

SYMULATOR PARAMETRÓW OBWODOWYCH OBWODU ZE SPRZĘŻENIEM INDUKCYJNYM

METODY NUMERYCZNE – PROJEKT ZALICZENIOWY

ARTUR CYRWUS

Spis treści

1	PRZEDMIOT I ZAKRES PROJEKTU	2
2	ROZWIĄZANIE PROBLEMU TECHNICZNEGO	2
3	MATEMATYCZNY MODEL OBWODU	3
4	IMPLEMENTACJA.....	4
4.1	Zarys rozwiązań przyjętych w projekcie	4
4.2	Dokumentacja kodu źródłowego	5
4.2.1	Klasa CModel.....	5
4.2.2	Klasa CTrafo.....	7
4.2.3	Klasa CSignal.....	10
4.3	Podstawy teoretyczne implementacji funkcji numerycznych	11
4.3.1	Metody numerycznego rozwiązywania równań różniczkowych.....	11
4.3.2	Metody interpolacji i aproksymacji wielomianami algebraicznymi	12
4.3.3	Metody interpolacji funkcjami sklejanymi	13
4.3.4	Metody całkowania numerycznego	14
4.3.5	Metody numerycznego poszukiwania pierwiastków funkcji nieliniowych	15
5	SYMULACJA PARAMETRÓW OBWODOWYCH – CZĘŚĆ 1.	16
5.1	Symulacja przebiegów prądów i napięcia	17
5.1.1	Wymuszenie prostokątne, $U = 120V$ dla $t < T/2$, oraz $0V$ dla $t \geq T/2$, $T = 3 s$	17
5.1.2	Wymuszenie sinusoidalne, $e(t) = 240 \sin(t)$	18
5.1.3	Wymuszenie sinusoidalne, $e(t) = 210 \sin(2\pi ft)$, dla $f = 5 Hz$	19
5.1.4	Wymuszenie sinusoidalne, $e(t) = 120 \sin(2\pi ft)$, dla $f = 50 Hz$	20
5.2	Omówienie wyników i wnioski	21
6	SYMULACJA PARAMETRÓW OBWODOWYCH – CZĘŚĆ 2.	22
6.1	Porównanie metod interpolacji/aproksymacji charakterystyki	22
6.1.1	Wymuszenie sinusoidalne, $e(t) = 240 \sin(2t)$	23
6.1.2	Wymuszenie sinusoidalne, $120 \sin(2\pi ft)$, dla $f = 50 Hz$	25
6.2	Omówienie wyników i wnioski	26
7	SYMULACJA PARAMETRÓW OBWODOWYCH – CZĘŚĆ 3.	27
7.1	Praktyczne porównanie metod całkowania numerycznego	27
7.1.1	Moc czynna w układzie z transformatorem o stałej indukcyjności wzajemnej	27
7.1.2	Moc czynna w układzie z transformatorem o z nieliniową indukcyjnością wzajemną	28
7.2	Omówienie wyników i wnioski	28
8	SYMULACJA PARAMETRÓW OBWODOWYCH – CZĘŚĆ 4.	29
8.1	Zestawienie wyników obliczania funkcji celu	29
8.2	Omówienie wyników i wnioski	29

1 PRZEDMIOT I ZAKRES PROJEKTU

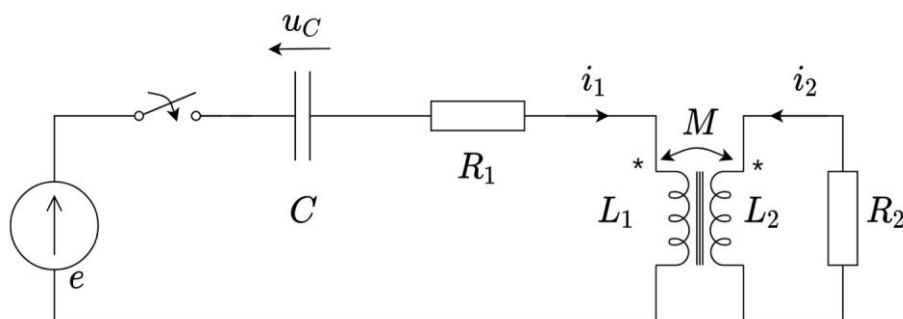
Zasadniczą część projektu stanowi opracowanie programu, służącego do symulacji parametrów obwodowych w układzie elektrycznym ze sprzężeniem indukcyjnym. Program ten umożliwi wykonanie szeregu symulacji, których celem będzie zweryfikowanie poprawności jego działania oraz porównanie wyników, osiąganych w zależności od wybranej metody numerycznej.

Projekt podzielony jest na 4 części:

1. Opracowanie symulatora i wykonanie symulacji stanów nieustalonych w badanym układzie elektrycznym, przy zastosowaniu metody Eulera i metody ulepszonej Eulera do obliczenia wartości chwilowych;
2. Zaimplementowanie funkcji zmiennej indukcyjności wzajemnej M , z możliwością wyboru metody numerycznej do interpolacji/aproksymacji charakterystyki $M = f(u)$ oraz porównanie wyników symulacji w zależności od rodzaju wymuszeń w obwodzie oraz w zależności od obranej metody numerycznej;
3. Implementacja numerycznego całkowania wartości chwilowych prądów w gałęziach obwodu, w celu wyznaczenia mocy czynnej na rezystancji R_1 , R_2 oraz porównanie precyzji obliczeń osiąganych poszczególnymi metodami całkowania, i w zależności od rodzaju wymuszeń w obwodzie;
4. Wykorzystanie symulatora do wyliczenia metodami iteracyjnymi (metodą bisekcji, siecznych i metodą Newtona) podanej funkcji celu, mianowicie przy jakiej częstotliwości napięcia wymuszającego układ pracuje z zadaną mocą czynną, a następnie porównanie precyzji obliczeń i złożoności obliczeniowej poszczególnych metod.

2 ROZWIĄZANIE PROBLEMU TECHNICZNEGO

W zadaniu projektowym rozpatrywany jest układ elektryczny ze sprzężeniem indukcyjnym. Składa się on ze źródła napięcia e , włącznika, kondensatora o pojemności C , oporników o rezystancji R_1 , R_2 , a także transformatora, dla którego znana jest indukcyjność uzwojenia pierwotnego L_1 i wtórnego L_2 oraz sprzężenie indukcyjne. Indukcyjność wzajemna M może być wartością stałą, ale również — zależnie od wybranego wariantu symulacji — może zmieniać się nieliniowo w funkcji napięcia na uzwojeniu pierwotnym. Schemat badanego układu elektrycznego przedstawiono na poniższym rysunku.



Rys. nr 1. Schemat badanego układu elektrycznego

Stosując prawa Kirchhoffa, badany obwód można opisać układem złożonym z trzech równań różniczkowych zwyczajnych pierwszego rzędu:

$$\begin{aligned} E &= u_C + R_1 i_1 + L_1 \frac{di_1}{dt} + M \frac{di_2}{dt} \\ 0 &= R_2 i_2 + L_2 \frac{di_2}{dt} + M \frac{di_1}{dt} \\ i_1 &= C \frac{du_C}{dt} \end{aligned}$$

W układzie równań występują trzy zmienne stanu, tj. prądy i_1 , i_2 oraz napięcie u_C . Ich wartość początkową przyjmuje się $i_1 = 0$ [A], $i_2 = 0$ [A] oraz $u_C = 0$ [V], co odpowiada założeniu, że w chwili początkowej włącznik jest otwarty. Po jego włączeniu, sygnały na elementach obwodu szybko stabilizują się i przyjmują wartości stałe lub periodyczne — zgodnie z częstotliwością sygnału wejściowego. Wówczas:

$$\begin{aligned} \frac{di_1}{dt} &= \frac{1}{\frac{L_1}{M} - \frac{M}{L_2}} \left[-\frac{R_1}{M} i_1 + \frac{R_2}{L_2} i_2 - \frac{1}{M} u_C - \frac{1}{M} e \right] \\ \frac{di_2}{dt} &= \frac{1}{\frac{M}{L_1} - \frac{L_2}{M}} \left[-\frac{R_1}{L_2} i_1 + \frac{R_2}{M} i_2 - \frac{1}{L_1} u_C - \frac{1}{L_1} e \right] \\ \frac{du_C}{dt} &= \frac{1}{C} i_1 \end{aligned}$$

3 MATEMATYCZNY MODEL OBWODU

Stosując dla zmiennych stanu zwykłe podstawienie $i_1 = y_1$, $i_2 = y_2$, $u_C = y_3$ podane wyżej równania można zapisać:

$$\begin{aligned} \frac{dy_1}{dt} &= \frac{1}{\frac{L_1}{M} - \frac{M}{L_2}} \left[-\frac{R_1}{M} y_1 + \frac{R_2}{L_2} y_2 - \frac{1}{M} y_3 - \frac{1}{M} e \right] \\ \frac{dy_2}{dt} &= \frac{1}{\frac{M}{L_1} - \frac{L_2}{M}} \left[-\frac{R_1}{L_2} y_1 + \frac{R_2}{M} y_2 - \frac{1}{L_1} y_3 - \frac{1}{L_1} e \right] \\ \frac{dy_3}{dt} &= \frac{1}{C} y_1 \end{aligned}$$

Przekształcając je na postać macierzową, otrzymuje się:

$$\begin{bmatrix} \frac{dy_1}{dt} \\ \frac{dy_2}{dt} \\ \frac{dy_3}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_1}{MD_1} & \frac{R_2}{L_2 D_1} & -\frac{1}{MD_1} \\ -\frac{R_1}{L_1 D_2} & \frac{R_2}{MD_2} & -\frac{1}{L_1 D_2} \\ \frac{1}{C} & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} + \begin{bmatrix} -\frac{1}{MD_1} \\ -\frac{1}{L_1 D_2} \\ 0 \end{bmatrix} e$$

gdzie:

$$\begin{aligned} D_1 &= \frac{L_1}{M} - \frac{M}{L_2} \\ D_2 &= \frac{M}{L_1} - \frac{L_2}{M} \end{aligned}$$

Pamiętając, że wielkości R_1 , R_2 , L_1 , L_2 , D_1 , D_2 oraz C to w istocie z góry założone w zadaniu wartości liczbowe, widzimy tu czytelną zależność — dla danego wektora y_1 , y_2 , y_3 , znany jest wektor pochodnych dy_1 , dy_2 , dy_3 po dt . Z kolei wektor tych pochodnych pozwala określać wartości y_1 , y_2 , y_3 , dla dowolnego t w sposób przybliżony, przy zastosowaniu algorytmów dyskretnych.

Przedstawiona powyżej zależność stanowi podstawę implementacji symulatora obwodu przedstawionego na Rys. nr 1.

4 IMPLEMENTACJA

4.1 Zarys rozwiązań przyjętych w projekcie

Matematyczny model obwodu zaimplementowano w języku skryptowym środowiska *MatLab*. Wielkości fizyczne układu, takie jak pojemność C rezystancja R_1, R_2 indukcyjność L_1, L_2 , a także indukcyjność wzajemna M to zmienne skalarne, którym przypisuje się wartości podane w zadaniu.

Symulacja parametrów obwodowych polega na wyliczaniu szeregów czasowych, określających wartości chwilowe e (napięcia w źródle) oraz zmiennych stanu y_1, y_2, y_3 — czyli prądów i_1, i_2 w gałęziach obwodu i napięcia u_C . Dla szeregów czasowych przyjmuje się odpowiednio mały krok czasowy Δt , co zapewnia rozsądny kompromis pomiędzy ciągłością uzyskiwanych przebiegów prądów i napięć w czasie (a tym samym dokładnością i stabilnością symulacji), a ilością koniecznych do wyliczenia stanów chwilowych układu, bowiem im krótsze Δt , tym więcej stanów chwilowych do wyliczenia.

Jeśli chodzi o wartości chwilowe e , są one wyliczane z funkcji, która w zadanym scenariuszu symulacji jednoznacznie określa zależność napięcia względem czasu. Natomiast wektory wartości chwilowych y_1, y_2, y_3 , wylicza się sukcesywnie dla każdego kolejnego stanu przez:

- rozwiązanie dla danego stanu chwilowego macierzowej postaci układu równań, celem wyliczenia wektora różniczek dy/dt ;
- wyliczenie metodą Eulera (lub ulepszoną metodą Eulera) kolejnych wartości zmiennych stanu.

W tym miejscu, na krótkie omówienie zasługuje sposób wyliczania macierzy i wektora współczynników dla układu równań opisujących badany obwód. Ich wartości zależą wprost od R_1, R_2, L_1, L_2 i C a także od M . O ile w 1. części zadania projektowego wszystkie te wielkości są stałe, to już w części 2. należy przyjąć, że indukcyjność wzajemna M zależy od napięcia u_L , a zatem jest zmienna w czasie. Oznacza to, że nie w każdym przypadku wystarczy jednokrotne wyliczyć elementy macierzy i wektora przez podstawienie danych z zadania. Implementacja musi uwzględniać ich przeliczanie dla każdej chwili czasowej, gdy zajdzie taka konieczność.

Z wprowadzeniem zmiennej indukcyjności wzajemnej, wiąże się również konieczność zaimplementowania kilku metod numerycznych służących do interpolacji/aproksymacji zależności $M = f(u)$. Tę nieliniową zależność zwykle określa jedynie kilka punktów zdjętych z charakterystyki, a ponieważ symulator ma płynnie odwzorowywać przebiegi napięć i prądów, indukcyjność wzajemna M musi być opisana w postaci optymalnie dopasowanej funkcji interpolującej lub aproksymującej. Projekt obejmuje opracowanie czterech takich metod: interpolacja wielomianem Lagrange’a, interpolacja funkcjami sklejanymi 3. stopnia, aproksymacja wielomianem 3. stopnia oraz wielomianem 5. stopnia.

Część 3. projektu polega na wyliczaniu mocy czynnej na elementach rezystancyjnych. Wielkość tą wylicza się przez numeryczne scałkowanie wartości chwilowych $P = i^2 R$ przy czym symulator ma udostępniać dwie metody całkowania — metodę złożoną prostokątów i metodę złożoną parabol.

