



Politechnika Białostocka
Wydział Elektryczny
Katedra Elektrotechniki, Energoelektroniki i Elektroenergetyki

Instrukcja
do pracowni specjalistycznej z przedmiotu

Technologie informacyjne

Kod przedmiotu: **EMS1A1008**

(studia stacjonarne)

MATLAB CZ. 3 OBLICZENIA Z UŻYCIEM SKRYPTÓW I FUNKCJI

Numer ćwiczenia

TI_08

Autor:

dr inż. Adam Steckiewicz

Białystok 2025



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Spis treści

1. Opis stanowiska	3
2. Cel ćwiczenia	3
3. Skrypty	3
3.1. Wprowadzenie.....	3
3.2. Tworzenie kodu	4
3.3. Przydatne funkcje.....	7
4. Przykład rozwiązania zagadnienia elektrotechnicznego	9
5. Przebieg ćwiczenia.....	12
5.1. Analiza przykładowego zadania	12
5.2. Zasilanie lamp samochodowych.....	12
5.3. Prądy i napięcia w obwodzie	13
5.4. Dobór zabezpieczenia elektrowni wodnej.....	14
5.5. Test technologii ładowania samochodu elektrycznego* (dodatkowe) ...	15
6. Literatura.....	17
7. Pytania kontrolne	17
8. Wymagania BHP	18

Materiały przygotowane w ramach projektu „PB 5.0 – dostosowanie oferty dydaktycznej Politechniki Białostockiej do potrzeb nowoczesnej gospodarki oraz zielonej i cyfrowej transformacji” (nr umowy FERS.01.05-IP.08-0327/23-00) w ramach programu Fundusze Europejskie dla Rozwoju Społecznego 2021-2027 współfinansowanego ze środków Europejskiego Funduszu Społecznego Plus.

Publikacja jest udostępniona na licencji Creative Commons - Uznanie autorstwa 4.0 (CC BY 4.0). Pełna treść licencji dostępna na stronie creativecommons.org.

Materiały dydaktyczne przeznaczone dla studentów Wydziału Elektrycznego PB.

© Wydział Elektryczny, Politechnika Białostocka, 2025 (wersja 1.0)

1. Opis stanowiska

Podczas zajęć wykorzystywany jest komputer klasy PC z systemem operacyjnym *Microsoft Windows 10/11*. Na komputerach zainstalowane jest oprogramowanie inżynierskie w postaci środowiska *Matlab R2022b* (lub nowsze) [1], z licencją typu *classroom license*.

2. Cel ćwiczenia

Celem jest poznanie oraz opanowanie wykorzystywania środowiska do tworzenia i zapisywania kodu w postaci tzw. skryptu, służącego do prowadzenia obliczeń i rozwiązywania zagadnień inżynierskich. Celem jest też przećwiczenie zastosowania wybranych funkcji wykonujących działania matematyczne i logiczne.

3. Skrypty

3.1. Wprowadzenie

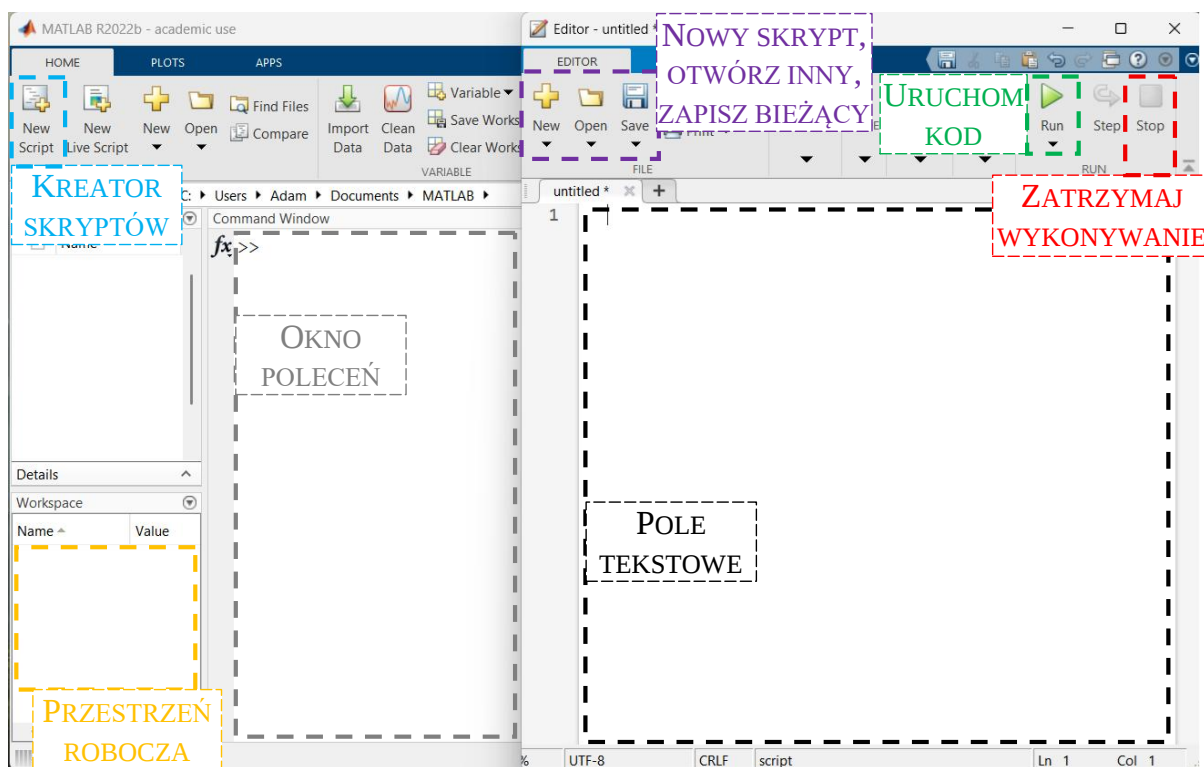
W *Matlab*-ie możliwe jest pisanie swoistych „podprogramów” wykonywanych w programie głównym. Taki „podprogram”, zwany skryptem, zawiera kod, który tworzy się podobnie jak w typowych środowiskach programistycznych w rodzaju *Visual Studio* (język C), *PyCharm* (język Python), czy *Android Studio* (platforma Android). Różnicą jest jednak to, że kod napisany w *Matlab*-ie musi być również w nim wykonywany – bez tego środowiska skrypt sam się nie uruchomi. Ma to swoje zalety i wady. Plusem jest gotowy pakiet zapewniający pełną obsługę kodu (pisanie, edytowanie, formatowanie, testowanie, debugowanie, uruchamianie, itp.) oraz możliwość wykorzystywania wszystkich funkcji, poleceń i innych elementów (np. modeli symulacyjnych) zaimplementowanych przez twórców *Matlab*-a i samego użytkownika. Wadą zaś niemożność uruchomienia skryptu bez zainstalowania i używania programu *Matlab* (albo *Octave* lub *Scilab*) i ograniczone możliwości użycia podprogramu, np. do zaprogramowania układu scalonego.

