

imię i nazwisko: nr indeksu:

1	2	3	Σ

4. FILTRACJA SYGNAŁÓW

ZADANIA DO SPRAWOZDANIA – 24 maja 2019

Uwaga: Za sprawozdanie można uzyskać maksymalnie 3 punkty. Rozwiązania zadań (w formie pojedynczego pliku MATLABa o nazwie `imię_nazwisko_lab4.m`) należy oddać (przysłać mailem na adres `L.Blaszczyk@mini.pw.edu.pl`) najpóźniej do piątku 31 maja do godziny 8:59.

Zadanie 1. Prosty system, który ma na celu sprawić, że dźwięk będzie brzmiał jakby był odtwarzany w starym radiu (albo w czymś dość podobnym ☺) można zaprojektować jako filtr, który przetwarza sygnał w następujący sposób:

$$Y(f) = H(f) \cdot X(f),$$

gdzie X i Y to transformaty Fouriera na wejściu i wyjściu systemu, a transmitancja filtru H ma postać

$$H(f) = \begin{cases} \frac{k_1}{2\pi i \cdot \tau_1 f + 1} & \text{dla } f \leq 300\text{Hz}, \\ \frac{k_2}{2\pi i \cdot \tau_2 f + 1} & \text{dla } 300\text{Hz} < f \leq 1000\text{Hz}, \\ 0 & \text{dla } 1000\text{Hz} < f, \end{cases}$$

gdzie $k_1 = 0,01$, $k_2 = 0,35$, $\tau_1 = 10^{-2}$, $\tau_2 = 10^{-4}$.

Zaimplementuj ten system w pliku skryptowym `imię_nazwisko_lab4.m`. Transformaty Fouriera należy przybliżać za pomocą funkcji `fft`. Dźwięk można filtrować w całości (jako pojedynczy wektor), można też podzielić go na ramki czasowe o określonej długości (jak na zajęciach) i filtrować osobno każdą ramkę. Oba rozwiązania są poprawne.

Zadanie 2. Sprawdź poprawność działania skryptu z zadania 1. Możesz go przetestować na pliku `speech_dft.wav`, który jest domyślnie dostępny w MATLABie, możesz też skorzystać z własnego nagrania (w tym przypadku należy dołączyć plik z nagraniem do maila). Plik `speech_dft.wav` można wgrać za pomocą kodu

```
[audio, Fs] = audioread('speech_dft.wav');
```

i wektor `audio` zawiera sygnał dźwięku, a zmienna `Fs` częstotliwość próbkowania.

Zadanie 3. Porównaj (na osobnych wykresach) widmo amplitudowe i fazowe sygnału na wejściu i wyjściu systemu. Jakiego rodzaju filtrem jest H (dolno-/pasmowo-/górnoprzepustowym)?