Wyżej opisaną funkcjonalność w pełni wykorzystuje 4. część projektu. Rozwiązuje się w niej zadanie wyszukania pierwiastka funkcji celu, mianowicie przy jakiej częstotliwości napięcia wymuszającego obwód pracuje z zadaną mocą czynną. Poszukiwanie rozwiązania funkcji celu polega na iteracyjnej zmianie częstotliwości napięcia wymuszającego i wznowianiu całej procedury obliczeniowej tak długo, aż poszukiwana wartość mocy zostanie osiągnięta z założoną precyzją. Zadanie to realizuje się metodą bisekcji, metodą siecznych lub metodą Newtona.

4.2 Dokumentacja kodu źródłowego

4.2.1 Klasa `CModel`

Klasa `CModel` stanowi abstrakcję badanego układu elektrycznego, w którym występują takie elementy jak kondensator C , oporniki R_1 i R_2 , oraz transformator o indukcyjności uzwojeń L_1 , L_2 oraz indukcyjności wzajemnej M .

4.2.1.1 Konstruktor

Obiekty tej klasy tworzy się bezparametrowym konstruktorem `CModel`.

4.2.1.2 Właściwości publiczne (pola publiczne)

Nazwa właściwości	Typ domniemany	Przeznaczenie
<code>R1</code>	liczba rzeczywista	rezystancja opornika R_1 [Ω]
<code>R2</code>	liczba rzeczywista	rezystancja opornika R_2 [Ω]
<code>C</code>	liczba rzeczywista	pojemność kondensatora C [F]
<code>L1</code>	liczba rzeczywista	indukcyjność uzwojenia pierwotnego L_1 [H]
<code>L2</code>	liczba rzeczywista	indukcyjność uzwojenia wtórnego L_2 [H]
<code>M</code>	liczba rzeczywista	indukcyjność wzajemna M [H]
<code>Trafo</code>	obiekt klasy <code>CTrafo</code>	charakterystyka indukcyjności wzajemnej transformatora

W nowo utworzonym obiekcie klasy `CModel`, właściwościom `R1`, `R2`, `C`, `L1`, `L2` przypisuje się odpowiednie wartości liczbowe, określając w ten sposób podstawowe parametry elektryczne badanego układu. Obiekt musi również „znać” charakterystykę transformatora, dlatego też przed rozpoczęciem symulacji należy ustawić właściwość `Trafo`. Osiąga się to przez przypisanie do tej właściwości odpowiednio zainicjowanego obiektu z klasy `CTrafo`.

Nie jest natomiast konieczne ustawianie właściwości `M`. Wartość przechowywana w tym polu znajduje się pod pełną kontrolą zaimplementowanych w obiekcie algorytmów. Automatycznie ustalają one tę wartość, na podstawie zadanej charakterystyki indukcyjności wzajemnej transformatora, przy czym wartość ta pozostaje niezmienna przez cały cykl symulacji obwodu z transformatorem o stałej indukcyjności wzajemnej albo — w przypadku transformatora o zmiennej indukcyjności — wyliczana jest w każdym kroku symulacji, stosownie do bieżącego stanu w jakim znajduje się obwód.

4.2.1.3 Właściwości prywatne (pola prywatne)

Nazwa właściwości	Typ domniemany	Przeznaczenie
<code>A</code>	tablica [3,3] liczb rzeczywistych	Macierz współczynników z układu równań opisujących badany obwód: $\begin{bmatrix} -\frac{R_1}{MD_1} & \frac{R_2}{L_2D_1} & -\frac{1}{MD_1} \\ -\frac{R_1}{L_1D_2} & \frac{R_2}{MD_2} & -\frac{1}{L_1D_2} \\ \frac{1}{C} & 0 & 0 \end{bmatrix}$
<code>V</code>	wektor [3] liczb rzeczywistych	Wektor współczynników z układu równań opisujących badany obwód: $\begin{bmatrix} -\frac{1}{MD_1} \\ -\frac{1}{L_1D_2} \\ 0 \end{bmatrix}$

Powyższe właściwości nie są dostępne dla użytkownika i pozostają pod wyłączną kontrolą klasy `CModel`. Wartości tych pól wyliczane są przez funkcję prywatną `update`, jednorazowo lub w każdym cyklu symulacji.

4.2.1.4 Funkcje publiczne (metody publiczne)

Nazwa funkcji / parametry	Typ zwracany	Cel
SolveEuler Lista argumentów: <code>Signal</code> (obiekt klasy <code>CSignal</code>)	tablica [3,n] liczb rzeczywistych	Wylicza dla zadanego wymuszenia kompletne szeregi czasowe prądów i_1 , i_2 oraz napięcia u_C w układzie, stosując metodę Eulera.
SolveImprovedEuler Lista argumentów: <code>Signal</code> (obiekt klasy <code>CSignal</code>)	tablica [3,n] liczb rzeczywistych	Wylicza dla zadanego wymuszenia kompletne szeregi czasowe prądów i_1 , i_2 oraz napięcia u_C w układzie, stosując metodę ulepszoną Eulera.

Dostępne w klasie `CModel` funkcje (metody publiczne) służą do wyliczania kompletnych szeregów czasowych prądów i_1 , i_2 oraz napięcia u_C w układzie elektrycznym, przy zadanym wymuszeniu e . Własności sygnału wymuszającego, takie jak napięcie, częstotliwość, kształt sygnału, itp., w pełni określa obiekt klasy `CSignal` będący argumentem funkcji.

Wynik obliczeń zwracany jest w postaci 3-wierszowej tablicy liczb rzeczywistych. Zawiera ona wyliczone szeregi czasowe, przy czym wiersze tej tablicy reprezentują odpowiednio:

1. wartości chwilowe prądu i_1
2. wartości chwilowe prądu i_2
3. wartości chwilowe napięcia u_C

W funkcjach `SolveEuler` i `SolveImprovedEuler` przyjęty został następujący algorytm:

- a) pobiera się wartość chwilową napięcia e (z wektora liczb przekazanego w obiekcie klasy `CSignal`);
- b) pobiera się zapisane dla poprzedniej chwili czasowej wartości zmiennych i_1 , i_2 , u_C (z tworzonej właśnie 3-elementowej tablicy liczb);
- c) ww. wielkości tj. skalar e oraz wektor zmiennych $[y_1, y_2, y_3]$ podstawia się do zapisanego w postaci macierzowej układu równań różniczkowych, a specjalna funkcja różniczkująca (metoda `response`) oblicza i zwraca wektor różniczek. Różniczki te określają „przyrosty” i_1 , i_2 , u_C w bieżącej chwili czasowej;
- d) wartości pobrane w pkt. b) oraz przyrosty obliczone w pkt. c) używa się do wyliczenia nowych wartości zmiennych i_1 , i_2 , u_C dla bieżącej chwili czasowej, stosując metodę Eulera (funkcja `calcEuler`) lub ulepszoną metodę Eulera (funkcja `calcImprovedEuler`);
- e) wyniki obliczeń z pkt. d) dopisywane są w kolejnej wolnej kolumnie tablicy liczb, wspomnianej w pkt. b);
- f) kroki z pkt. od a) do e) wykonywane są w pętli, tak długo, aż zostanie przetworzony cały zadany szereg czasowy napięcia wymuszającego, tj. wszystkie elementy wektora liczb, wspomniane w pkt. a).

4.2.1.5 Funkcje prywatne (metody prywatne)

Nazwa funkcji / parametry	Typ zwracany	Cel
update Lista argumentów: brak	—	Przelicza elementy macierzy A i wektora V . Struktury te przechowują macierzowy zapis układu równań opisujących badany obwód. Funkcja <code>update</code> uruchamiana jest w obiekcie jednorazowo albo w każdym kroku obliczeń, gdy indukcyjność wzajemna M jest wielkością zmienną, zależną od u_L .
response Lista argumentów: Y (wektor [3] liczb rzeczywistych) e (liczba rzeczywista)	wektor [3] liczb rzeczywistych	Oblicza i zwraca wektor różniczek dla zmiennych stanu, tj. pojedyncze rozwiązanie układu równań różniczkowych, dla chwilowej wartości napięcia e i zadanego wektora zmiennych stanu Y .

4.2.1.6 Numeryczne funkcje lokalne

Klasa `CModel` wykorzystuje następujące funkcje numeryczne:

Nazwa funkcji / parametry	Typ zwracany	Cel
<code>calcEuler</code> Lista argumentów: <code>dY</code> (funkcja) <code>Y</code> (wektor [3] liczb rzeczywistych) <code>e</code> (liczba rzeczywista) <code>h</code> (liczba rzeczywista)	wektor [3] liczb rzeczywistych	Stosując zadaną funkcję różniczkującą dY , oblicza i zwraca nowe wartości zmiennych stanu, które po upływie chwili czasowej h , przy chwilowym napięciu wymuszającym e , opiszą obwód znajdujący się w stanie Y . Obliczenie nowych wartości zmiennych stanu odbywa się według formuły Eulera.
<code>calcImprovedEuler</code> Lista argumentów: <code>dY</code> (funkcja) <code>Y</code> (wektor [3] liczb rzeczywistych) <code>e</code> (liczba rzeczywista) <code>h</code> (liczba rzeczywista)	wektor [3] liczb rzeczywistych	Stosując zadaną funkcję różniczkującą dY , oblicza i zwraca nowe wartości zmiennych stanu, które po upływie chwili czasowej h , przy chwilowym napięciu wymuszającym e , opiszą obwód znajdujący się w stanie Y . Obliczenie nowych wartości zmiennych stanu odbywa się według formuły ulepszonej Eulera.

Są to numeryczne funkcje o zastosowaniu ogólnym, dlatego zdefiniowano je poza obrębem klasy. Podstawy teoretyczne ich implementacji podano w pkt. 4.3.1.

4.2.2 Klasa `CTrafo`

Klasa `CTrafo` to abstrakcja zjawiska sprzężenia indukcyjnego, występującego w transformatorach. Może ono mieć stałą wartość lub zmieniać się w zależności od napięcia u_L na uzwojeniu pierwotnym transformatora.

W przypadku gdy zjawisko indukcji wzajemnej ma charakter stały, funkcjonalność klasy `CTrafo` ogranicza się jedynie do przechowania i zwracania na żądanie stałej wartości liczbowej M , podanej w trakcie tworzenia obiektu. Natomiast gdy indukcyjność wzajemna jest wielkością zależną od u_L , zwrócenie odpowiedniej wartości M wymaga skomplikowanych obliczeń. Ich ciężar „bierze na siebie” jedna z kilku wyspecjalizowanych metod numerycznych. Dokonują one interpolacji/aproksymacji charakterystyki $M = f(u_L)$ (określonej zwykle przez kilka punktów), a następnie zwracają konkretną wartość liczbową dla zadanego argumentu.

Metodę numeryczną wybiera się przez ustawienie odpowiedniej wartości w polu `smoothing`. Wybrana metoda, zostaje przyjęta jako obowiązująca przy wyliczaniu wartości M to jest przy każdorazowym wywołaniu funkcji `MutualInductance`.

4.2.2.1 Konstruktor

Tworzenie obiektów tej klasy umożliwia jednoparametrowy konstruktor `CTrafo(MI)`

W parametrze określa się rodzaj i wartość indukcyjności wzajemnej. Podanie konkretnej wartości skalarnej jest równoznaczne z przyjęciem jej jako stałej wielkości. Natomiast jeśli indukcyjność wzajemna zależy od u_L , w parametrze podaje się dwuwierszową tablicę (u, m) określającą współrzędne punktów charakterystyki $M = f(u_L)$. Każda kolumna tablicy określa jeden punkt charakterystyki. Wszystkie współrzędne mają być dodatnimi liczbami rzeczywistymi.

4.2.2.2 Właściwości publiczne (pola publiczne)

Nazwa właściwości	Typ domniemany	Przeznaczenie
<code>smoothing</code>	liczba całkowita	Identyfikator metody wygładzania charakterystyki $M = f(u_L)$.

Dla obiektów `CTrafo` ze stałą indukcyjnością wzajemną nie ma potrzeby ustawiania jakichkolwiek właściwości. Natomiast gdy indukcyjność jest zmienna, można dokonać wyboru konkretnej metody wygładzania

charakterystyki $M = f(u_L)$, ustawiając w polu `smoothing` wartość 1,2,3,4 lub 5, co odpowiada przyjęciu następującej metody:

1. interpolacja wielomianem Lagrange'a (**domyślna**);
2. interpolacja funkcjami sklejanymi 1. stopnia (węzły połączone odcinkami prostymi);
3. interpolacja funkcjami sklejanymi 3. stopnia (węzły połączone krzywymi giętymi tzw. „splajnami”);
4. aproksymacja wielomianem 3. stopnia;
5. aproksymacja wielomianem 5. stopnia;

Przy ustawianiu właściwości `smoothing` można również posłużyć się „słownikiem” `smoothingMethod` dostępnym w klasie. Wybór metody odbywa się wówczas przez podanie nazwy metody — odpowiednio: 'Lagrange', 'Polyline', 'Spline', '3degPolynomial' lub '5degPolynomial'.

4.2.2.3 Właściwości tylko do odczytu (pola chronione)

Nazwa właściwości	Typ domniemany	Przeznaczenie
<code>M</code>	liczba rzeczywista; tablica [2,n] liczb rzeczywistych	Indukcyjność wzajemna jako skalar lub jako tablica punktów określających charakterystykę $M = f(u_L)$ gdy indukcyjność zależy od napięcia na uzwojeniu pierwotnym.
<code>variableInductance</code>	wartość logiczna	Znacznik określający, czy indukcyjność wzajemna jest zmienna (określona tablicą punktów charakterystyki) czy jest wartością stałą.
<code>smoothingMethod</code>	obiekt klasy <code>Map</code>	Słownik z nazwami zaimplementowanych metod wygładzania charakterystyki zmiennej indukcyjności wzajemnej.

