

Cyfrowe przetwarzanie sygnałów i obrazów

Przetwarzanie i analiza sygnału EKG

Autorki: Barbara Parzonka 263955, Joanna Zoglowek 264452

Ćwiczenie 1. Napisz skrypt w Pythonie/Matlabie umożliwiający wczytywanie i wizualizację badanych sygnałów. Program powinien umożliwiać obserwowanie wycinka sygnału dla zadanego przedziału czasowego, skalowanie osi wykresów i ich opis oraz zapis dowolnego wycinka sygnału do pliku o podanej nazwie.

Opis realicacji zadania

Wywołanie programu poprzez *Run>Run All Cells* wiąże się z wybraniem domyślnych ustawień dotyczących wczytywanego z pliku tekstowego sygnału (sygnał wczytany z pliku *ekg100.txt*, częstotliwość próbkowania 360 Hz). Jednym z ustawień dotyczących sygnału jest pole wyboru tego, czy pierwsza kolumna pliku tekstowego ma być traktowana jako czas poboru sygnału.

Program jest podzielony na trzy komórki - wywołanie pierwszej z nich tworzy i inicjalizuje część zmiennych potrzebnych w programie oraz odpowiada za interfejs umożliwiający użytkownikowi ustawienie wspomnianych wyżej właściwości sygnału (nazwa pliku, częstotliwość, stwierdzenie, czy pierwsza kolumna to czas).

Wywołanie drugiej komórki powinno odbyć się po uprzednim wywołaniu pierwszej komórki. Ta część jest odpowiedzialna za odczytanie odpowiedniego pliku tekstowego. Zadaniem trzeciej części jest narysowanie wykresu i umożliwienie interakcji z wykresem - w tym zmianę wyświetlanego przedziału i zapis sygnału do pliku o zadanej nazwie.

```
In [1]: %run methods.ipynb
from collections.abc import Mapping
from ipywidgets import interact, interactive, fixed, interact_manual, Layout
import ipywidgets as widgets
from IPython.display import display
import matplotlib.pyplot as plt
import asyncio
import sys
xarray=[]
valuesArray=[]
numberOfCases=1
datapath='./data/'
bottomLimit=0
upperLimit=len(xarray)
frequency=360
filename=datapath+"ekg100.txt"
hasFirstTime=False
changed=False
```

```

fileOption=widgets.Dropdown(
    options=['ekg100.txt', 'ekg1.txt', 'ekg_noise.txt'],
    description='Wybierz plik z danymi: \n',
    disabled=False
)

freq=widgets.IntText(
    value=360,
    min=1,
    description='Częstotliwość próbkowania (Hz):',
    disabled=False,
)
freq.style.description_width = 'auto'

timeFirst=widgets.Checkbox(
    value=False,
    description='W pierwszej kolumnie znajduje się czas',
    disabled=False
)
timeFirst.style.description_width = 'auto'

out = widgets.Output(layout={'border': '1px solid black'})

domain_slider=widgets.FloatRangeSlider(
    value=[0,len(xarray)* 1 / frequency],
    min=0,
    max=len(xarray)* 1 / frequency,
    step=0.1,
    description='Przedział czasowy',
    continuous_update=False,
    layout=Layout(width='80%')
)

file_name_input = widgets.Text(
    placeholder='Podaj nazwę pliku',
    description='Nazwa pliku, do którego ma zostać zapisana wartość:',
    disabled=False
)

def on_save_button_clicked(b):
    file_name = file_name_input.value.strip()
    if file_name:
        file_path = datapath+file_name + ".txt" # Assuming you want to save as text
        save_values_to_file(file_path)
    else:
        with out:
            print('Nazwa pliku jest nieodpowiednia')
    plt.xlim(1,3)

save_button = widgets.Button(
    description='Zapisz do pliku'
)

```

```

save_button.on_click(on_save_button_clicked)

def change_values(frequency_r, filename_r, hasFirstTime_r):
    global frequency, filename, hasFirstTime
    frequency=int(frequency_r)
    filename=datapath+filename_r
    hasFirstTime=hasFirstTime_r
    domain_slider.max=len(xarray)* 1 / frequency
    domain_slider.value=[0, frequency]
    zadanie1(frequency=frequency, filepath=filename, hasFirstTime=hasFirstTime)

def plotEKG(domain):
    global bottomLimit, upperLimit, numberOfCases
    if domain[1]>0:
        bottomLimit=int(domain[0]*frequency)
        upperLimit=int(domain[1]*frequency)
    else:
        bottomLimit=0
        upperLimit=len(xarray)
    plt.figure(figsize=(15, 3 * numberOfCases)) # Ustawienie większego rozmiaru fi
    plt.subplots_adjust(hspace=1.0) # Większe odstępy między wykresami

    for i in range(numberOfCases):
        plt.subplot(numberOfCases, 1, i+1)
        plt.plot(xarray[bottomLimit:upperLimit], valuesArray[i][bottomLimit:upperLi
        plt.xlabel('Czas [s]')
        if i== int(numberOfCases/2):
            plt.ylabel('Amplituda [mV]')
            plt.legend(loc='upper right')
            plt.grid(True)
    plt.tight_layout() # Automatyczna regulacja odstępów
    plt.show()

def zadanie1(frequency=frequency, filepath=filename, hasFirstTime=hasFirstTime):
    global numberOfCases
    #change_values(frequency, filename, hasFirstTime)
    if frequency<=0:
        with out:
            print('Podano nieodpowiednia czestotliwosc')
    else:
        (red, cases)=readFile(filepath, frequency, hasFirstTime)
        if red:
            numberOfCases= cases
            return True
        else:
            exit()

def save_values_to_file(file_path):
    global bottomLimit,upperLimit, numberOfCases
    try:
        with open(file_path, "w") as file:
            for i in range (bottomLimit,upperLimit):
                pom=''

```

```

        for j in range(numberOfCases):
            pom+=str(float(valuesArray[j][i]))+" "
            file.write(pom+"\n")
    with out:
        print("Zapisano w pliku o ścieżce:", file_path)
except Exception as e:
    with out:
        print("Podczas zapisu doszło do wyjątku: ", e)

```

```

In [2]: interact(change_values, frequency_r=freq, filename_r=['ekg100.txt', 'ekg1.txt', 'ekg10.txt'],
interact(plotEKG, domain=domain_slider)
display(file_name_input)
display(save_button)
display(out)

```

```

interactive(children=(IntText(value=360, description='Częstotliwość próbkowania (Hz):', style=DescriptionStyle...
interactive(children=(FloatRangeSlider(value=(0.0, 0.0), continuous_update=False, description='domain', layout...
Text(value='', description='Nazwa pliku, do którego ma zostać zapisana wartość:', placeholder='Podaj nazwę pliku...
Button(description='Zapisz do pliku', style=ButtonStyle())
Output(layout=Layout(border_bottom='1px solid black', border_left='1px solid black', border_right='1px solid black',
border_top='1px solid black'))

```

In []: