Uniwersytet Zielonogórski
Instytut Sterowania i Systemów Informatycznych

Laboratorium cyfrowego przetwarzania i kompresji danych

Filtracja cyfrowa I.

Cele ćwiczenia

Celem ćwiczenia jest zapoznanie się z projektowaniem filtrów cyfrowych.

Uwagi do ćwiczenia

Część wykonanych zadań może być wykorzystana w kolejnych ćwiczeniach, więc proponowane jest przechowywanie wyników na potrzeby kolejnych ćwiczeń.

Polecenia języka Python

Zapoznać się z następującymi poleceniami w języku Python: `np.arange`, `np.linspace`, `np.sin`, `np.pi`, `plt.figure`, `plt.plot`, `plt.stem`, `plt.grid`, `plt.xlabel`, `plt.ylabel`, `plt.legend`, `np.fft.fft`, `np.abs`, `np.random.rand`, `signal.butter`, `signal.lfilter`, `signal.dlti`, `signal.dimpulse`, `signal.convolve`.
Jeśli jest to możliwe, użyj powyższych poleceń do implementacji rozwiązań poniższych zadań.

Zadania do wykonania

1. Wygenerować sumę przebiegów sinusoidalnych według następującego wzoru

$$y(t) = \sum_{i=0}^{N} A_i \sin(2\pi f_i t + \phi_i), \tag{1}$$

gdzie: i - numer przebiegu, t - czas, A_i - amplituda sygnału sinusoidalnego $[V]$, f_i - częstotliwość sygnału sinusoidalnego [Hz], ϕ_i - przesunięcie fazowe sygnału $[^o]$ dla wartości podanych w poniższej tabeli z uwzględnieniem: częstotliwości próbkowania

Nr. przebiegu	$A_i[V]$	$f_i[Hz]$	$\phi_i[^o]$
1	230	50	10
2	115	100	20
3	75	250	30
4	35	400	40
5	15	800	50

$F_s = 2400$[Hz], liczba próbek $L = 240$. Narysować przebieg czasowy oraz widmo amplitudowe sygnału. Wynikiem ma być wykres 1 oraz 2.

2. Zaprojektować filtr Butterwortha dolnoprzepustowy, górnoprzepustowy, środkowoprzepustowy oraz środkowozaporowy z wykorzystaniem funkcji `signal.butter` dla następujących częstotliwości odcięcia

   - **Filtr dolnoprzepustowy** - 100[Hz], 500[Hz], 1000[Hz],
   - **Filtr górnoprzepustowy** - 100[Hz], 500[Hz], 1000[Hz],
   - **Filtr środkowoprzepustowy** - [300,500][Hz], [600,800][Hz], [900,1100][Hz],
   - **Filtr środkowozaporowy** - [300,500][Hz], [600,800][Hz], [900,1100][Hz],

Parametr funkcji `signal.butter` określający częstotliwość odcięcia należy podać jako wartość znormalizowaną z przedziału od 0 do 1 gdzie wartość 1 odpowiada połowie częstotliwości próbkowania. Wartość znormalizowaną można obliczyć z następującego wzoru

$$Wn = \frac{F_c}{0.5F_s}, \tag{2}$$

gdzie F_s częstotliwością próbkowania, F_c częstotliwość odcięcia. Jako rząd filtru przyjąć wartości odpowiednio: 2, 4, 6, 8, 10. Po zaprojektowaniu filtru wykonać filtrację oraz sprawdzić wyniki według następującego schematu:

Krok 1 - Wykonać filtrację z wykorzystaniem funkcji `signal.lfilter`

Krok 2 - Wyznaczyć widmo amplitudowe uzyskanego sygnału

Krok 3 - Na jednym wykresie porównać widmo amplitudowe uzyskanego przebiegu z przebiegiem z zdania 1.

Krok 4 - Wyznaczyć odpowiedź impulsową zaprojektowanego filtru z wykorzystaniem funkcji `signal.dimpulse`.