Powyższe właściwości przeznaczone są wyłącznie do odczytu. Pole `M` zostaje ustawione podczas tworzenia obiektu i nie może być zmieniane. Wartość logiczna w polu `variableInductance` ustawiana jest automatycznie przez konstruktor klasy, stosownie do typu danych określających indukcyjność wzajemną. Natomiast `smoothingMethod` to statyczny element klasy `CTrafo`, który zawiera enumeratywną listę metod wygładzania charakterystyki (tzw. słownik).

4.2.2.4 Właściwości prywatne (pola prywatne)

Nazwa właściwości	Typ domniemany	Przeznaczenie
<code>s3Knots</code>	tablica [2,n] liczb rzeczywistych	Tablica węzłów interpolacji dla funkcji sklepanej 3. stopnia, stanowiącej metodę interpolacji charakterystyki $M = f(u_L)$.
<code>s3Coeffs</code>	wektor [n] liczb rzeczywistych	Wektor współczynników dla funkcji sklepanej 3. stopnia, stanowiącej metodę interpolacji charakterystyki $M = f(u_L)$.
<code>s3Alpha</code> , <code>s3Beta</code>	liczba rzeczywista	Parametry graniczne dla funkcji sklepanej 3. stopnia (wartości dobrane empirycznie, wyłącznie dla zadania projektowego!).

Powyższe właściwości nie są dostępne dla użytkownika i pozostają pod wyłączną kontrolą klasy `CTrafo`. Pola `s3Knots`, `s3Coeffs` wyliczane są jednorazowo w trakcie zainicjowania obiektu, a wartości w polach `s3Alpha`, `s3Beta` zostały ustalone empirycznie (mogą być poprawne wyłącznie dla niniejszego zadania projektowego!).

4.2.2.5 Funkcje publiczne (metody publiczne)

Nazwa funkcji / parametry	Typ zwracany	Cel
<code>MutualInductance</code> Lista argumentów: <code>uL</code> (liczba rzeczywista)	liczb rzeczywista	Oblicza i zwraca indukcyjność wzajemną dla zadanego napięcia u_L (mierzonego na uzwojeniu pierwotnym transformatora), stosując wybraną metodę wygładzania tzn. interpolacji / aproksymacji charakterystyki $M = f(u_L)$.

Funkcja `MutualInductance` pełni rolę „interfejsu”, który na każde żądanie zwraca wartość indukcyjności wzajemnej M . Jeśli jest to wielkość zależna od napięcia u_L , wszystkie obliczenia niezbędne do wyznaczenia jej wartości wykonują wyspecjalizowane funkcje numeryczne, stosownie do przyjętej metody obliczeniowej.

4.2.2.6 Numeryczne funkcje lokalne

Klasa `CTrafo` wykorzystuje następujące funkcje numeryczne:

Nazwa funkcji / parametry	Typ zwracany	Cel
LagrangePolynomial Lista argumentów: x (liczba rzeczywista) X (wektor [n] liczb rzeczywistych) Y (wektor [n] liczb rzeczywistych)	liczba rzeczywista	Oblicza wielomian interpolacyjny Lagrange’a dla serii punktów o współrzędnych x,y (danych w wektorach X, Y) oraz zwraca wartość y tego wielomianu dla podanego x .
PolynomialApprox Lista argumentów: deg (liczba całkowita) x (liczba rzeczywista) X (wektor [n] liczb rzeczywistych) Y (wektor [n] liczb rzeczywistych)	liczba rzeczywista	Oblicza wielomian aproksymacyjny stopnia deg dla serii punktów o współrzędnych x,y (danych w wektorach X, Y) oraz zwraca wartość y tego wielomianu dla podanego x .
Polyline Lista argumentów: x (liczba rzeczywista) X (wektor [n] liczb rzeczywistych) Y (wektor [n] liczb rzeczywistych)	liczba rzeczywista	Oblicza i zwraca wartość y dla podanego x , gdy węzły funkcji sklejanej 1. stopnia mają współrzędne x,y (w wektorach X, Y)
Spline Lista argumentów: K (tablica [2,n] liczb rzeczywistych) C (wektor [n] liczb rzeczywistych) x (liczba rzeczywista)	liczba rzeczywista	Oblicza i zwraca wartość y dla podanego x , gdy węzły funkcji sklejanej 3. stopnia podane są w tablicy K a współczynniki c podane w wektorze C . Elementy tablicy K wylicza funkcja <code>splineKnots</code> a wektora C funkcja <code>splineCoeffs</code> . Wielkości te wylicza się dla zadanej krzywej interpolowanej jednorazowo.
SplinePhi Lista argumentów: K (tablica [2,n] liczb rzeczywistych) i (liczba całkowita) x (liczba rzeczywista)	liczba rzeczywista	Oblicza i zwraca wartość i -tej funkcji bazowej dla podanego x gdy węzły funkcji sklejanej 3. stopnia podane są w tablicy K . Jest to funkcja pomocnicza, używana przez funkcję <code>Spline</code> .
SplineCoeffs Lista argumentów: K (tablica [2,n] liczb rzeczywistych) $alpha$ (liczba rzeczywista) $beta$ (liczba rzeczywista)	wektor [n] liczb rzeczywistych	Oblicza i zwraca wektor współczynników c , gdy węzły funkcji sklejanej 3. stopnia podane są w tablicy K , a parametry graniczne α i β są znane (dane w polach prywatnych klasy). Jest to funkcja pomocnicza, wykonywana jednorazowo przez konstruktor klasy <code>CTrafo</code> .
SplineKnots Lista argumentów: X (wektor [n] liczb rzeczywistych) Y (wektor [n] liczb rzeczywistych) k (liczba całkowita)	tablica [2,k] liczb rzeczywistych	Rozmieszcza k równoodległych węzłów funkcji sklejanej 3. stopnia, najlepiej dopasowanych do serii punktów o współrzędnych x,y (danych w wektorach X, Y), oraz zwraca tablicę tych węzłów. Jest to funkcja pomocnicza, wykonywana jednorazowo przez konstruktor klasy <code>CTrafo</code> .

Powyższe funkcje zdefiniowano je poza obrębem klasy `CTrafo` ponieważ są to numeryczne funkcje ogólnego zastosowania. Podstawy teoretyczne ich implementacji podano w pkt. 4.3.2.

4.2.3 Klasa `CSignal`

Klasa `CSignal` stanowi abstrakcję sygnału wymuszającego. Może być nim napięcie stałe lub periodyczne — o kształcie sinusoidalnym lub prostokątnym. Parametry sygnału, takie jak czas trwania, jego kształt i okres a także wartość napięcia określa się w trakcie tworzenia obiektu. W rezultacie, klasa tworzy złożoną strukturę zawierającą m.in. dwa wektory:

1. wektor chwil czasowych `time`
2. wektor napięć chwilowych `voltage`

Wygenerowane w ww. wektorach serie liczb określają w sposób dyskretny dwie wielkości: czas i napięcie, a konkretnie — jaka jest wartość napięcia wymuszającego w każdej chwili trwania sygnału, przy czym cały zadany czas trwania sygnału podzielony jest na równe i odpowiednio krótkie przedziały czasu Δt , a każde napięcie chwilowe wyliczane jest z odpowiedniej funkcji opisującej zależność $e(t)$. Wielkości skwantyfikowane w taki sposób, stanowią ważną część „danych wejściowych” niezbędnych do numerycznej analizy zmian stanu układu.

4.2.3.1 Konstruktor

Obiekty tej klasy tworzy się sparametryzowanym konstruktorem `CSignal(timespan, sig, u, T, density)`

W parametrach określa się czas trwania sygnału `timespan`, jego kształt `sig` oraz okres `T`, a także wartość napięcia `u`, przy czym kształt sygnału `sig` określa się odpowiednim ciągiem znakowym:

- 'sine' — sygnał sinusoidalny;
- 'sqre' — sygnał prostokątny nieujemny;

Użycie innego identyfikatora kształtu np. 'DC', powoduje domyślne przyjęcie sygnału stałonapięciowego.

Dodatkowo, parametrem `density` steruje się długością przedziału Δt . Im „większą” zadano gęstość sygnału, tym krótszy jest przedział czasu w wygenerowanych szeregach czasowych. Gęstość `density` określa się odpowiednim ciągiem znakowym:

- 'dense' — wysoka gęstość $\Delta t = T / 512$
- 'normal' — gęstość normalna $\Delta t = T / 128$
- 'light' — niska gęstość $\Delta t = T / 32$

4.2.3.2 Właściwości tylko do odczytu (pola chronione)

Nazwa właściwości	Typ domniemany	Przeznaczenie
<code>sig</code>	ciąg znakowy	Identyfikator kształtu sygnału ('sine', 'sqre', ...)
<code>u</code>	liczba rzeczywista	Napięcie [V]
<code>T</code>	liczba rzeczywista	Okres sygnału [s]
<code>f</code>	liczba rzeczywista	Częstotliwość [Hz]
<code>step</code>	liczba rzeczywista	Krok czasowy [s]
<code>time</code>	wektor [n] liczb rzeczywistych	Wektor chwil czasowych [s]
<code>voltage</code>	wektor [n] liczb rzeczywistych	Wektor napięć chwilowych [V]

Powyższe właściwości przeznaczone są wyłącznie do odczytu. Wszystkie pola zostają ustawione w trakcie tworzenia obiektu i nie mogą być zmieniane. Pole `f` obliczane jest z prostej i znanej zależności $f = 1 / T$. Jeśli chodzi o wielkość `step`, która reprezentuje wspomnianą już długość przedziału Δt , ustalana jest ona zgodnie z zależnością podaną w pkt. 4.2.3.1. Wektory chwilowe `time` i `voltage` generowane są automatycznie przez klasę `CSignal`.

4.2.3.3 Funkcje publiczne (metody publiczne)

Nazwa funkcji / parametry	Typ zwracany	Cel
<code>caption</code> Lista argumentów: Brak	łańcuch znakowy	Zwraca napis, który określa aktualnie ustawiony w obiekcie typ sygnału ('sine', 'sqre', ...), napięcie [V], a dla sygnałów periodycznych również okres [s] i częstotliwość [Hz]

Funkcja `caption` zwraca zwięzłą informację tekstową o parametrach sygnału wymuszającego.

4.3 Podstawy teoretyczne implementacji funkcji numerycznych

4.3.1 Metody numerycznego rozwiązywania równań różniczkowych

W pliku skryptowym *CModel.m* zaimplementowane zostały dwie metody numerycznego rozwiązywania równań różniczkowych — prostą oraz ulepszoną metodę Eulera.

4.3.1.1 Metoda prosta Eulera

Funkcja `calcEuler` stanowi realizację metody Eulera i służy do przybliżonego rozwiązywania równań różniczkowych postaci $y' = f(x, y)$ przy znanym warunku początkowym, tzn. gdy znana jest wartość $y(x_0) = y_0$. Metoda prosta Eulera opiera się na następującym wzorze iteracyjnym:

$$y_i = y_{i-1} + hf(x_{i-1}, y_{i-1})$$

gdzie:

y_i — rozwiązanie obliczane dla wartości x_i

y_{i-1} — rozwiązanie obliczone w kroku poprzednim

x — kolejne wartości z badanego przedziału $a \leq x \leq b$ takie, że:

$$x_i = x_{i-1} + h, \quad i = 0, 1, 2, \dots, n$$

h — krok iteracyjny

$$h = \frac{b - a}{n}$$

Obliczenia realizowane są w następujący sposób:

- przedział $\langle a, b \rangle$ dzieli się na n równych podprzedziałów o szerokości h
- dla znanych wartości początkowych $x_0 = a$; $y(x_0) = y_0$ oblicza się wartość zadanej funkcji $f(x_0, y_0)$ a następnie rozwiązanie y_1 z ww. formuły Eulera — jest to rozwiązanie dla wartości $x_1 = x_0 + h$
- obliczenia dla kolejnych x_i, y_i wykonuje się iteracyjnie wg ww. formuły i powtarza tak długo aż zostanie osiągnięty $x_n = b$
- otrzymany zbiór wartości $y_0, y_1, y_2, \dots, y_n$ jest rozwiązaniem równania dla przedziału $a \leq x \leq b$

4.3.1.2 Metoda ulepszona Eulera

Funkcja `calcImprovedEuler` stanowi realizację ulepszonej metody Eulera. W metodzie tej uzyskuje się nieco dokładniejsze wyniki niż za pomocą metody prostej. Obliczenia przebiegają według podobnego algorytmu jak opisano to w pkt. 4.3.1.1, ale wzór iteracyjny w metodzie ulepszonej Eulera ma postać:

$$y_i = y_{i-1} + hf\left(x_{i-1} + \frac{h}{2}, y_{i-1} + \frac{h}{2}f(x_{i-1}, y_{i-1})\right)$$

Istotną różnicą jest fakt, że w każdym kroku iteracyjnym funkcja f wykorzystywana jest w obliczeniach dwukrotnie.

4.3.2 Metody interpolacji i aproksymacji wielomianami algebraicznymi

W związku z wykorzystywaniem zagadnienia interpolacji i aproksymacji dla charakterystyk określających indukcyjność wzajemną, w pliku skryptowym *CTrafo.m* zaimplementowano szereg metod numerycznych w tym:

- interpolację wielomianem Lagrange’a;
- interpolację funkcjami sklejanymi 3. Stopnia, tzw. „splajnami”;
- aproksymację wielomianową stopnia 3. i stopnia 5.

4.3.2.1 Interpolacja wielomianem Lagrange’a

Funkcja `LagrangePolynomial` stanowi realizację interpolacji wielomianem Lagrange’a. Dla znanych $n + 1$ punktów (tzw. węzłów interpolacji), wielomian Lagrange’a ma stopień n i opisuje go następujący wzór:

$$L(x) = \sum_{i=0}^n y_i \cdot \prod_{j=0 \wedge j \neq i}^n \frac{x - x_j}{x_i - x_j}$$

Do powyższego wzoru podstawia się współrzędne węzłów interpolacji (x_i, y_i) dla $i = 0, 1, 2, \dots, n$, i zadany x a następnie wylicza odpowiadającą mu wartość $y = L(x)$.