Oprogramowanie wykorzystuje język wysokiego poziomu (uproszczenia), co daje możliwość szybkiego i relatywnie prostego pisania poleceń. Oczywiście ten

sam język i polecenia wykorzystuje się w skryptach. Ale do czego same skrypty najlepiej jest używać? Otóż głównym ich przeznaczeniem jest realizowanie krok po kroku kolejnych działań (instrukcji), najczęściej związanych z obliczeniami i obróbką wyników. Na przykład można utworzyć kod programu, który po uruchomieniu sam wczyta dane, dokona zadanych obliczeń, porówna wyniki wedle ustalonych kryteriów, a na koniec wyświetli rezultaty w **oknie poleceń**.

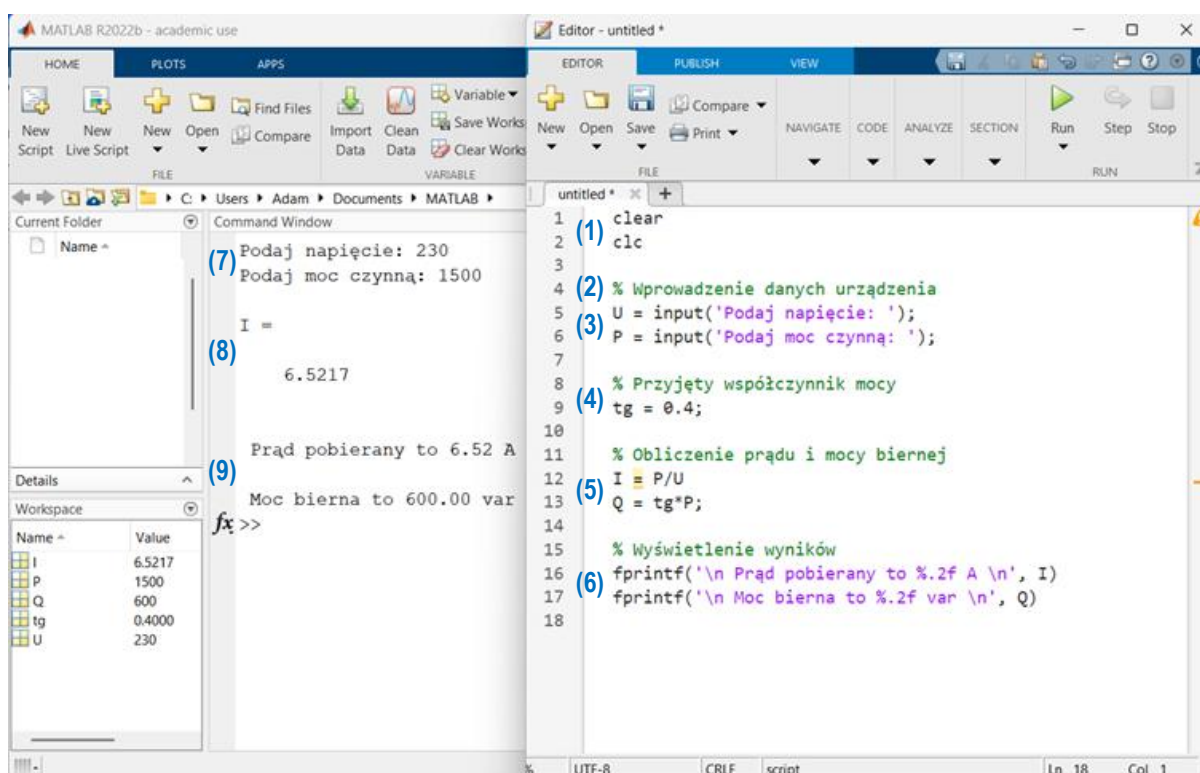
3.2. Tworzenie kodu

Aby rozpocząć pisanie skryptu należy uruchomić program *Matlab* i w lewym górnym rogu nacisnąć **przycisk 'New Script'** (Rys. 1). Pojawi się wówczas dodatkowe okno, przypominające plik tekstowy z pustym **polem tekstowym**. To właśnie w tym oknie zapisywany będzie kod podprogramu. Po jego utworzeniu cały skrypt uruchamia się klikając na **zieloną strzałkę 'Run'** na wstążce. Aby przerwać wykonywanie/zatrzymać skrypt wystarczy kliknąć na **'Stop' po prawej**. Zapisanie utworzonego kodu jest możliwe po **wybraniu opcji 'Save'** – pamiętaj, aby nazwać plik bez użycia polskich liter, znaków specjalnych i spacji! Zapisany plik posiada rozszerzenie **'m'**, stąd inna nazwa skryptów, czyli *m*-plik.



Rys. 1. Widok na okno główne programu i okno tworzenia skryptów

Tworzenie kodu zostanie omówione na praktycznym przykładzie (Rys. 2). Utwórzmy więc podprogram, który na podstawie podanego przez użytkownika napięcia zasilania (U) i mocy (P) urządzenia wyliczy wartość prądu (I) i moc bierną (Q) dowolnego sprzętu AGD, a wyniki wyświetli w **oknie poleceń**. Pisanie skryptu warto zacząć od komend 'clear' oraz 'clc' – pierwsza usuwa wszystko z **przestrzeni roboczej** ('Workspace') przechowującej stare dane i wyniki, a druga czyści **okno poleceń**, będące dla skryptu swoistym „wirtualnym wyświetlaczem”.



Rys. 2. Przykładowy skrypt i wyniki jego działania

Analizę skryptu z Rys. 2 zaczniemy od prawego okna i poleceń z **pola tekstowego**:

- (1) u góry wpisane są komendy 'clear' i 'clc' – wyczyszczą **przestrzeń roboczą** ze starych zmiennych oraz **okno poleceń** z dawnych komunikatów.
- (2) Następnie dodano tytuł sekcji '% Wprowadzenie danych urządzenia' – zapisany jest on jako tzw. komentarz. Komentarze zawsze zaczyna się do znaku '%' i nie są one wykonywane przez program, bowiem służą jedynie zawarciu informacji dla użytkownika w dowolnym miejscu kodu.

- (3) zadeklarowano zmienne 'U' i 'P', które będą przechowywać wartości napięcia i mocy. Ale musi je podać użytkownik, dlatego by program zapytał go o wartości tych parametrów zastosowano funkcję 'input()' – w nawiasie funkcji, między apostrofami, wpisano prośbę o podanie wartości.
- (4) potem zadano współczynnik mocy 'tg = 0.4'. W skrypcie można bowiem od razu deklarować zmienne jak, np. wspomniane 'tg' i od razu przypisać im konkretną wartość (w tym zespoloną albo jako działanie matematyczne).
- (5) w trzeciej sekcji podano wzory na poszukiwane natężenie prądu 'I' i moc bierną 'Q'. Program podstawia do nich wcześniej podane przez użytkownika 'U' i 'P' by obliczyć 'I' oraz 'P' i wpisany współczynnik 'tg' aby wyliczyć 'Q'.
- (6) na koniec użyto funkcji 'fprintf()', wyświetlającej tekst wraz z wartością danej zmiennej. Kod pokazuje przykład zapisania takiego łącznego komunikatu.

W następstwie wykonywania poleceń ze skryptu, w **oknie poleceń** wyświetliły się:

- (7) prośby o podanie wartości napięcia i mocy – tutaj użytkownik po kolei wpisał, np. 230 woltów i 1500 watów.
- (8) wartość zmiennej 'I'. Ale czemu tam się pojawiła? Otóż jest to następstwem braku średnika (;) na końcu wzoru na prąd (w **polu tekstowym**, linijka 12). Brak średnika jest komendą dla programu, że ma wyświetlić w **oknie poleceń** wynik z danej linii. Zaś jeżeli średnik pojawi się na końcu linijki, wtedy nic z niej nie będzie automatycznie wyświetlane w **oknie poleceń** (np. średnik stoi na końcu linii 9 i 13, więc 'tg' i 'Q' nie zostały wyświetlone).
- (9) komunikaty o obliczonych przez skrypt wartościach prądu i mocy biernej, które wcześniej zaprogramowano w liniijkach 16 i 17 w **polu tekstowym**.

Ciekawostką jest, że wszystkie zadeklarowane zmienne, czyli zarówno te wprowadzone w trakcie działania skryptu (np. 'U' i 'P'), zadane w kodzie (np. 'tg') oraz wyliczone po uruchomieniu *m*-pliku (np. 'I' i 'Q') są przechowywane w pamięci programu. Można je podejrzeć w **przestrzeni roboczej**. Wartości te można potem edytować, zapisać, a nawet wykorzystywać do dalszych obliczeń. Pamiętaj jedynie, że wykonanie polecenia 'clear' usunie wszystkie te wartości z **przestrzeni roboczej**! Więc jeśli nie chcesz ich kasować, nigdzie potem nie używaj 'clear'.

3.3. Przydatne funkcje

Pisząc skrypty wykorzystywać można różne funkcje dostępne w programie, jak np. `input()` i `fprintf()` pokazane wcześniej. Wśród funkcji wyróżnić też można tzw. operatory – porównują one dwie wartości i zwracają liczbę **1** (jeśli relacja między wartościami jest prawdziwa) lub liczbę **0** (jeśli relacja nie jest spełniona). Są też wykorzystywane w instrukcjach warunkowych i pętlach. Poniżej można znaleźć wybrane, często wykorzystywane funkcje, przydatne przy kolejnych zadaniach.