Krok 5 - Wykonać filtrację z wykorzystaniem funkcji `signal.convolve`.

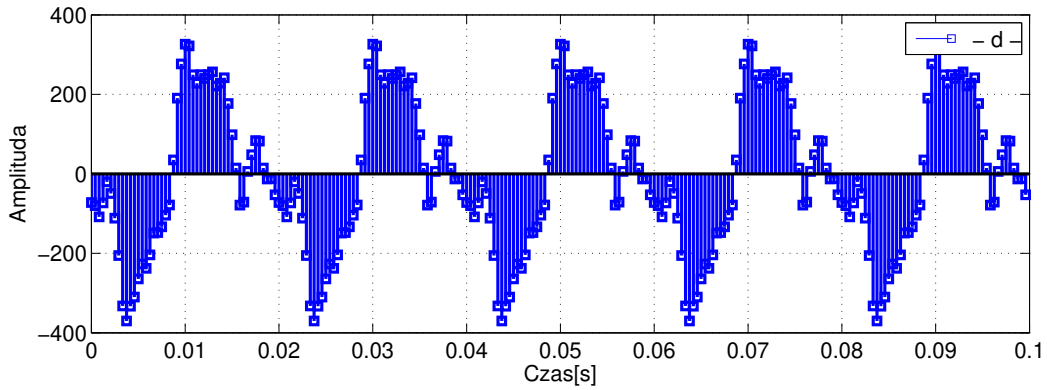
Krok 6 - Wyznaczyć widmo amplitudowe uzyskanego sygnału.

Krok 7 - Na jednym wykresie porównać widmo amplitudowe uzyskanego przebiegu z przebiegiem z zdania 1.

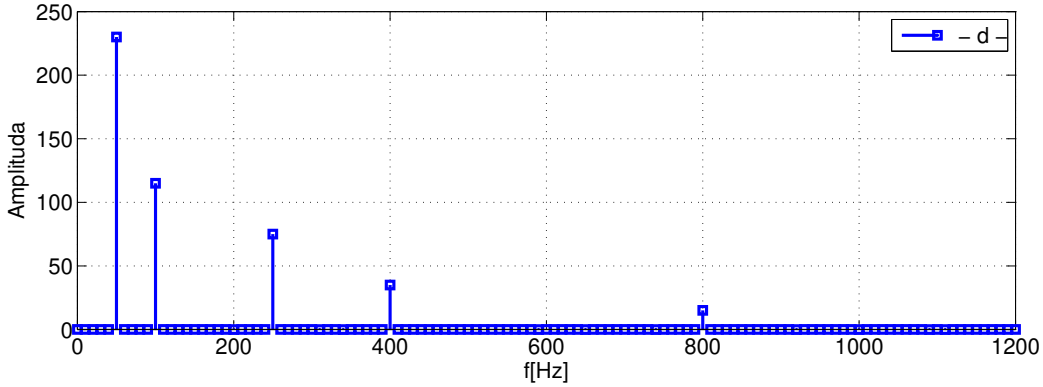
Krok 8 - Na jednym wykresie porównać widma amplitudowe dla sygnałów z kroku 3 oraz 7.

Wynikiem ma być wykres od 3 do 5 dla przypadku filtru dolnoprzepustowego 100[Hz] o rzędzie 2.