4.3.2.2 Aproksymacja wielomianami algebraicznymi

Funkcja `PolynomialApprox` realizuje metodę aproksymacji wielomianem algebraicznym stopnia m . Postać ogólną wielomianu określa równanie:

$$W_m(x) = a_0 + a_1x + a_2x^2 + \dots + a_{m-1}x^{m-1} + a_mx^m$$

gdzie a_i dla $i = 0, 1, 2, \dots, m$, to współczynniki rzeczywiste wielomianu.

Mając co najmniej m równań obliczenie współczynników jest możliwe, ponieważ będzie to układ m równań z m niewiadomymi. Zatem, gdy znane jest przynajmniej m punktów, wielomian aproksymacyjny stopnia m da się wyliczyć. Zwykle liczba punktów jest znaczna, a stopień wielomianu niski.

Z postaci ogólnej widzimy, że bazą poszukiwanego wielomianu są funkcje $\{1, x, x^2, \dots, x^m\}$. W celu rozwiązania układu równań buduje się macierz złożoną z tylu wierszy, ile znanych jest punktów i tylu kolumn, ile jest funkcji bazowych.

Załóżmy, że znane jest $n + 1$ punktów a poszukiwany dla nich wielomian aproksymacyjny ma być stopnia m . Wówczas macierz przedstawia się następująco:

$$M = \begin{bmatrix} 1 & x_0 & \dots & x_0^m \\ 1 & x_1 & \dots & x_1^m \\ \dots & \dots & \dots & \dots \\ 1 & x_n & \dots & x_n^m \end{bmatrix}$$

Aby obliczyć współczynniki a_i trzeba rozwiązać układ $M^T M \cdot A = M^T \cdot Y$

gdzie:

- Y — wektor (kolumnowy) zadanych wartości y_i
- A — wektor (kolumnowy) poszukiwanych współczynników a_i
- M — macierz złożona z wartości wyliczonych przez podstawienie zadanych x_i do funkcji bazowych

Oczywiście M^T oznacza transponowaną macierz M .

Przez rozwiązanie powyższego układu uzyskuje się wartości wszystkich współczynników a_i . Po ich obliczeniu, do równania ogólnego wielomianu aproksymacyjnego podstawia się wyliczone a_i oraz zadane x i w prosty sposób oblicza wartość $y = W_m(x)$ według wzoru:

$$y = a_0 + \sum_{i=1}^m a_i x^i$$

4.3.3 Metody interpolacji funkcjami sklejanymi

4.3.3.1 Interpolacja funkcjami sklejanymi 3. stopnia

Funkcje [Spline](#), [SplinePhi](#), [SplineCoeffs](#), [SplineKnots](#), realizują łącznie algorytm interpolacji splajnami tzn. funkcjami sklejanymi 3. stopnia.

W interpolacji tego typu, szukamy krzywej będącej nie jedną, lecz złożeniem wielu funkcji, z których każda jest odpowiednio dobranym wielomianem 3. stopnia. Funkcje te wyznacza się dla następujących po sobie podprzedziałów równej długości, przy czym muszą one być tak dobrane, aby po „sklejeniu” ich jedna za drugą, tworzyły razem ciągłą krzywą, „płynnie” przechodzącą przez wszystkie zadane węzły interpolacji. W miejscach „sklejeń” (czyli na krańcu podprzedziałów), funkcje te muszą mieć tę samą pierwszą i drugą pochodną.

Ze względu na szereg wymagań jakie muszą spełnić funkcje bazowe, określa je skomplikowany wzór:

$$\Phi_i(x) = \frac{1}{h^3} \begin{cases} (x - x_{i-2})^3 & : x \in \langle x_{i-2}, x_{i-1} \rangle \\ (x - x_{i-2})^3 - 4(x - x_{i-1})^3 & : x \in \langle x_{i-1}, x_i \rangle \\ (x_{i+2} - x)^3 - 4(x_{i+1} - x)^3 & : x \in \langle x_i, x_{i+1} \rangle \\ (x_{i+2} - x)^3 & : x \in \langle x_{i+1}, x_{i+2} \rangle \\ 0 & : x \in \mathbb{R} - \langle x_{i-2}, x_{i+2} \rangle \end{cases}$$

gdzie:

x_i — współrzędna x i -tego węzła interpolacji

x — współrzędna x punktu, dla którego poszukiwana jest wartość funkcji bazowej

h — odległość między kolejnymi węzłami interpolacji

$$h = \frac{b - a}{n}$$

przy czym przedział $\langle a, b \rangle$ wyznacza pierwszy i ostatni węzeł, a ponieważ wszystkie węzły są równoodległe, prawdziwa jest zależność:

$$x_i = a + i \cdot h$$

Zakładając, że jest $n + 1$ węzłów, funkcji bazowych jest tyle ile węzłów, plus dodatkowo 2, albowiem oprócz funkcji $\Phi_0, \Phi_1, \Phi_2, \dots, \Phi_n$, na początku i na końcu jest jeszcze Φ_{-1} i Φ_{n+1}

Za obliczenia związane z funkcją bazową odpowiada funkcja [SplinePhi](#).

Funkcja sklejana $S_3(x)$ opisana jest równaniem:

$$S_3(x) = c_{-1}\Phi_{-1}(x) + c_0\Phi_0(x) + c_1\Phi_1(x) + \dots + c_n\Phi_n(x) + c_{n+1}\Phi_{n+1}(x)$$

lub w skrócie:

$$S_3(x) = \sum_{i=-1}^{n+1} c_i \Phi_i(x)$$

W celu wyliczenia współczynników c , rozwiązuje się układ równań:

$$c_{i-1} + 4c_i + c_{i+1} = y_i \quad i = 0, 1, \dots, n$$

$$-c_{-1} + c_1 = \frac{h}{3} \cdot \alpha$$

$$-c_{n-1} + c_{n+1} = \frac{h}{3} \cdot \beta$$

z którego, po przekształceniach otrzymujemy $n + 1$ równań z $n + 1$ niewiadomymi:

$$\begin{array}{rcll} 4c_0 & + & 2c_1 & = y_0 + \frac{h}{3} \cdot \alpha \\ c_0 & + & 4c_1 & + c_2 = y_1 \\ & & c_1 & + 4c_2 + c_3 = y_2 \\ & & \dots & \dots = \dots \\ & & c_{n-2} & + 4c_{n-1} + c_n = y_{n-1} \\ & & 2c_{n-1} & + 4c_n = y_n - \frac{h}{3} \cdot \beta \end{array}$$

Powyższy układ równań można zapisać w postaci macierzowej:

$$\begin{bmatrix} 4 & 2 & & & & \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \\ & & & & 2 & 4 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \dots \\ c_{n-1} \\ c_n \end{bmatrix} = \begin{bmatrix} y_0 + \frac{h}{3} \cdot \alpha \\ y_1 \\ y_2 \\ \dots \\ y_{n-1} \\ y_n - \frac{h}{3} \cdot \beta \end{bmatrix}$$

Wyliczenie współczynników c zgodnie z powyższym równaniem realizuje funkcja [SplineCoeffs](#). Jej zadaniem jest przygotowanie odpowiednio wypełnionej macierzy oraz wektora wyrazów wolnych, a następnie wykonanie operacji macierzowej. Wynik rozwiązania powyższego układu równań zostaje zapisany w wektorze [s3Coeffs](#), w celu wielokrotnego wykorzystania tych wartości w dalszych obliczeniach.

Mając obliczone współczynniki c oraz wartości funkcji bazowych oblicza się ostatecznie wartość funkcji sklejanej $S_3(x)$ dla zadanego x . Realizuje to zaimplementowana funkcja [Spline](#)

Oczywiście cały opisany wyżej proces realizowany jest dla zadanych węzłów interpolacji o współrzędnych (x_i, y_i) , przy czym zakłada się, że węzły te są równoodległe, tzn. spełniona jest zależność $x_{i+1} = x_i + h$.

Warunek co do równoodległego położenia węzłów niekoniecznie jest spełniony w przypadku zadanego zbioru punktów określających interpolowaną charakterystykę, dlatego dodatkowo, aby ominąć tę niedogodność, zaimplementowano funkcję [SplineKnots](#). Wyznacza się w niej rozmieszczenie równoodległych węzłów interpolacji, na podstawie punktowo określonej charakterystyki.

4.3.4 Metody całkowania numerycznego

Numeryczne całkowanie metodami złożonymi prostokątów i parabol wchodzi w zakres części 3. projektu. Dlatego w plikach skryptowych *Cz3S.m* i *Cz3Z.m* zaimplementowano te metody. W obu przypadkach całka interpretowana jest jako suma pól obszarów pod wykresem krzywej w zadanym przedziale całkowania. Przedział całkowania jest już podzielony na podprzedziały o równej długości, gdyż wynika to z metody rozwiązywania równań różniczkowych, w wyniku których uzyskuje się przebiegi podlegające całkowaniu w tej części zadania projektowego.

4.3.4.1 Całkowanie metodą złożoną prostokątów

Funkcja [intRect](#) jest realizacją całkowania metodą złożoną prostokątów, którą prowadzi się zgodnie ze wzorem:

$$I = \int_a^b f(x)dx \cong h \sum_{i=0}^{n-1} f(x_i)$$

gdzie:

- h — długość podprzedziału, wynikająca z podziału przedziału całkowania $\langle a, b \rangle$ na n równych części
- $f(x_i)$ — wartość funkcji podcałkowej dla x

4.3.4.2 Całkowanie metodą złożoną parabol (metoda Simpsona)

Funkcja [intSimpson](#), stanowi realizację numerycznego całkowania metodą złożoną parabol. W metodzie tej n podprzedziałów o długości h , rozpatruje się jako $n/2$ podprzedziałów o długości $2h$, a na każdym podprzedziale wyznaczonym przez $[x_i, x_{i+2}]$, $i = 0, 1, 2, \dots, m-2$ stosuje się następujący wzór parabol:

$$\int_{x_i}^{x_{i+2}} f(x)dx \cong \frac{h}{3} (f(x_i) + 4f(x_{i+1}) + f(x_{i+2}))$$

Sumując otrzymane całki otrzymuje się wynik końcowy.

4.3.5 Metody numerycznego poszukiwania pierwiastków funkcji nieliniowych

Poszukiwanie pierwiastków dla funkcji nieliniowych jest tematem części 4. projektu, toteż w pliku skryptowym *Cz4.m* zaimplementowano trzy metody realizacji tego zadania:

- bisekcji;
- siecznych;
- Newtona.

W każdej z powyższych metod rozpatruje się przedział izolacji (a, b) , w którym znajduje się aktualnie szukany pierwiastek. Optymalny dobór przedziałów izolacji realizuje się algorytmicznie wg ściśle określonych reguł, jednak w niniejszym projekcie pominięto tę część implementacji, a przedział izolacji zapewniający zbieżność procesu poszukiwania pierwiastka został określony empirycznie.

4.3.5.1 Metoda bisekcji

Funkcja `bisectonRoot`, stanowi implementację numerycznego wyszukiwania pierwiastka metodą bisekcji. W metodzie tej wykonywany jest iteracyjnie podział badanego przedziału „na pół” oraz wybór do kolejnego kroku iteracji tej części, w której funkcja zmienia znak. Proces realizowany jest tak długo, aż bieżąca szerokość przedziału jest mniejsza niż wielkość dopuszczalnego błędu wyniku.

4.3.5.2 Metoda siecznych

Funkcja `secantRoot`, stanowi realizację numerycznego wyszukiwania pierwiastka metodą siecznych. Metoda ta również realizowana jest iteracyjnie. W kolejnych krokach poszukiwana wartość przybliżana jest przez wnoszenie poprawki Δx wyrażonej poniższym równaniem:

$$x_{n+1} = x_n - \Delta x$$
$$\Delta x = \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$

Wzór jest przeliczany tak długo, aż poprawka będzie mniejsza niż wielkość dopuszczalnego błędu wyniku.

Podkreślić należy, że zgodnie z podanym równaniem, w każdym kroku iteracyjnym tej metody zachodzi konieczność obliczania funkcji dwukrotnie, co w przypadku, gdy funkcję tę cechuje duża złożoność obliczeniowa może to negatywnie wpływać na szybkość obliczeń realizowanych tą metodą.

4.3.5.3 Metoda Newtona

Funkcja `newtonRoot` jest kolejną metodą iteracyjnego poszukiwania przybliżonej wartości pierwiastka. w której założoną wartość początkową przybliża się przez wnoszenie poprawki Δx . W przypadku metody Newtona, wartość poprawki wyliczana jest wg równania:

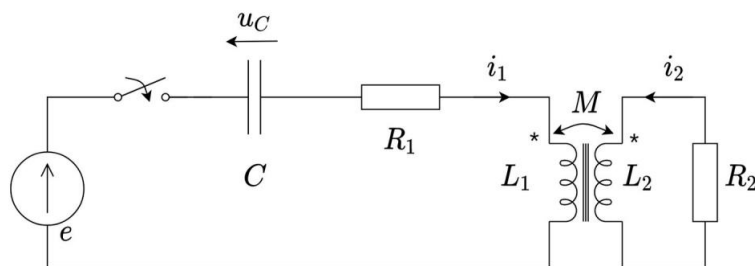
$$x_{n+1} = x_n - \Delta x$$
$$\Delta x = \frac{f(x_n)}{f'(x_n)}$$

Wzór jest przeliczany tak długo, aż poprawka będzie mniejsza niż wielkość dopuszczalnego błędu wyniku.