Tabela 1. Wybrane polecenia stosowane do realizacji operacji w ramach skryptów

Operatory relacyjne i logiczne		
Operator	Opis	Źródło
<code>A == B</code>	wartość A równa/tożsama wartości B	link
<code>A ~= B</code>	wartość A różna od wartości B	link
<code>A < B</code>	wartość A mniejsza do wartości B	link
<code>A > B</code>	wartość A większa od wartości B	link
<code>A <= B</code>	wartość A mniejsza lub równa wartości B	link
<code>A >= B</code>	wartość A większa lub równa wartości B	link
<code>A B</code>	wartość A LUB wartość B	link
<code>A & B</code>	wartość A ORAZ wartość B	link
Wprowadzanie i wyświetlanie danych		
Funkcja	Opis	Źródło
<code>input('<tekst>')</code>	wyświetla <tekst> jako komunikat dla użytkownika i czeka na wpisanie odpowiedzi	link
<code>load <nazwa></code>	wczytuje do przestrzeni roboczej zmienne z pliku <nazwa>.mat	link
<code>disp('<element>')</code>	wyświetla <element> w oknie poleceń, który może być tekstem lub zmienną	link
<code>fprintf('<tekst>', <zmiennie>)</code>	wyświetla w oknie poleceń <tekst> zawierający w sobie <zmiennie>	link
Działania matematyczne		
Funkcja	Opis	Źródło
<code>real(<liczba>)</code>	wyznacza część rzeczywistą <liczby> zespolonej	link

imag(<liczba>)	wyznacza część urojoną <liczby> zespolonej	link
angle(<liczba>)	wyznacza argument (kąt w radianach) <liczby> zespolonej	link
conj(<liczba>)	wyznacza sprzężenie <liczby> zespolonej	link
abs(<liczba>)	wyznacza moduł <liczby>	link
sqrt(<liczba>)	wyznacza pierwiastek kwadratowy <liczby>	link
exp(<liczba>)	wyznacza potęgę liczby Eulera, czyli $e^{<liczba>}$	link
log(<liczba>)	wyznacza logarytm naturalny, $\ln$, z <liczby>	link
log10(<liczba>)	wyznacza logarytm dziesiętny, $\log_{10}$, z <liczby>	link
sin(<liczba>) cos(<liczba>) tan(<liczba>) cot(<liczba>)	wyznacza wartość odpowiedniej funkcji trygonometrycznej dla podanej <liczby> w postaci kąta w radianach	link
asin(<liczba>) acos(<liczba>) atan(<liczba>) acot(<liczba>)	wyznacza wartość arcusa (czyli kąt w radianach) odpowiedniej funkcji trygonometrycznej dla podanej <liczby>	link
ceil(<liczba>)	zaokrągla <liczbę> w górę	link
floor(<liczba>)	zaokrągla <liczbę> w dół	link
round(<liczba>)	zaokrągla <liczbę> do najbliższej wartości całkowitej	link
mean(<tablica>)	wyznacza średnią wartość elementów wskazanej <tablicy>	link
sum(<tablica>)	wyznacza sumę wszystkich wartości ze wskazanej <tablicy>	link
min(<tablica>)	wyznacza najmniejszą wartość ze wskazanej <tablicy>	link
max(<tablica>)	wyznacza największą wartość ze wskazanej <tablicy>	link
det(<macierz>)	wyznacza wyznacznik wskazanej <macierzy>	link
inv(<macierz>)	Wyznacza odwrotność wskazanej <macierzy>	link

Oprócz nich używane są również bardziej zaawansowane funkcje, czyli instrukcje warunkowe **if** oraz **switch**, a także pętle **for** oraz **while**, znane z zajęć informatyki. Przed przystąpieniem do rozwiązywania ćwiczeń przypomnij sobie jak działają te instrukcje, oglądając krótkie poradniki [4 - 7]. Zaś, aby poznać budowę i stosowanie tych instrukcji odwiedź poświęcone im strony internetowe [8 - 11].

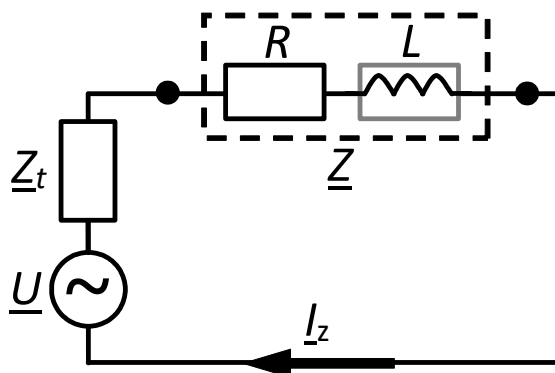
4. Przykład rozwiązania zagadnienia elektrotechnicznego

W tej części przedstawione zostanie rozwiązanie przykładowego zadania. Na tej podstawie można prześledzić użycie kolejnych poleceń i funkcji w ramach skryptu realizującego zadane obliczenia. Tym samym pełni to rolę wzorca pokazującego, w jaki sposób poradzić sobie z ćwiczeniami w kolejnym rozdziale.

Wstęp: Firma produkuje zestawy elektrycznego dogrzewania (termowentylator z termostatem, **Błąd! Nie można odnaleźć źródła odwołania.**). Dział projektujący termowentylatory określa i przesyła informacje o jego parametrach zastępczych, czyli o rezystancji R i indukcyjności L .



Rys. 3. Produkowany zestaw grzewczy



Rys. 4. Schemat elektryczny zestawu

Zadania:

- napisz skrypt obliczający sprawność termowentylatora, w zależności od jego parametrów zastępczych R i L ;
- dodaj instrukcję warunkową wyświetlającą ostrzeżenie, gdy sprawność będzie niższa niż 90%.

Dane:

- napięcie zasilające $\underline{U} = 230 \text{ V}$;
- pulsacja napięcia $\omega = 314 \text{ rad/s}$;
- impedancja termostatu $\underline{Z}_t = 0.7 + j1.2 \Omega$.

