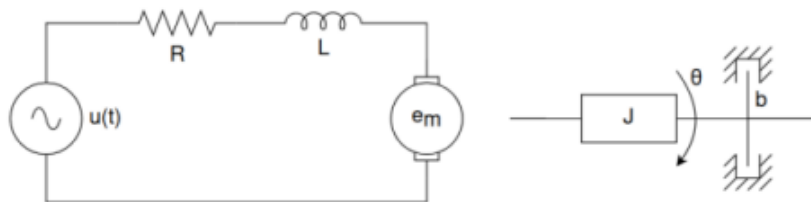


# Sprawozdanie z projektu z przedmiotu Metody Modelowania Matematycznego

## Opis badanego układu oraz treść zadania

Projekt 2. Dany jest model silnika elektrycznego



gdzie moment obrotowy wału zależny jest od prądu w obwodzie elektrycznym jako  $T = K_T i$ , natomiast wsteczna siła elektromotoryczna generowana w obwodzie twornika dana jest jako  $e_m = K_e \dot{\theta}$ . Należy opracować model oraz zaimplementować symulację układu pozwalającą na wykreślenie prądu płynącego w obwodzie oraz prędkości kątowej wału silnika. Symulator powinien umożliwiać pobudzenie układu przynajmniej trzema rodzajami sygnałów wejściowych (prostokątny o skończonym czasie trwania, trójkątny, harmoniczny). Symulator powinien umożliwiać zmianę wszystkich parametrów układu oraz sygnałów wejściowych.

## Użycie

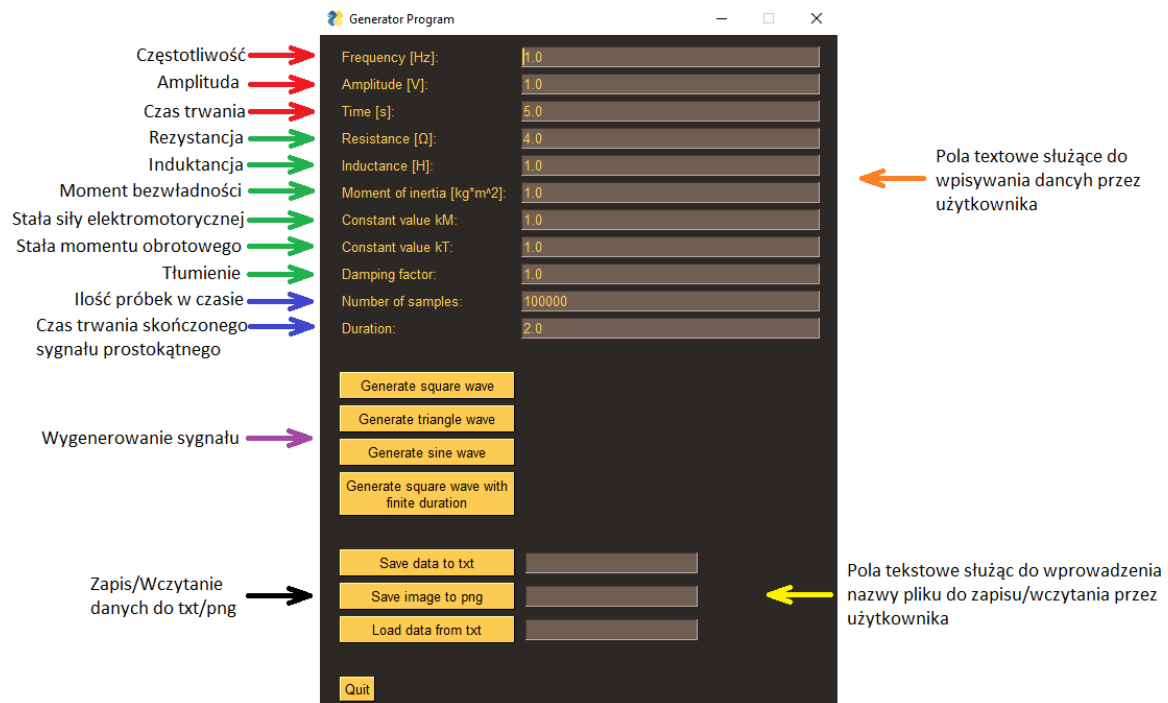
Program jest w większości odporny na błędy użytkownika. Gdy użytkownik wprowadzi coś innego niż liczba, to program wyświetli popup, który zawiadomi o nieprawidłowości.

Oczekuje się, że użytkownik nie będzie wprowadzał zbyt dużych lub zbyt małych wartości niektórych sygnałów, gdyż pamięć komputera może ulec przeciążeniu i program przestanie działać zgodnie z założeniami.

Program posiada cztery generacje sygnałów oraz dla każdego z nich generuje się wykres prądu w obwodzie oraz prędkość kątową wału.

Aby zapisać dane do pliku tekstowego należy najpierw wygenerować wykres, gdyż dane najpierw muszą zostać zatwierdzone. Gdy program wykryje nieprawidłowość w wprowadzonej nazwie pliku (np plik nie istnieje) wyświetli komunikat o nieprawidłowości.

## Interfejs



### Strzałki:

**Czerwone** - dane niezbędne do wygenerowania sygnału

**Zielone** - dane niezbędne w badanym układzie

**Niebieskie** - dane niezbędne do wygenerowania próbek czasu oraz sterowania czasem

**Fioletowa** - przyciski służące do wygenerowania wykresu sygnału, prądu w obwodzie oraz prędkości kątowej

**Pomarańczowa** - pola tekstowe służące do wpisywania danych przez użytkownika

**Żółta** - pola tekstowe służące do wprowadzenia nazwy pliku do zapisu/wczytania przez użytkownika

## Kod

### Użyte biblioteki

```
import numpy as np
import matplotlib.pyplot as plt
import PySimpleGUI as sg
from scipy import signal
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
```

## Ważniejsze funkcje

```
def response(name, sig, L, R, Km, KT, b, J, t):
    figure, axis = plt.subplots(3, figsize=(14, 11))
    I = [0]
    w = [0]
    for i in range(len(sig)):
        if i + 1 == len(sig):
            break
        else:
            ts = t[i + 1] - t[i]
            I.append(I[i] + ts * (sig[i] / L - (R * I[i]) / L - (Km * w[i]) / L))
            w.append(w[i] + ts * ((KT * I[i]) / J - (b * w[i]) / J))
```

Na początku definiowane są warunki początkowe oraz zerowanie tablicy, następnie wywołana jest pętla for aby obliczyć ze wzoru wartości sygnału prądu oraz prędkości kątowej

$$\frac{I[t+1] - I[t]}{T_s} = \frac{U(t)}{L} - \frac{R}{L} I(t) - \frac{K_m}{L} \cdot \omega(t)$$

$$\frac{\omega[t+1] - \omega[t]}{T_s} = \frac{K_T}{J} \cdot I(t) - \frac{b}{J} \cdot \omega(t)$$

$T_s$  – stała czasu

## Funkcja pozwalająca stworzyć przestrzeń do narysowania wykresu

```
def draw_figure(canvas, figure):
    figure_canvas_agg = FigureCanvasTkAgg(figure, canvas)
    figure_canvas_agg.draw()
    figure_canvas_agg.flush_events()
    figure_canvas_agg.get_tk_widget().pack(side="top", fill="both", expand=1)
    return figure_canvas_agg
```

## Stworzenie okna

```
window = sg.Window("Generator Program", layout, location=(500, 50), size=(500, 650), finalize=True)
```