5 SYMULACJA PARAMETRÓW OBWODOWYCH – CZĘŚĆ 1.

W pierwszej części, symulacji poddano układ o następujących parametrach:

- $R_1 = 0.1 \, \Omega$
- $R_2 = 10 \, \Omega$
- $C = 0,5 \, \text{F}$
- $L_1 = 3 \, \text{H}$
- $L_2 = 5 \, \text{H}$
- $M = 0.8 \text{H}$



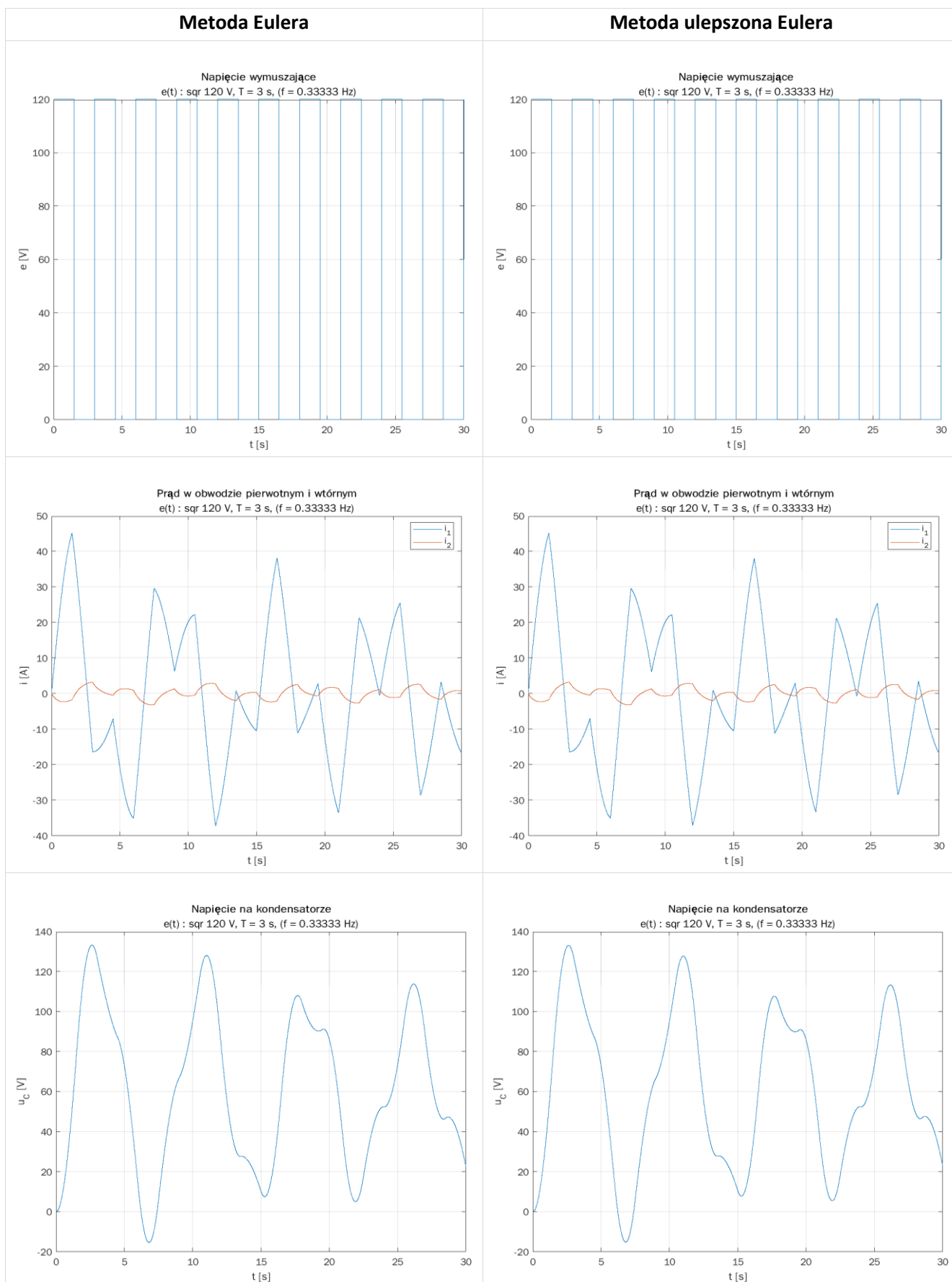
Badany jest stan nieustalony w czasie $0 < t < 30 \, \text{s}$ dla następujących wymuszeń:

1. sygnał prostokątny, $U = 120 \text{V}$ dla $t < T/2$, oraz 0V dla $t \geq T/2$, $T = 3 \, \text{s}$.
2. sygnał sinusoidalny, $e(t) = 240 \sin(t)$
3. sygnał sinusoidalny, $e(t) = 210 \sin(2\pi ft)$, dla $f = 5 \, \text{Hz}$
4. sygnał sinusoidalny, $e(t) = 120 \sin(2\pi ft)$, dla $f = 50 \, \text{Hz}$

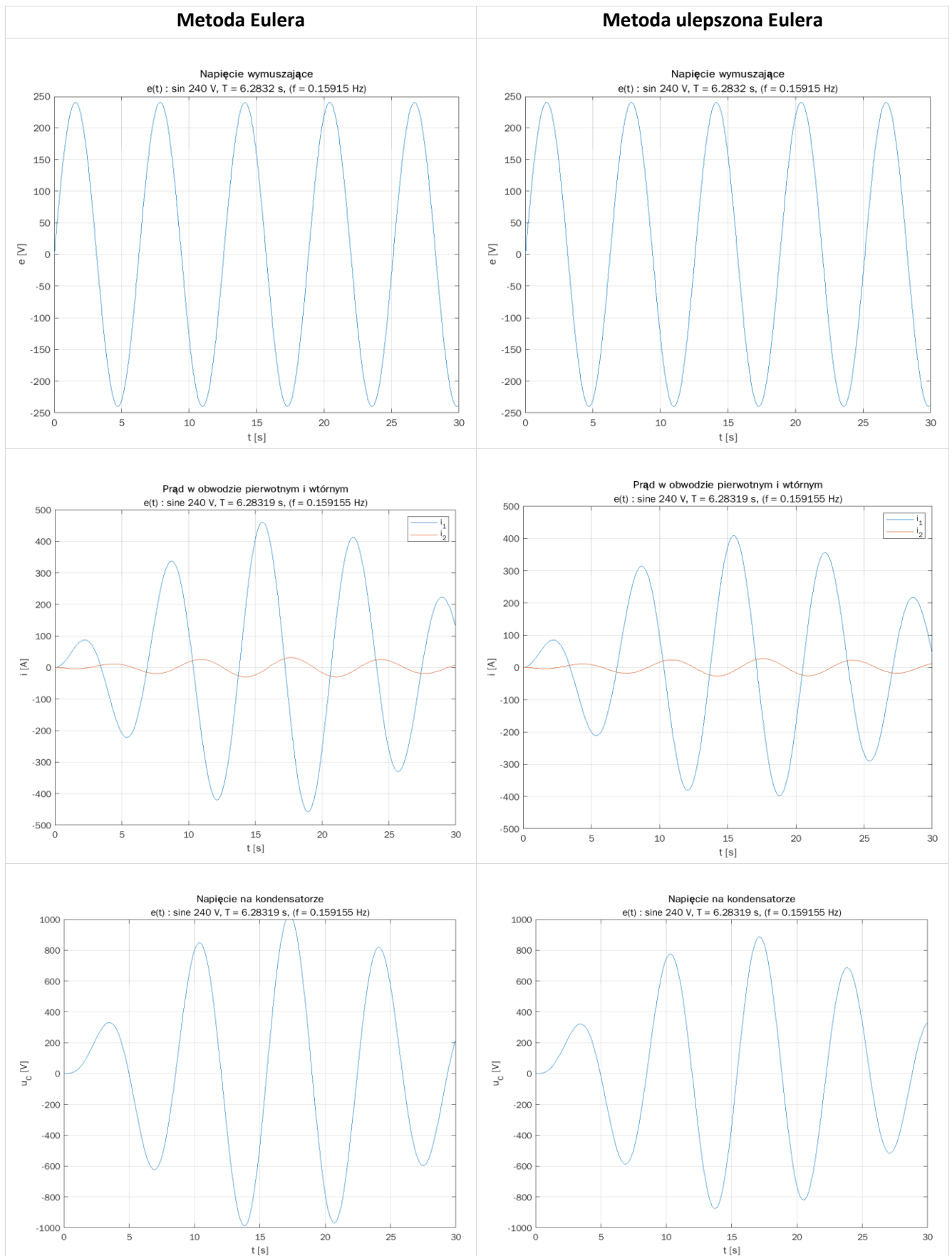
W badaniu obserwuje się przebiegi prądów i_1 i i_2 oraz napięcia u_C uzyskane za pomocą metody Eulera oraz ulepszonej metody Eulera.

5.1 Symulacja przebiegów prądów i napięcia

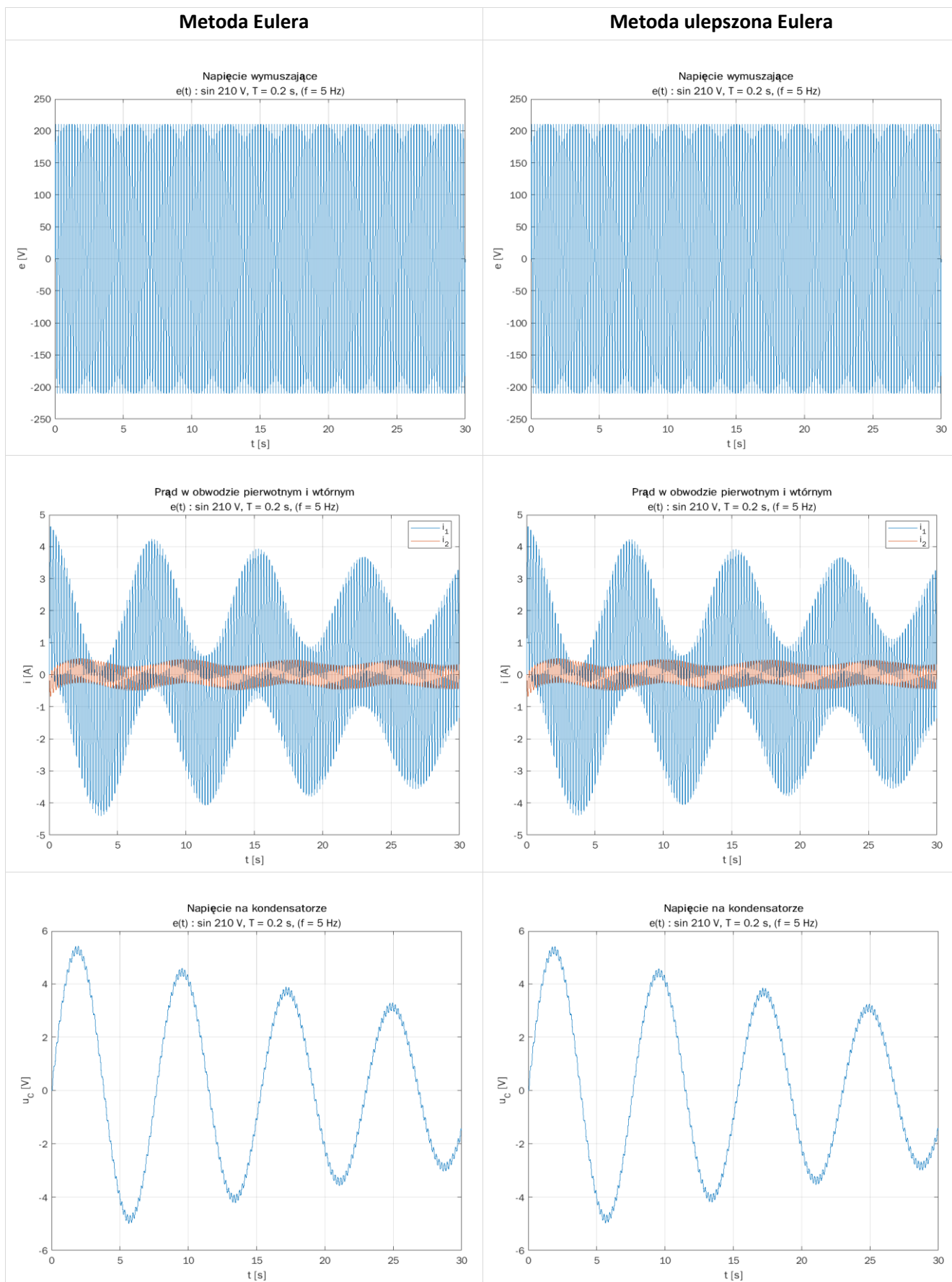
5.1.1 Wymuszenie prostokątne, $U = 120\text{V}$ dla $t < T/2$, oraz 0V dla $t \geq T/2$, $T = 3\text{ s}$