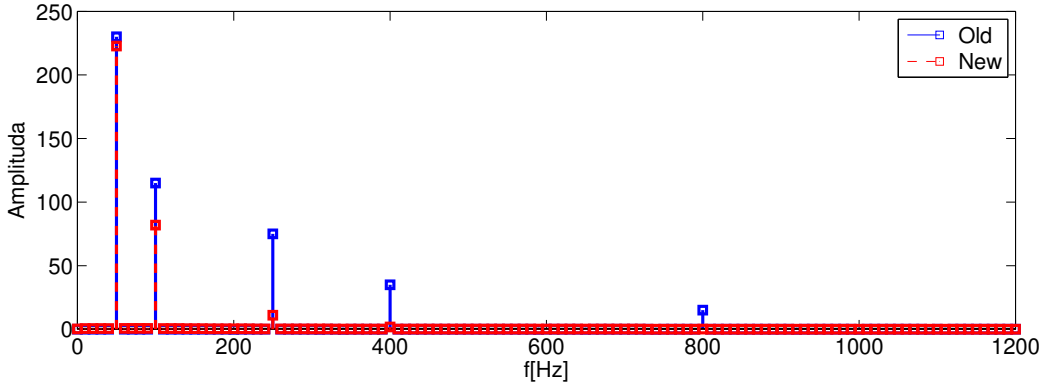
3. Bazując na zadaniu 2 przeanalizować wpływ wartości rzędu filtru na charakterystykę częstotliwościową. Na jednym wykresie porównać charakterystyki poszczególnych filtrów dla wartości rzędu filtru z zadania 2.



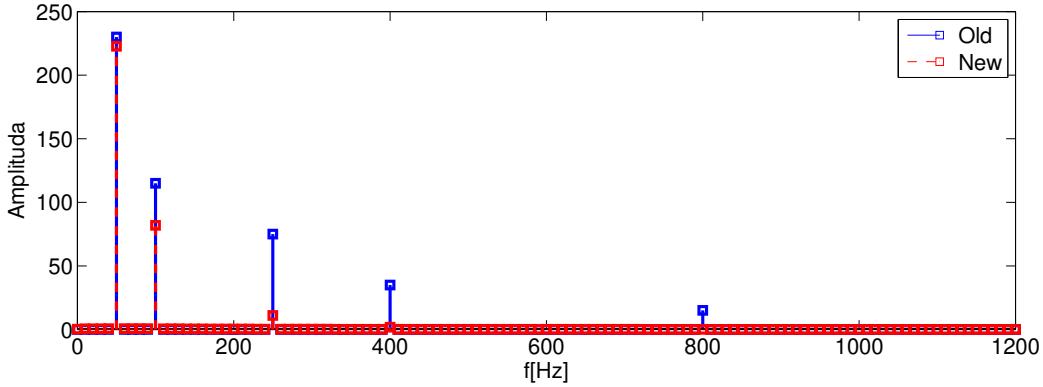
Wykres. 1: d – przebieg dyskretny



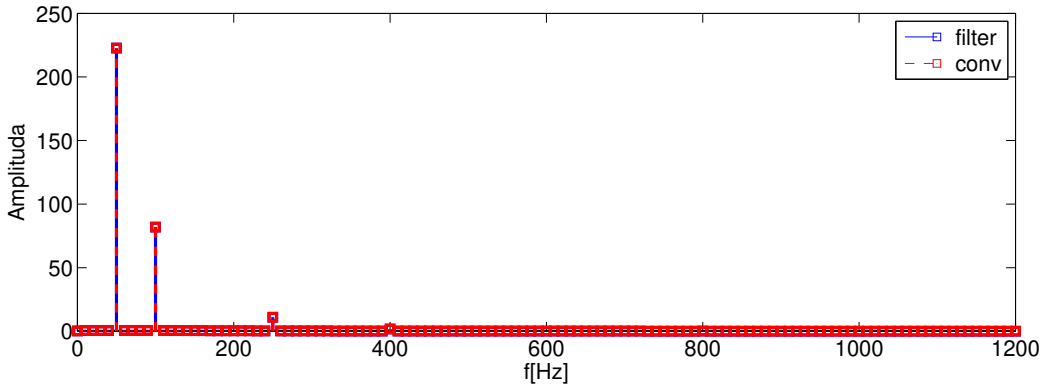
Wykres. 2: d – przebieg dyskretny



Wykres. 3: c – przebieg ciągły, d – przebieg dyskretny



Wykres. 4: c – przebieg ciągły, d – przebieg dyskretny



Wykres. 5: c – przebieg ciągły, d – przebieg dyskretny