

Przeciek widma. Okna czasowe

Cele ćwiczenia

Celem ćwiczenia jest zapoznanie się z przeciek widma oraz oknami czasowymi

Uwagi do ćwiczenia

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

Polecenia języku 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.exp`, `np.zeros`, `np.where`, `np.fft.fft`, `np.abs`, `np.random.rand`, `signal.triang`.

Jeśli jest to możliwe, użyj powyższych poleceń do implementacji rozwiązań poniższych zadań.

Przykład pomocniczy do zadań

Poniższy przykład pozwala na wygenerowanie widma amplitudowego sygnału sinusoidalnego wraz z nałożonym oknem czasowym (wykres. 1 oraz 2). Kod można wykorzystać do rozwiązywania zadań w ramach tej listy.

```
Fs = 1600                                % Częstotliwości próbkowania
Ts = 1/Fs                                % Czas próbkowania
L = 32                                   % Liczba próbek
t_d = (0:L-1)*Ts                         % Oś czasu - wartości dyskretne
t_c = linspace(t_d(1),t_d(end),100)      % Oś czasu - wartości ciągłe
A = 10                                   % Parametry sygnały - Amplituda
f = 50                                   % Parametry sygnały - Częstotliwości
Phi = 0                                  % Parametry sygnały - Przesunięcie fazowe
y_c = A*sin(2*pi*f*t_c + Phi)             % Sygnał - wartości osi Y
y_d = A*sin(2*pi*f*t_d + Phi)             % Sygnał - wartości osi Y

# Wykres
plt.figure(figsize=(15,3))
plt.plot(t_c, y_c, 'r', linestyle = '-')
plt.stem(t_d,y_d , 'b' , markerfmt = 'bs', use_line_collection = True)
plt.grid()
plt.xlabel('Czas[s]')
plt.ylabel('Amplituda')
plt.legend(['- c -', '- d -'])

# DFT 1
Y1 = np.fft.fft(y_d1)
Y1 = np.abs(Y1)/L
Y1[1:-2] = 2*Y1[1:-2]
f1 = Fs*(np.linspace(0,L,L))/L
```

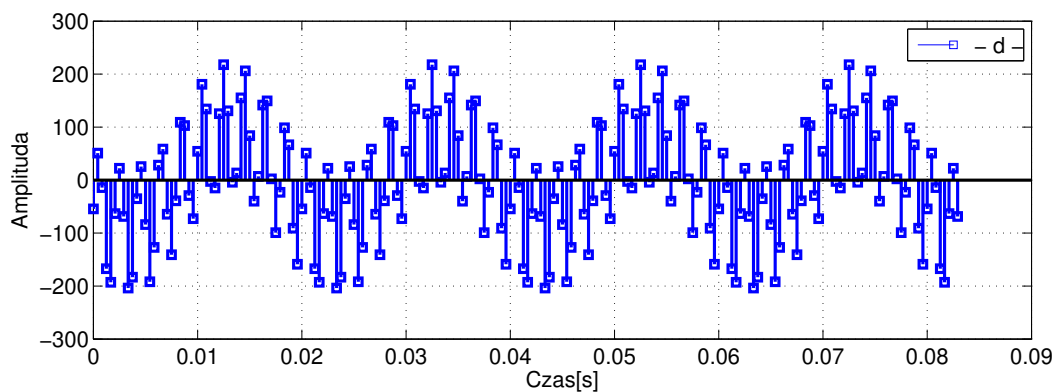
```

# Okno czasowe
w = signal.triang(L)
y_d2 = y_d1*w*2

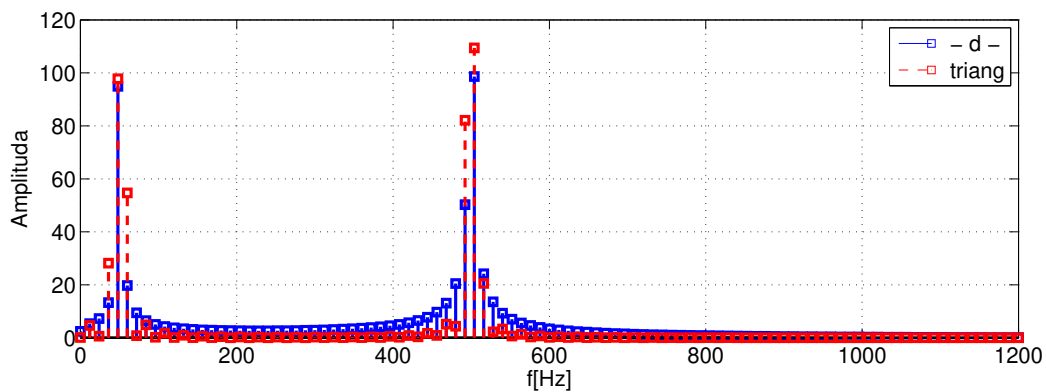
# DFT 2
Y2 = np.fft.fft(y_d2)
Y2 = np.abs(Y2)/L
Y2[1:-2] = 2*Y2[1:-2]
f2 = Fs*(np.linspace(0,L,L))/L

# Wykres
plt.figure(figsize=(15,3))
plt.stem(f1[:int(L/2+1)],Y1[:int(L/2+1)] , 'b' ,
        markerfmt = 'bs', use_line_collection = True)
plt.stem(f2[:int(L/2+1)],Y2[:int(L/2+1)] , '--r' ,
        markerfmt = 'rs', use_line_collection = True)
plt.grid()
plt.xlabel('f[Hz]')
plt.ylabel('Amplituda')
plt.legend(['- d -'])