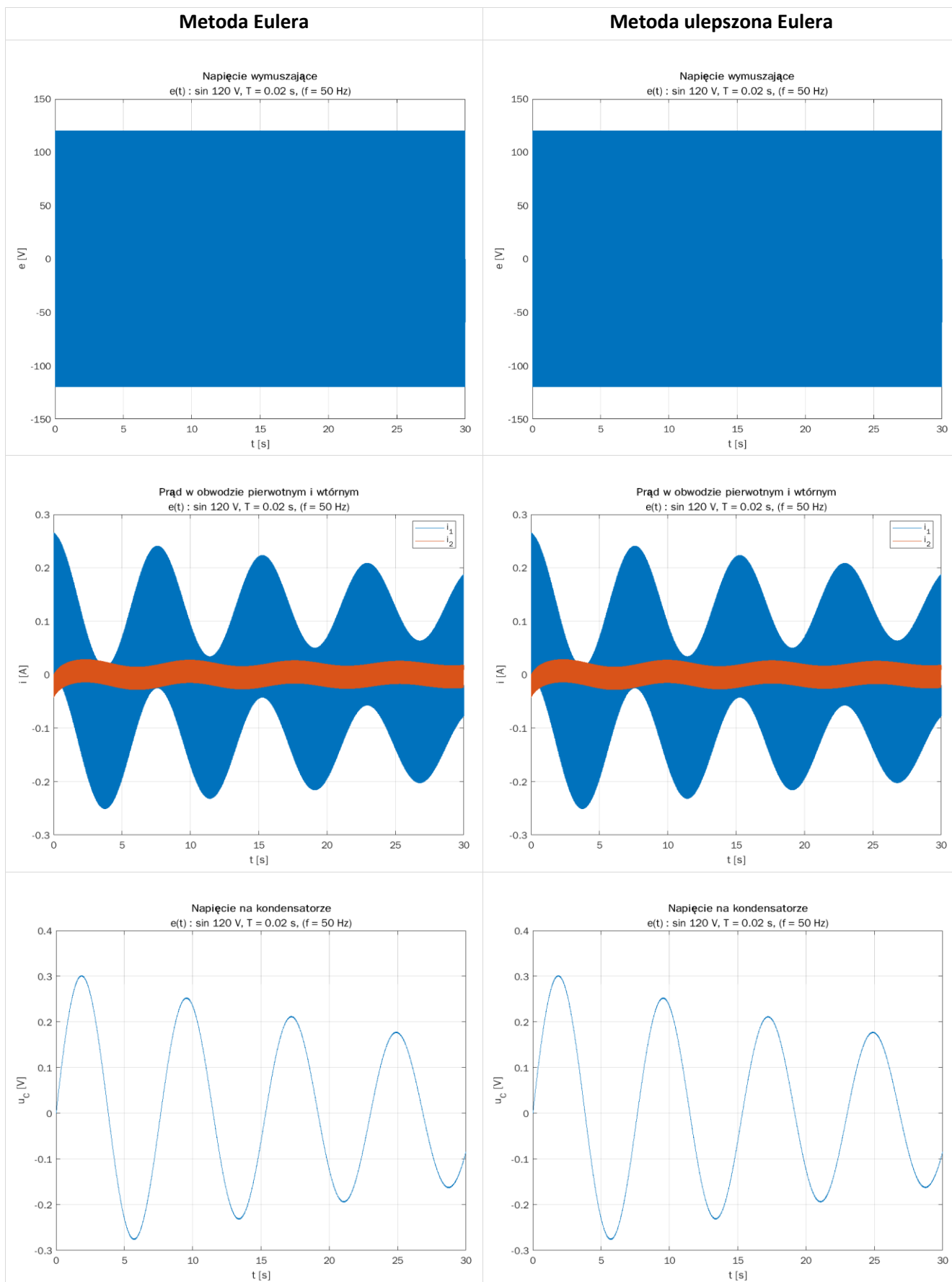
5.1.2 Wymuszenie sinusoidalne, $e(t) = 240 \sin(t)$



5.1.3 Wymuszenie sinusoidalne, $e(t) = 210 \sin(2\pi ft)$, dla $f = 5$ Hz



5.1.4 Wymuszenie sinusoidalne, $e(t) = 120 \sin(2\pi ft)$, dla $f = 50$ Hz

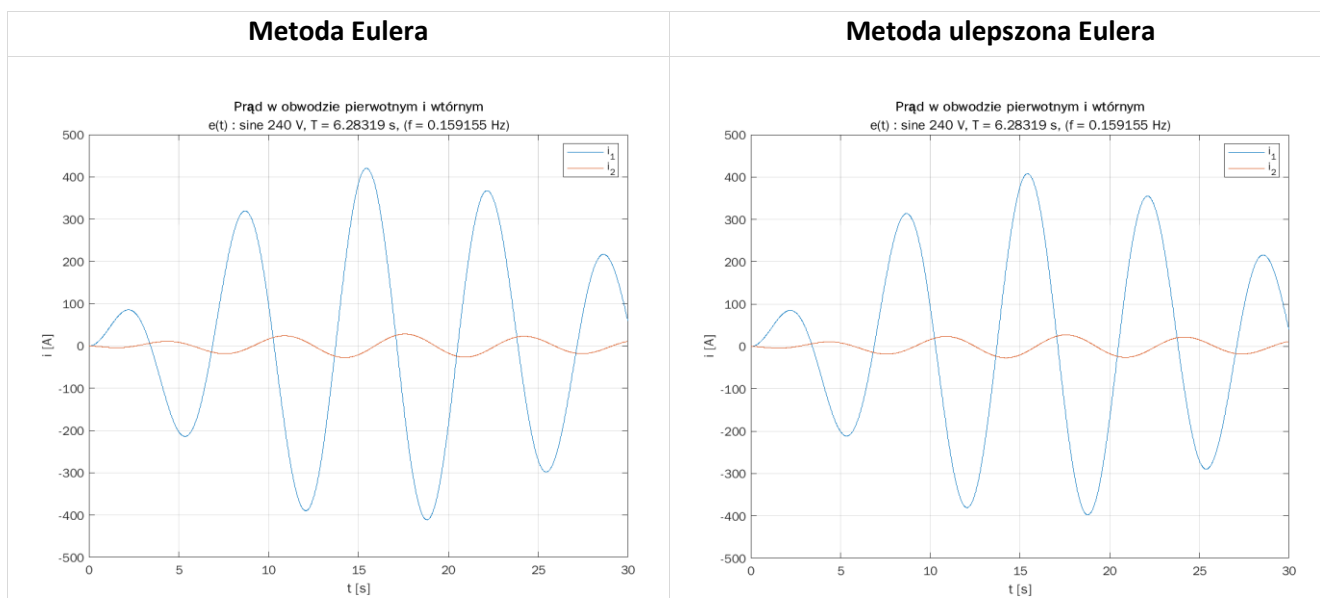


5.2 Omówienie wyników i wnioski

Obserwowane na wykresach przebiegi prądów i napięć wskazują, że model badanego układu elektrycznego został zaimplementowany prawidłowo. Wyniki uzyskane w symulacji z wymuszeniem prostokątnym o napięciu $U = 120\text{V}$ i okresie $T = 3\text{ s}$ (patrz pkt. 5.1.1) potwierdzają, że przebiegi prądów i_1 i_2 są takie same jak na wykresie zamieszczonym w instrukcji do zadania. Także w przypadku sygnału sinusoidalnego o napięciu 210 V i częstotliwości $f = 5\text{ Hz}$ (patrz pkt. 5.1.3) obraz uzyskany z symulatora jest zgodny z tym, który udokumentowano w instrukcji. Ponieważ zamieszczone w instrukcji wykresy można uznać za wiarygodny punkt odniesienia, poparty badaniami laboratoryjnymi, daje to podstawę do uznania symulatora za zrealizowany poprawnie.

Zamieszczone na stronach poprzednich wykresy zostały zaprezentowane w układzie dwuszpaltowym, aby ułatwić porównanie wyników uzyskiwanych dwiema metodami obliczania równań różniczkowych. Wykresy w lewej szpalcie przedstawiają przebiegi wyliczone z wykorzystaniem metody prostej Eulera, a w prawej — z metody ulepszonej Eulera. W przeważającej części symulacji wyniki są bardzo zbliżone. Niemniej, ciekawa różnica ujawnia się przy wymuszeniu sinusoidalnym $e(t) = 240 \sin(t)$. W tym przypadku przebieg stanu nieustalonego obliczony metodą prostą Eulera odbiega od tego, który uzyskuje się metodą ulepszoną — różnica wartości chwilowych sięga tu nawet 10%!

Warto zauważyć, że ów wyraźny defekt ujawnił się dla sygnału o okresie najdłuższym spośród badanych, a ponieważ krok czasowy Δt w symulatorze dobierany jest proporcjonalnie do okresu T , oznacza to że symulacja prowadzona z niską liczbą stanów chwilowych. Powstaje pytanie, czy zmniejszenie kroku czasowego (zwiększenie ilości obliczanych stanów chwilowych) pozwoli ograniczyć różnice w osiąganych wynikach? Ponieważ wszystkie symulacje prowadzono dla kroku czasowego $\Delta t = T/128$ („normal”), zagęszczono próbkowanie do $\Delta t = T/512$ („dense”) i wykonano symulację dla nowych warunków. Po zmianie kroku czasowego wykres prądów z metody prostej Eulera nie różni się już w sposób tak istotny od wykresu z metody ulepszonej — i wygląda następująco:



Opisany przypadek pokazuje, jak dobór odpowiedniego kroku czasowego może wpływać na uzyskiwane w symulatorze wyniki. Wskazuje na przewagę metody ulepszonej Eulera w sytuacjach, gdy dokładność wyników ma szczególnie znaczenie, a niedokładności rzędu kilku procent są po prostu niedopuszczalne.

6 SYMULACJA PARAMETRÓW OBWODOWYCH – CZĘŚĆ 2.

Druga część projektu zakłada wprowadzenie istotnej zmiany w sprzężeniu między obwodami pierwotnym i wtórnym. Indukcja wzajemna nie jest wartością stałą lecz jest nieliniowo zależna od napięcia u_L , tj. na indukcyjności L_I w obwodzie pierwotnym. Tę nieliniową zależność opisano w postaci poniższej tabeli pomiarów napięcia i indukcyjności wzajemnej.

u_L [V]	20	50	100	150	200	250	280	300
M [H]	0,46	0,64	0,78	0,68	0,44	0,23	0,18	0,18

Nieliniowa indukcyjność wzajemna opisana zaledwie 8 punktami musi być zastąpiona ciągłą funkcją $M = f(u_L)$, która wykorzystując zagadnienie interpolacji/aproksymacji będzie zwracać indukcyjność wzajemną dla całego spektrum napięć od 0 do 300V.

W ramach zadania, mają zostać zbadane przebiegi prądów i napięć, z tym że na wykresach należy pokazać wyniki osiągane różnymi metodami przybliżania charakterystyki indukcyjności wzajemnej, mianowicie stosując:

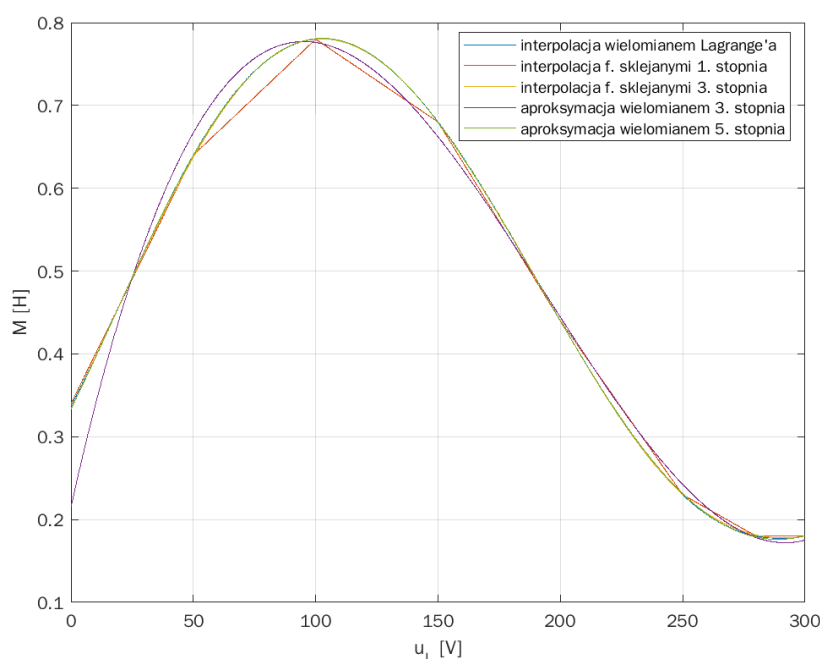
- interpolację wielomianową;
- interpolację funkcjami sklejanymi;
- aproksymację wielomianową z zastosowaniem wielomianu stopnia 3;
- aproksymację wielomianową z zastosowaniem wielomianu stopnia 5.

Przebiegi mają być wykonane dla dwóch różnych napięć wymuszających:

- sygnał sinusoidalny, $e(t) = 240 \sin(2t)$
- sygnał sinusoidalny, $e(t) = 120 \sin(2\pi ft)$, dla $f = 50$ Hz

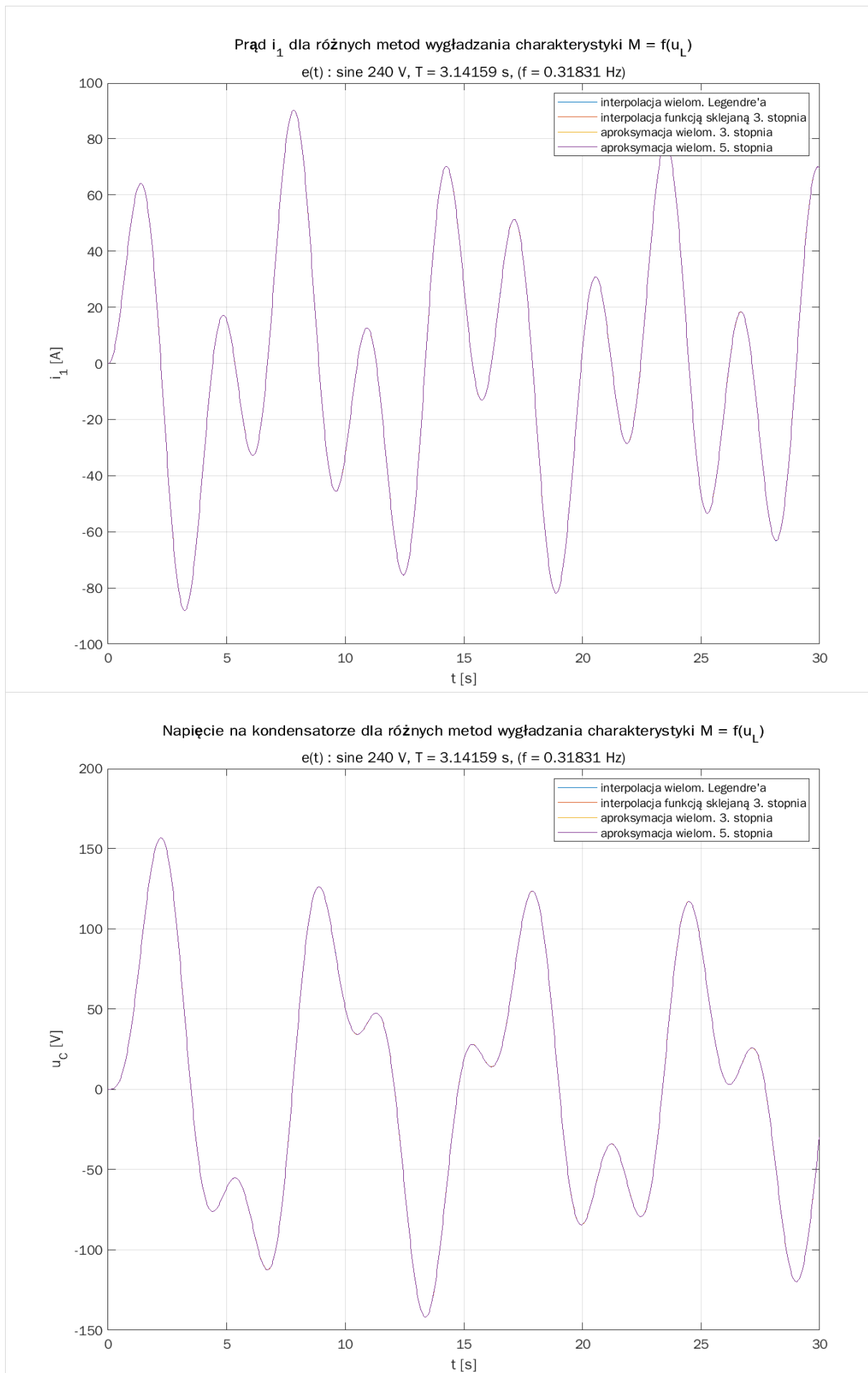
6.1 Porównanie metod interpolacji/aproksymacji charakterystyki