Rozwiązanie:

- 1) Najpierw należy zapisać wzór na impedancję termowentylatora. Ponieważ to urządzenie reprezentuje szeregowe połączenie rezystancji R i indukcyjności L (**Błąd! Nie można odnaleźć źródła odwołania.**), to wzór na impedancję zastępczą ma postać

$$\underline{Z} = R + j\omega L \quad (1)$$

- 2) Następnie należy rozwiązać obwód elektryczny (**Błąd! Nie można odnaleźć źródła odwołania.**), czyli obliczyć nieznany prąd $\underline{I}_z$. Zatem, np. stosując II prawo Kirchhoffa otrzymuje się

$$\underline{Z}_t \underline{I}_z + \underline{Z} \underline{I}_z = \underline{U} \quad (2)$$

$$(\underline{Z}_t + \underline{Z}) \underline{I}_z = \underline{U} \quad (3)$$

$$\underline{I}_z = \underline{U} / (\underline{Z}_t + \underline{Z}) \quad (4)$$

Wzór (4) to wyprowadzone równanie pozwalające wyliczyć nieznany prąd.

- 3) Do obliczenia sprawności muszą być znane moce czynne termowentylatora i źródła, a te są w istocie częściami rzeczywistymi mocy pozornych. Czyli najpierw należy obliczyć moc pozorną termowentylatora ze wzoru

$$\underline{S} = \underline{Z} \underline{I}_z \underline{I}_z^* \quad (5)$$

zaś moc pozorną źródła ze wzoru

$$\underline{S}_z = \underline{U} \underline{I}_z^* \quad (6)$$

- 4) Teraz można wyliczyć sprawność. Jest ona bowiem zdefiniowana, jako stosunek mocy czynnej urządzenia (termowentylatora) do mocy czynnej źródła

$$spr = \frac{P}{P_z} 100\% = \frac{\operatorname{Re}[\underline{S}]}{\operatorname{Re}[\underline{S}_z]} 100\% \quad (7)$$

- 5) Na koniec wynik ma zostać wyświetlony z właściwym komunikatem:

- jeżeli sprawność < 90%, to użytkownik ma zobaczyć monit ostrzegawczy;
- w innym przypadku użytkownik ma być poinformowany o sprawności.

Można to zrealizować za pomocą instrukcji warunkowej **if**, np. w postaci:

```
% Komunikat o wartości sprawności – instrukcja warunkowa
if spr < 90
    fprintf('\n SPRAWNOŚĆ ZA MAŁA! Trzeba zmniejszyć rezystancję! \n');
else
    fprintf('\n Sprawność w porządku i wynosi %.2f %% \n', spr);
end
```

6) Teraz można utworzyć kod wpisując dane, otrzymane wzory i komunikaty:

```
clear
clc

% Dane obwodu
U = 230; %napięcie zasilające
w = 314; %pulsacja napięcia
Zt = 0.7 + 1i*1.2; %impedancja termostatu

% Wprowadzenie rezystancji (R) i indukcyjności (L)
R = input('Podaj rezystancję: '); %tu użytkownik wprowadza wartość R
L = input('Podaj indukcyjność: '); %tu użytkownik wprowadza wartość L

% Obliczenie impedancji (Z) termowentylatora
Z = R + 1i*w*L; %tu wstawiono wzór (1)

% Obliczenie prądu zasilającego (Iz)
Iz = U/(Z + Zt); %tu wstawiono wyprowadzony wzór (4)

% Obliczenie mocy pozornych (S i Sz)
S = Z*Iz*conj(Iz); %moc pozorna termowentylatora liczona ze wzoru (5)
Sz = U*Iz; %moc pozorna źródła liczona ze wzoru (6)

% Obliczenie sprawności (spr) termowentylatora
spr = real(S)/real(Sz)*100; %moce czynne to części rzeczywiste S i Sz

% Komunikat o wartości sprawności – instrukcja warunkowa
if spr < 90 %sprawdzenie, czy sprawność jest mniejsza od 90%
    %komunikat, gdy sprawność jest mniejsza od 90%
    fprintf('\n SPRAWNOŚĆ ZA MAŁA! Trzeba zmniejszyć rezystancję! \n');
else %wykonanie akcji, gdy sprawność nie jest mniejsza od 90%
    %komunikat, gdy sprawność nie jest mniejsza od 90%
    fprintf('\n Sprawność w porządku i wynosi %.2f %% \n', spr);
end %zakończenie wykonywania instrukcji warunkowej
```

5. Przebieg ćwiczenia

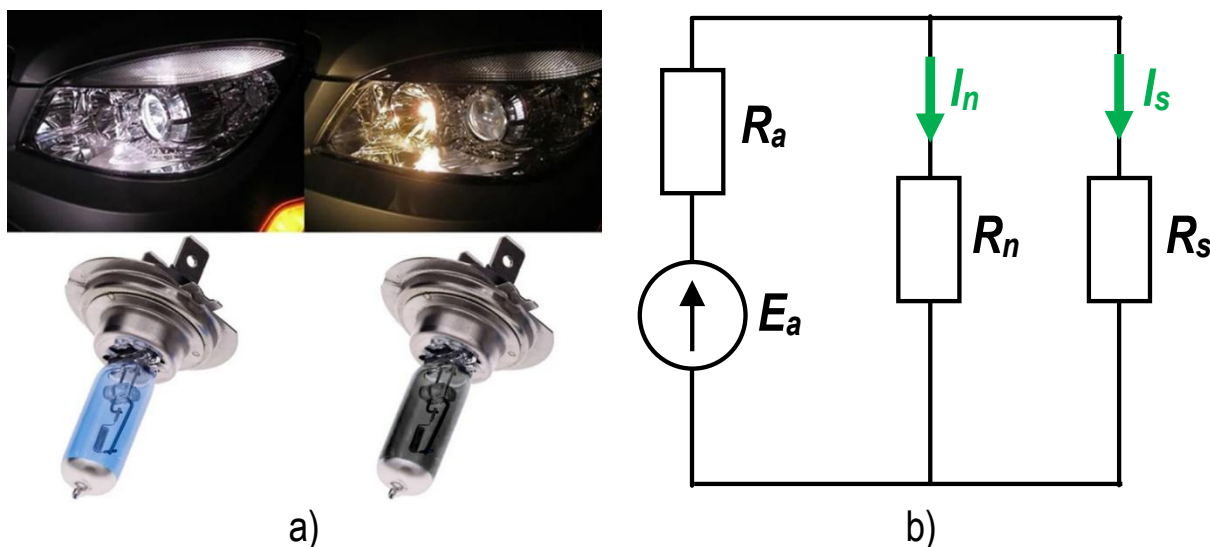
Skrypty idealnie nadają się do rozwiązywania różnych problemów inżynierskich. Ponieważ praktyka czyni mistrza, to niżej zamieszczono ćwiczenia dotyczące przygotowania skryptów, realizujących obliczenia wspomagające rozwiązywanie, projektowanie lub analizę układów elektrycznych.

5.1. Analiza przykładowego zadania

Samodzielnie przetestuj kod z punktu 4. **Przykład rozwiązania zagadnienia elektrotechnicznego.** W tym celu uruchom *Matlab*-a, włącz kreator skryptów, skopiuj do niego gotowy kod/skrypt, zapisz plik (pamiętaj, by nadać mu nazwę bez spacji, znaków polskich i specjalnych) i go uruchom. Gdy zostaniesz poproszony o podanie wartości R i L możesz zacząć od $R = 16$ i $L = 0.04$. Potem wprowadź inne dane i sprawdź, czy pojawi się monit o zbyt małej sprawności. Jeśli nie ma błędów i widzisz komunikaty to znak, że zadanie i test wykonano poprawnie :)

5.2. Zasilanie lamp samochodowych

Wstęp: Lampy przedniego oświetlenia samochodu wyposażone są w żarówki – jedną w nową, a druga w niemal wyeksploatowaną (Rys. 5). Wstawiono nowy akumulator stąd oczekuje się, że zapewne ta wymiana doprowadzi do szybszego zużycia starej żarówki. Rozwiązując obwód sprawdź, czy tak faktycznie będzie.



Rys. 5. a) wizualizacja działania żarówek; b) obwód akumulatora i żarówek

Zadania:

- napisz skrypt obliczający prądy płynące przez nową (I_n) i starą (I_s) żarówkę;
- przyjmij, że maksymalny prąd żarówki nie może przekroczyć 7 A.
- na podstawie wyników oceń, czy prądy zostały przekroczone oraz która żarówka może zużyć się szybciej (im wyższy prąd tym szybsze zużycie).

Dane:

- napięcie akumulatora $E_a = 14,5$ V;
- rezystancja wewnętrzna akumulatora $R_a = 0,4$ Ω ;
- rezystancja nowej żarówki $R_n = 1,3$ Ω ;
- rezystancja starej żarówki $R_s = 1,6$ Ω .

5.3. Prądy i napięcia w obwodzie

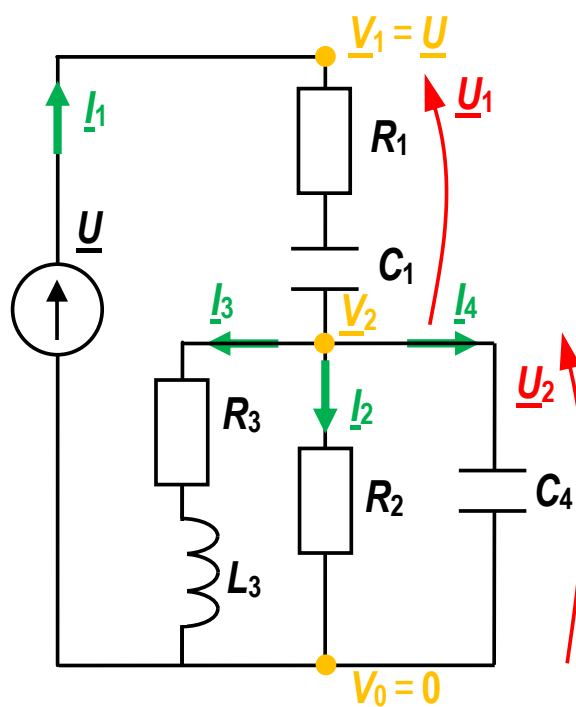
Wstęp: W laboratorium przygotowano stanowisko (Rys. 6), gdzie studenci poznają metodykę pomiaru prądu i napięcia. Ale przed uruchomieniem obwodu trzeba określić jakich wartości można się spodziewać...

Zadania:

- napisz skrypt obliczający wartości prądów i napięć, wskazanych na Rys. 6;
- na koniec wylicz wartości skuteczne (moduły) i przesunięcia fazowe tych prądów i napięć oraz wyświetl je w **oknie poleceń**.
- odpowiedź: możesz rozwiązać obwód metodą potencjałów węzłowych, gdzie $\underline{V}_0$ jest uziemiony i wtedy $\underline{V}_1 = \underline{U}$, a niewiadomą jest potencjał $\underline{V}_2$.

Dane:

- napięcie autotransformatora $\underline{U} = 100$ V;
- pulsacja napięcia $\omega = 314$ rad/s;
- rezystory $R_1 = 40$ Ω , $R_2 = 195$ Ω , $R_3 = 210$ Ω ;
- pojemności $C_1 = 10$ μ F, $C_4 = 10$ μ F;
- indukcyjność $L_3 = 1.42$ H.



