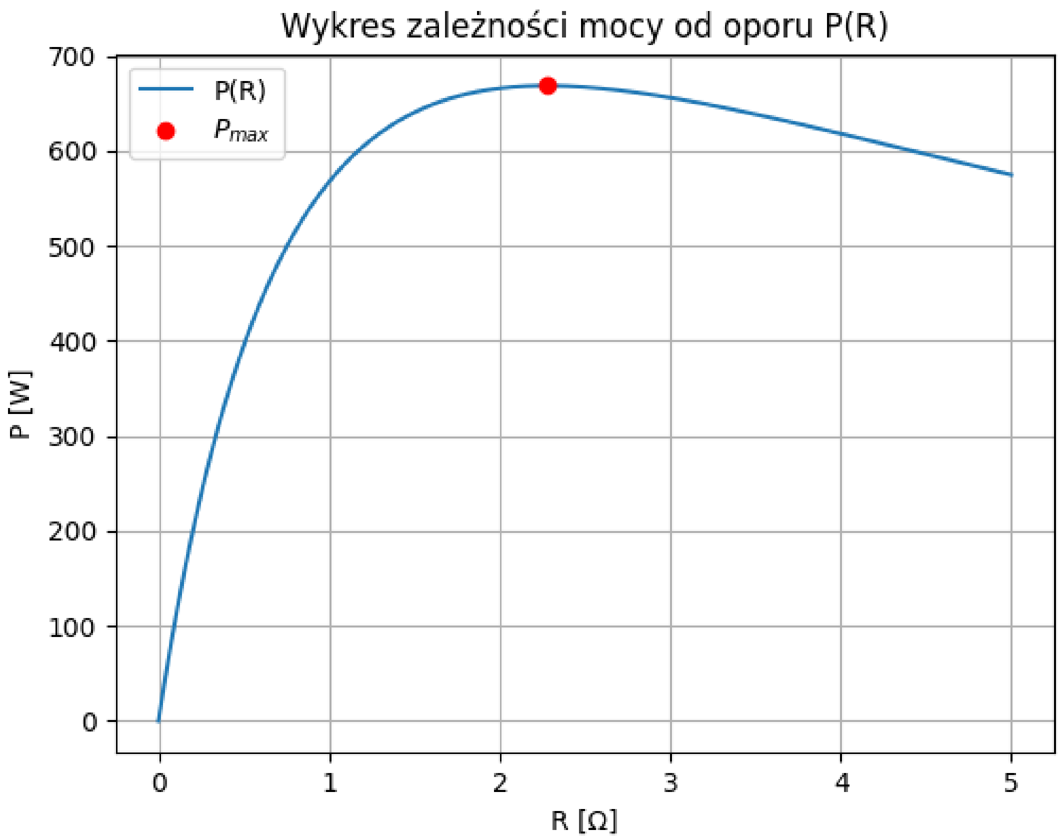
```
print(f'Maksymalna moc wydzielona przez opornik wynosi: {round(P_max, 2)} [W]')
print(f'Wartość oporu dla maksymalnej mocy wynosi: {round(R_max, 2)} [Ω]')
```

WYNIKI KOŃCOWE

Maksymalna moc wydzielona przez opornik wynosi: 668.57 [W]

Wartość oporu dla maksymalnej mocy wynosi: 2.28 [Ω]

WYKRES



KOD PROGRAMU

```python
#Karol Pichurski, Metody Numeryczne 1, Raport Zaliczeniowy Zadanie 7

import numpy as np
import matplotlib.pyplot as plt
import scipy.linalg as la

#dane
R = np.linspace(0, 5, 10000)
E= np.array([0, 120, 120])
I = []

#obliczenia
for Ri in R:
    matrix = np.array([[1, -1, -1],
                       [3.5, Ri, 0],
                       [3.5, 0, 6.5]])

    I.append(la.solve(matrix, E)[1])

P = [I[i]**2 * R[i] for i in range(len(I))]

#wyniki końcowe
P_max = max(P)
R_max = R[P.index(max(P))]

#wykres
plt.plot(R, P, R_max, P_max, 'ro')
plt.legend(['P(R)', '$P_{max}$'])
plt.grid()
plt.title('Wykres zależności mocy od oporu P(R)')
plt.xlabel('R [Ω]')
plt.ylabel('P [W]')
plt.show()

#wyświetlenie wyników końcowych
print(f'Maksymalna moc wydzielona przez opornik wynosi: {round(P_max, 2)} [W]')
print(f'Wartość oporu dla maksymalnej mocy wynosi: {round(R_max, 2)} [Ω]')
```

Kod programu można znaleźć na GitHubie po linkiem:
https://github.com/Khaotich/Metody_Numeryczne_1