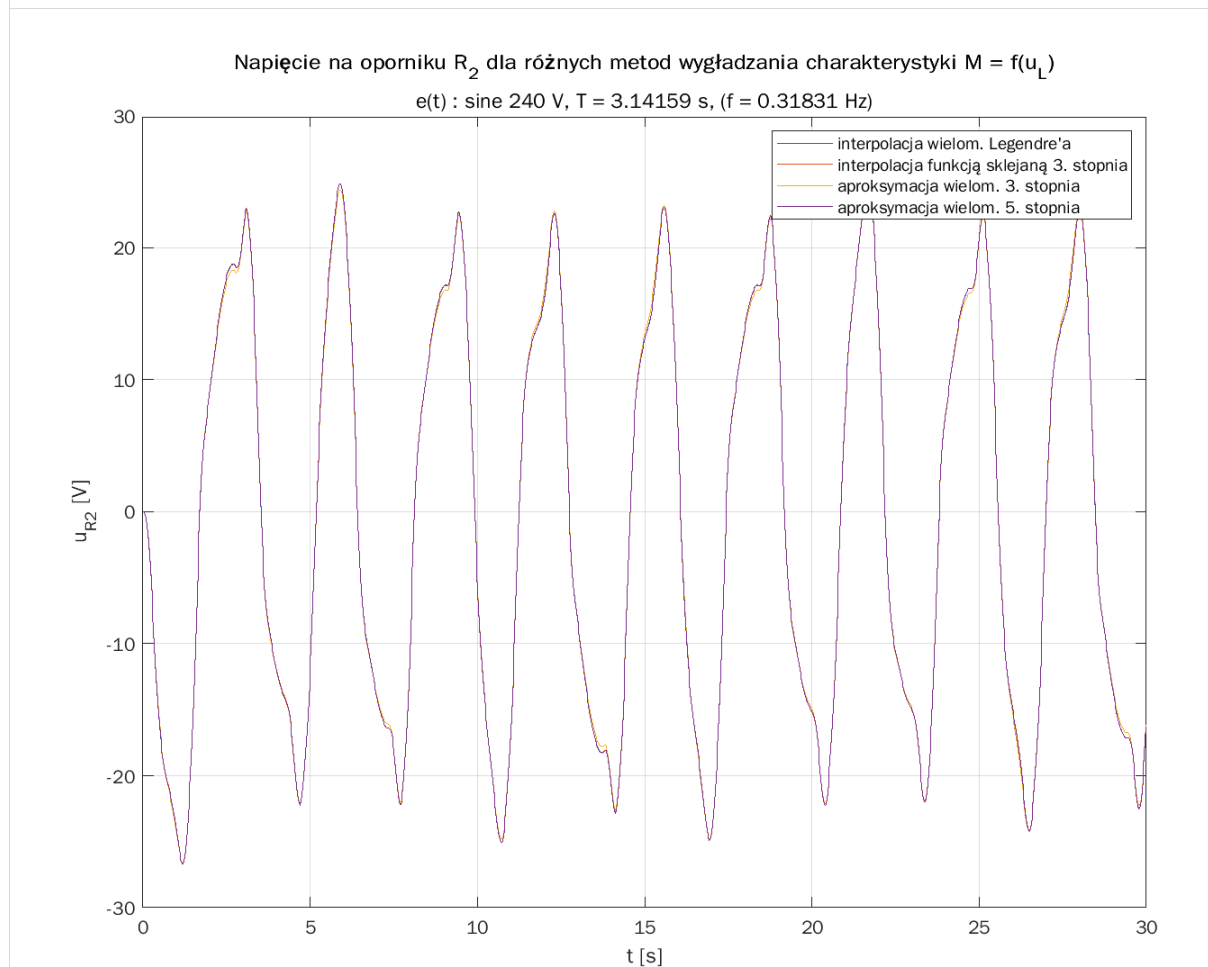
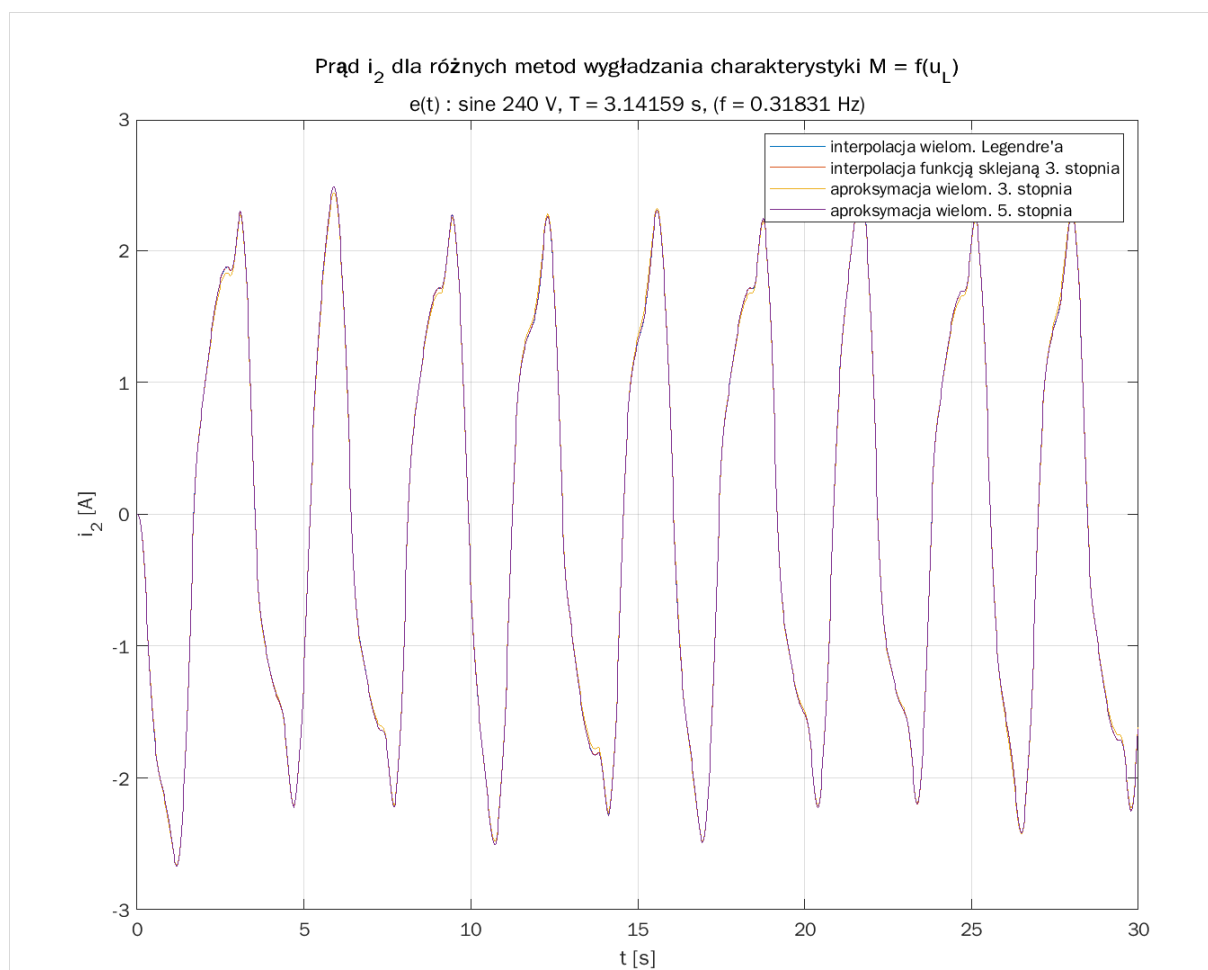
Przed obserwacją przebiegów prądów i napięć, najpierw wykonałem kontrolę efektu zastosowania zaimplementowanych w symulatorze metod interpolacji i aproksymacji. Dla serii danych punktowych określających zależność indukcyjności wzajemnej M od napięcia u_L otrzymuje się następujące warianty charakterystyki $M = f(u_L)$



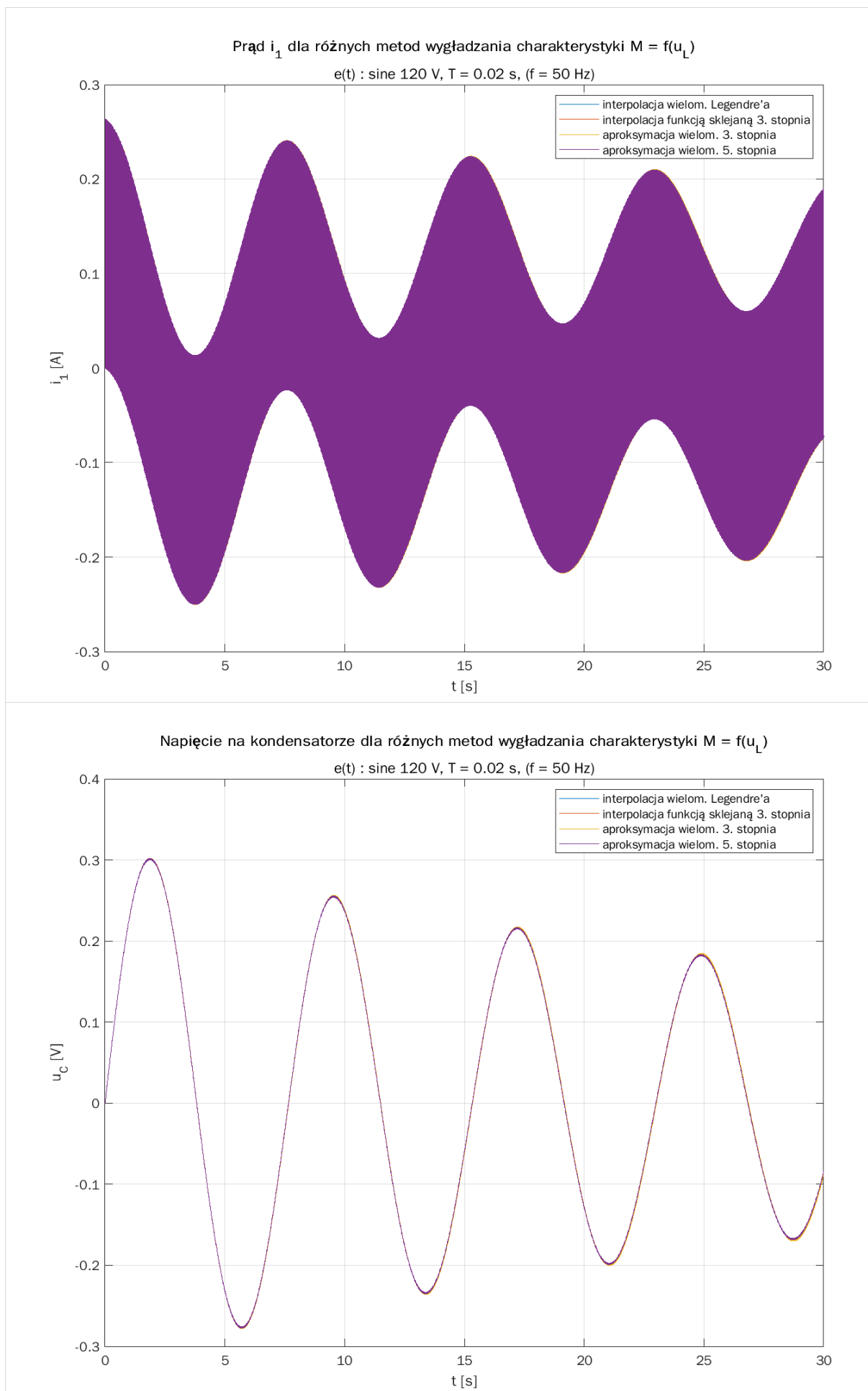
Rys nr 2. Wykres charakterystyki indukcyjności wzajemnej M