Rys. 6. Obwód badany przez studentów na zajęciach laboratoryjnych

5.4. Dobór zabezpieczenia elektrowni wodnej

Wstęp: Prosument chce uruchomić małą elektrownię wodną (Rys. 7). Ale nim to zrobi należy zabezpieczyć jej główny obwód za pomocą wyłącznika (Rys. 8) o właściwym prądzie zadziałania (I) – prąd ten nie może być za mały, bo wyłącznik rozłączy poprawnie działający obwód ani za duży, by nie „przegapić” zwarcia.



Rys. 7. Mała elektrownia wodna

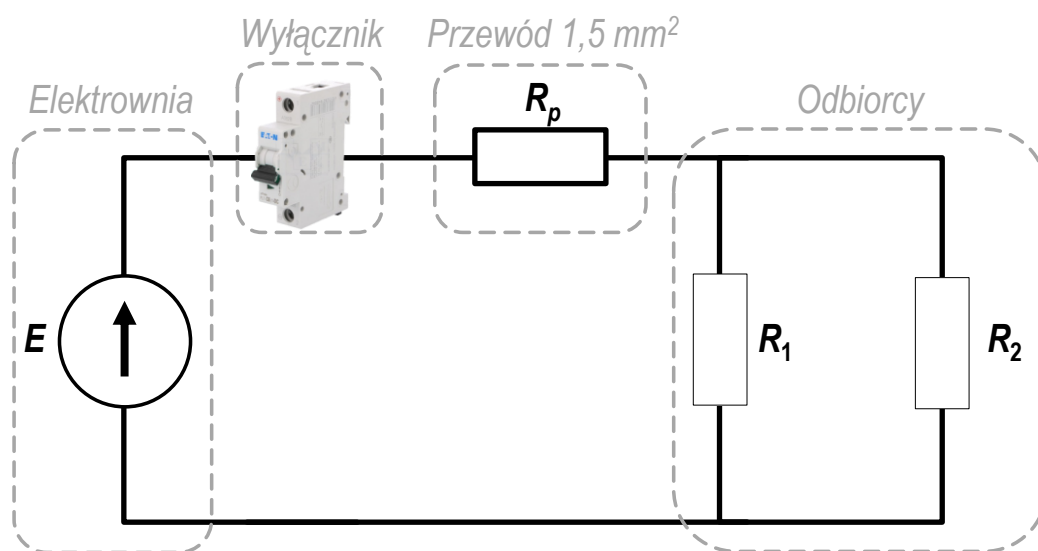


Rys. 8. Różne wyłączniki nadprądowe

Zadania:

- napisz skrypt obliczający prąd wypływający z generatora elektrowni (E), w zależności od rezystancji drugiego odbiorcy energii (R_2);

- dodaj instrukcję **if** lub **switch** wyświetlającą informację o obliczonym prądzie generatora i o tym, który wyłącznik z Rys. 8 należy zastosować (pamiętaj, że prąd wyłącznika musi być większy niż prąd generatora!);
- podczas testów skryptu załóż, że R_2 nigdy nie będzie mniejsze niż $40\ \Omega$.



Rys. 9. Schemat obwodu zasilanego z elektrowni wodnej

Dane:

- napięcie elektrowni $E = 390\text{ V}$;
- rezystancja przewodów zasilających $R_p = 2\ \Omega$;
- rezystancja pierwszego odbiornika $R_1 = 70\ \Omega$.

5.5. Test technologii ładowania samochodu elektrycznego* (dodatkowe)

Wstęp: Dwa samochody elektryczne „tankowane” są z tej samej stacji. Producent chce sprawdzić, czy układ połączeń obwodu ładowania akumulatorów ma wpływ na czas ładowania. Dlatego w samochodzie 1 obwód skojarzono w tzw. gwiazdę, a w samochodzie 2 w trójkąt (Rys. 10).

Zadania:

- napisz skrypt obliczający prądy fazowe (Rys. 10) ładujące oba samochody;
- w skrypcie policz potem moce pozorne pobierane przez każdy samochód

$$\underline{S}_x = \frac{\underline{I}_x \underline{I}_x^*}{\underline{Y}} \quad (8)$$

gdzie: x – faza A, B, C w samochodzie 1 lub AB, BC, CA w samochodzie 2;

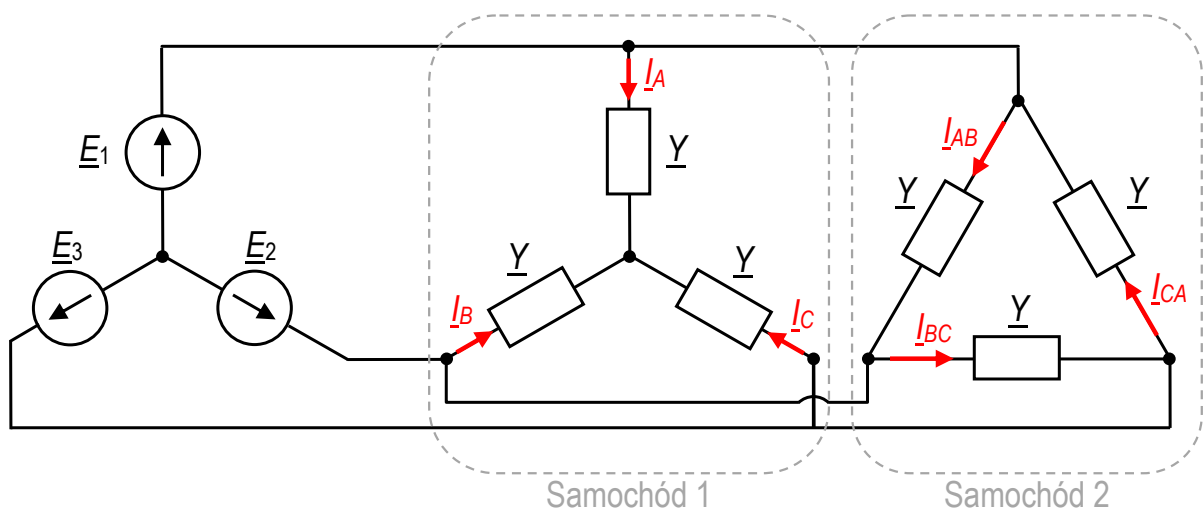
- następnie wyznacz czas ładowania obu pojazdów ze wzorów

$$t_1 = \frac{p}{S_A + S_B + S_C} \quad (9)$$

$$t_2 = \frac{p}{S_{AB} + S_{BC} + S_{CA}}$$

gdzie: $S_A, S_B, S_C, S_{AB}, S_{BC}, S_{CA}$ – moduły mocy pozornych kolejnych faz;