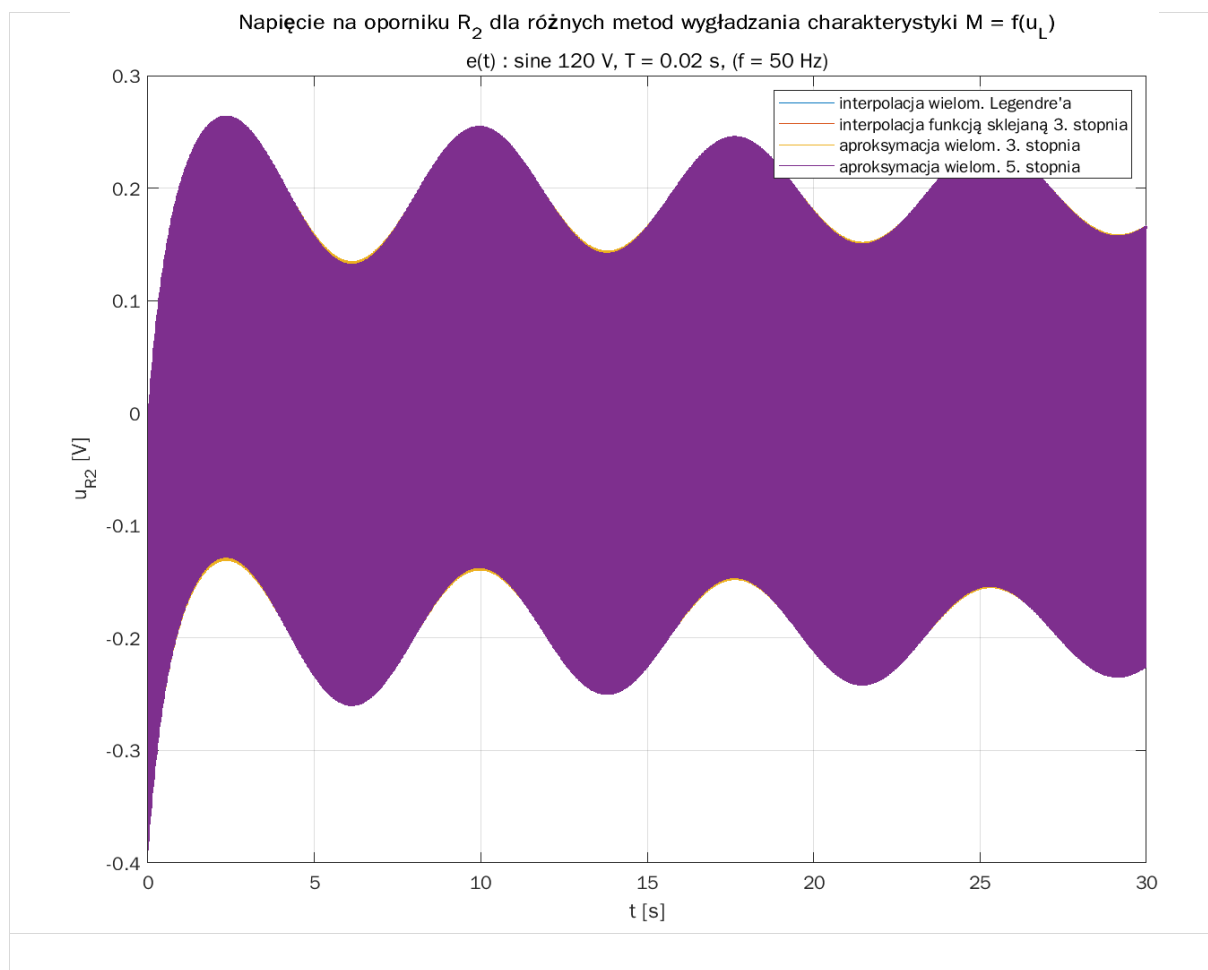
6.1.1 Wymuszenie sinusoidalne, $e(t) = 240 \sin(2t)$





6.1.2 Wymuszenie sinusoidalne, $120 \sin(2\pi ft)$, dla $f = 50$ Hz





6.2 Omówienie wyników i wnioski

Każda z opracowanych metod numerycznego przybliżania charakterystyki $M = f(u_L)$, pracuje w sposób prawidłowy oraz daje zbliżone lub praktycznie identyczne wyniki przebiegów czasowych, o czym świadczy fakt, że przebiegi uzyskiwane każdą metodą pokrywają się na większości wykresów.

Pewna różnica zauważalna jest w przypadku metody aproksymacji wielomianem 3. stopnia na wykresie prądu i_2 w obwodzie wtórnym, jak również napięcia na rezystorze R_2 (te dwie wielkości są zależne wprost). W okolicach „peaków” ujawnia się rozbieżność wyników tej metody, w stosunku do wszystkich pozostałych. Jest to najwyraźniej spowodowane faktem, że wielomian 3. stopnia nienajlepiej przybliża charakterystykę $M = f(u_L)$, w pewnym zakresie napięć, co uwidacznia się na Rys. nr 2, w zakresie 0-20V.

Wśród wszystkich zaimplementowanych metod, zwraca moją uwagę interpolacja wielomianem Lagrange’a, ze względu na dokładność oraz stosunkowo niską złożoność obliczeniową.

7 SYMULACJA PARAMETRÓW OBWODOWYCH – CZĘŚĆ 3.

Trzecie zadanie projektowe polega na wyznaczeniu mocy czynnej tzn. mocy wydzielanej na obu rezystorach w układzie. Scenariusz badań jest tutaj dość obszerny, gdyż należy określić wynik dla wszystkich wymuszeń jak w zadaniu 1, odrębnie dla różnych metod całkowania, a w każdej z nich z podziałem na dwa różne kroki czasowe. Całość należy wykonać dwukrotnie — pierwszy raz w badaniu przyjąć należy stałą indukcyjność wzajemną, a za drugim razem indukcyjność ma być nieliniowa, przyjęta tak, jak w zadaniu 2.

Reasumując rozpatrujemy następujące wymuszenia:

1. sygnał prostokątny, $U = 120V$ dla $t < T/2$, oraz $0V$ dla $t \geq T/2$, $T = 3$ s.
2. sygnał sinusoidalny, $e(t) = 240 \sin(t)$
3. sygnał sinusoidalny, $e(t) = 210 \sin(2\pi ft)$, dla $f = 5$ Hz
4. sygnał sinusoidalny, $e(t) = 120 \sin(2\pi ft)$, dla $f = 50$ Hz

Obliczenia wykonywane są dla dwóch różnych metod całkowania, w tym:

- a) całkowanie metodą złożoną prostokątów
- b) całkowanie metodą złożoną parabol

Wyjaśnienia wymaga niestandardowa w moim projekcie realizacja zagadnienia związanego z krokiem całkowania. Jest on oczywiście taki sam, jak krok przyjęty w szeregach czasowych symulowanych sygnałów, lecz należy mieć na uwadze, że szeregi czasowe generowane są z krokiem zależnym od okresu sygnału. Jest to rozwiązanie, które ma gwarantować optymalny dobór kroku symulacji w zależności od częstotliwości sygnału. Zatem dla sygnału o częstotliwości 50 Hz krok czasowy będzie zupełnie inny, niż np. dla częstotliwości 1 Hz lub mniej. Utrzymanie stałego Δt dla tak zróżnicowanego spektrum sygnałów mogłoby być nadmiernym uproszczeniem, niosącym ryzyko dla dokładności lub nawet stabilności symulacji.

Z uwagi na powyższe, przyjęty w badaniu podział na obliczenia dla dwóch różnych kroków czasowych Δt_1 i Δt_2 należy rozumieć jako dwa odmienne ustawienia wartości względnej dla kroku czasowego. Symulację z krokiem czasowym Δt_1 „bardzo krótkim” i Δt_2 „bardzo długim” zrealizowano przyjmując następujące ustawienia:

- $\Delta t_1 = T / 512$ (ustawienie dla szeregów czasowych „dense”), np. dla $\sin(t)$: $\Delta t_1 = 6,283 / 512 \approx 0,012$ sek.
- $\Delta t_2 = T / 32$ (ustawienie dla szeregów czasowych „light”), np. dla $\sin(t)$: $\Delta t_2 = 6,283 / 32 \approx 0,196$ sek.

Czas „bardzo długi” należy rozumieć jako 16-krotnie mniejszą gęstość próbkowania niż czas „bardzo krótki”.

Scenariusz badań wykonany będzie najpierw dla transformatora o stałej indukcyjności wzajemnej, a za drugim razem — przy nieliniowej indukcyjności wzajemnej, z zaangażowaniem metody interpolacji wielomianem Lagrange’a do uzyskania funkcyjnej zależności $M = f(u)$. W moim symulatorze metoda ta jest domyślną opcją, z uwagi na stosunkowo niską złożoność obliczeniową przy zadowalającej dokładności. Stany chwilowe obliczane są w każdym przypadku metodą ulepszoną Eulera.

7.1 Praktyczne porównanie metod całkowania numerycznego

7.1.1 Moc czynna w układzie z transformatorem o stałej indukcyjności wzajemnej

Wymuszenie	Wynik całkowania metodą złożoną prostokątów		Wynik całkowania metodą złożoną parabol	
	Δt_1	Δt_2	Δt_1	Δt_2
$e(t) = 1V$	0,006243	0,006241	0,006243	0,006242
$E = (120 \text{ dla } t < T/2; 0 \text{ dla } t \geq T/2)$ $T = 3$ sek.	59,586	59,670	59,589	59,548
$e(t) = 240 \sin(t)$	7080,4	7393,9	7080,7	7400,3
$e(t) = 210 \sin(2\pi f t)$ dla $f = 5$ Hz	1,1579	1,1594	1,1579	1,1596
$e(t) = 210 \sin(2\pi f t)$ dla $f = 50$ Hz	0,003786	0,003792	0,003786	0,003792

7.1.2 Moc czynna w układzie z transformatorem o z nieliniową indukcyjnością wzajemną

Wymuszenie	Wynik całkowania metodą złożoną prostokątów		Wynik całkowania metodą złożoną parabol	
	Δt_1	Δt_2	Δt_1	Δt_2
$e(t) = 1V$	0,005518	0,005516	0,005518	0,005517
$E = (120 \text{ dla } t < T/2; 0 \text{ dla } t \geq T/2)$ $T = 3 \text{ sek.}$	57,976	58,052	57,979	57,941
$e(t) = 240 \sin(t)$	5164,7	5431,7	5164,8	5436,7
$e(t) = 210 \sin(2\pi f t)$ dla $f = 5 \text{ Hz}$	0,70529	0,70598	0,70529	0,70606
$e(t) = 210 \sin(2\pi f t)$ dla $f = 50 \text{ Hz}$	0,003530	0,003535	0,003530	0,003535

7.2 Omówienie wyników i wnioski

Uzyskane wyniki całkowania wykazały ogromną zależność mocy czynnej od częstotliwości sygnału wymuszającego. Jeśli chodzi o porównanie metod całkowania, to na uwagę zasługują bardzo podobne wyniki — w przypadku gdy krok czasowy jest bardzo mały (czyli sygnał próbkowany jest gęsto) rozbieżności są znikome i nie przekraczają ułamek promila. Dla dużego kroku czasowego, również nie widać istotnych różnic pomiędzy metodami — rozbieżności nie przekraczają 1%.

Opisane wyżej prawidłowości są zgodne z oczekiwaniami. Im mniejszy krok czasowy, tym bardziej nieistotny staje się błąd całkowania najprostszą metodą, tj. metodą złożoną prostokątów.

8 SYMULACJA PARAMETRÓW OBWODOWYCH – CZĘŚĆ 4.

W ostatniej, czwartej części projektu, symulator zostanie użyty do wyliczenia metodami iteracyjnymi podanej funkcji celu, mianowicie przy jakiej częstotliwości napięcia wymuszającego układ pracuje z zadaną mocą czynną. Symulacji zostaje poddany układ w wariancie ze stałą indukcyjnością wzajemną. Zadany sygnał wymuszający to sinusoidalne napięcie $U = 100V$. Poszukujemy, przy jakiej częstotliwości f , moc czynna osiągnie dokładnie $P = 406 W$.

Proces poszukiwania dokładnie takiego wyniku polega na iteracyjnej zmianie częstotliwości napięcia wymuszającego i wznowianiu całej procedury obliczeniowej tak długo, aż poszukiwana wartość mocy zostanie osiągnięta z założoną precyzją. Wymaganą precyzję oszacowania wartości pierwiastka ustawiono w symulatorze na $\varepsilon = 0,0001$.

Do obliczania funkcji celu użyte zostaną trzy metody numeryczne:

- metoda bisekcji
- metoda siecznych
- metoda Newtona

Wyniki obliczeń, precyzję osiąganą przez rozwiązania oraz różnice złożoności obliczeniowej poszczególnych metod przedstawia poniższa tabela.

8.1 Zestawienie wyników obliczania funkcji celu

Nazwa metody	Wartość rozwiązania częstotliwość f	Wartość funkcji celu F	Liczba iteracji metody	Liczba obliczeń mocy
Metoda bisekcji	0,18530	-0,02729	9	27
Metoda siecznych	0,18488	3,54870	4	10
Metoda Quasi-Newtona	0,18531	-0,05098	5	23

8.2 Omówienie wyników i wnioski

Jeśli chodzi o zbieżność, badanie potwierdziło spodziewaną przewagę metody siecznych i Quasi-Newtona, nad metodą bisekcji. Liczba koniecznych iteracji w metodzie bisekcji była około 2-krotnie większa niż w pozostałych. Ilość koniecznych obliczeń mocy również jest najwyższa w tym przypadku. Zależność między liczbą iteracji n a liczbą obliczeń mocy wynosi $3n$.

Metoda siecznych okazała się w tym przypadku najlepsza, pod każdym względem. Oprócz niskiej liczby iteracji wyróżnia ją również mała ilość koniecznych obliczeń mocy. Zważywszy na to jak „kosztowne obliczeniowo” jest scałkowanie numeryczne dużych szeregów czasowych (czyli wyliczenie mocy), każda oszczędność pod tym względem jest bardzo pożądana, dlatego metoda siecznych niewątpliwie jest zwycięzcą wydajności. Zależność między liczbą iteracji n a liczbą obliczeń mocy można określić w niej ogólnym wzorem $2 + 2n$.

Znana z dobrej zbieżności metoda Quasi-Newtona, przegrała w porównaniu z metodą siecznych — głównie ze względu na sporą liczbę koniecznych obliczeń mocy. Proces obliczania poprawki w każdym kroku iteracyjnym jest w niej złożony. Wymaga to zarówno obliczania wartości funkcji jak i jej pochodnej, co w przypadku funkcji wyrażonej w sposób dyskretny wiąże się właściwie z potrójnym obliczaniem mocy w jednym kroku iteracyjnym. Ustalenie warunku początkowego, to również nakład pracy obliczeniowej. Ostatecznie, zależność między liczbą iteracji n a liczbą obliczeń mocy określa w tym przypadku wzór $8 + 3n$.