- wyświetl komunikaty z informacją o czasie ładowania obu samochodów;
- dodaj pętlę (np. **for**), która wyliczy czasy ładowania w zależności od różnych wartości pojemności akumulatorów p (przyjmij, że p może się zmieniać od 10000 Wh do 150000 Wh) – wyniki wyświetl, np. na wykresie.



Rys. 10. Układ połączeń stacji ładującej i dwóch samochodów elektrycznych

Dane:

- napięcia faz stacji $\underline{E}_1 = 400 \text{ V}$, $\underline{E}_2 = -200 + j346 \text{ V}$, $\underline{E}_3 = -200 - j346 \text{ V}$;
- admitancja fazy obwodu ładującego $\underline{Y} = 0.01 + j0.08 \text{ S}$;
- pojemność energetyczna akumulatorów $p = 60000 \text{ Wh}$.

6. Literatura

- [1] J. Brzóska, L. Dorobczyński, *Matlab: środowisko obliczeń naukowo-technicznych*, Mikom, Wydawnictwo Naukowe PWN, Warszawa, 2008.
- [2] A. Gilat, V. Subramaniam, *Numerical methods for engineers and scientists: an introduction with applications using Matlab*, John Wiley & Sons, 2011.
- [3] Praca zbiorowa pod redakcją J. Makala, *Practical problems for introductory electrical engineering*, Oficyna Wydawnicza Politechniki Białostockiej, Białystok, 2023.
- [4] MATLAB, *Relational Operators and Logicals | Coding Basics for MATLAB Onramp, Part 6*, Mathworks, 2020, on-line: [\[link\]](#)
- [5] MATLAB, *If/Else Statements | Coding Basics for MATLAB Onramp, Part 7*, Mathworks, 2020, on-line: [\[link\]](#)
- [6] MATLAB, *How to Use Switch Cases in MATLAB*, Mathworks, 2022, on-line: [\[link\]](#)
- [7] MATLAB, *For Loops and While Loops | Coding Basics for MATLAB Onramp, Part 8*, Mathworks, 2020, on-line: [\[link\]](#)
- [8] MATLAB Help Center, *if, elseif, else*, Mathworks, 2025, on-line: [\[link\]](#)
- [9] MATLAB Help Center, *switch, case, otherwise*, Mathworks, 2025, on-line: [\[link\]](#)
- [10] MATLAB Help Center, *for*, Mathworks, 2025, on-line: [\[link\]](#)
- [11] MATLAB Help Center, *while*, Mathworks, 2025, on-line: [\[link\]](#)

7. Pytania kontrolne

1. Jakie jest rozszerzenie i w jaki sposób pisać i edytować skrypty w *Matlab*-ie?
2. Gdzie pojawiają się rezultaty wykonania kodu i poleceń z *m*-pliku?
3. Jaka jest ogólna struktura wbudowanej w program funkcji, którą wywołuje się (wykorzystuje) w skrypcie?

8. Wymagania BHP

Warunkiem przystąpienia do praktycznej realizacji ćwiczenia jest zapoznanie się z instrukcją BHP i instrukcją przeciw pożarową oraz przestrzeganie zasad w nich zawartych.

W trakcie zajęć laboratoryjnych należy przestrzegać następujących zasad:

- Sprawdzić, czy urządzenia dostępne na stanowisku laboratoryjnym są w stanie kompletnym, nie wskazującym na fizyczne uszkodzenie.
- Jeżeli istnieje taka możliwość, należy dostosować warunki stanowiska do własnych potrzeb, ze względu na ergonomię. Monitor ustawić w sposób zapewniający stałą i wygodną obserwację dla wszystkich członków zespołu.
- Sprawdzić prawidłowość połączeń urządzeń.
- Włączenie komputera może nastąpić po wyrażeniu zgody przez prowadzącego.
- W trakcie pracy z komputerem zabronione jest spożywanie posiłków i napojów.
- W przypadku zakończenia pracy należy zakończyć sesję przez wydanie polecenia wylogowania. Zamknięcie systemu operacyjnego może się odbywać tylko na wyraźne polecenie prowadzącego.
- Zabronione jest dokonywanie jakichkolwiek przełączeń oraz wymiana elementów składowych stanowiska.
- Zabroniona jest zmiana konfiguracji komputera, w tym systemu operacyjnego i programów użytkowych, która nie wynika z programu zajęć i nie jest wykonywana w porozumieniu z prowadzącym zajęcia.
- W przypadku zaniku napięcia zasilającego należy niezwłocznie wyłączyć wszystkie urządzenia.
- Stwierdzone wszelkie braki w wyposażeniu stanowiska oraz nieprawidłowości w funkcjonowaniu sprzętu należy przekazywać prowadzącemu zajęcia.
- Zabrania się samodzielnego włączania, manipulowania i korzystania z urządzeń nie należących do danego ćwiczenia.
- W przypadku wystąpienia porażenia prądem elektrycznym należy niezwłocznie wyłączyć zasilanie. Przed odłączeniem napięcia nie dotykać porażonego.