```



Wykres. 1: c – przebieg ciągły, d – przebieg dyskretny



Wykres. 2: d – przebieg dyskretny

Zadania do wykonania

1. Wygenerować sumę przebiegów sinusoidalnych według następującego wzoru

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

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

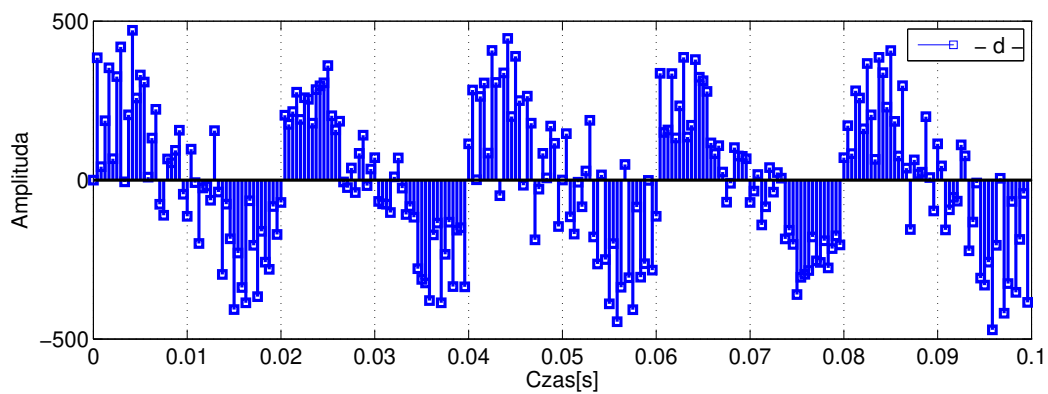
Nr. przebiegu	$A_i[V]$	$f_i[Hz]$	$\phi_i[^\circ]$
1	230	50	0
2	115	100	0
3	75	250	0
4	35	480	0
5	55	500	0
6	85	780	0
7	85	800	0
8	55	850	50

$F_s = 2400[Hz]$, liczba próbek $L = 240$. Wykonać wykres widma amplitudowego. Wynikiem ma być wykres 3 oraz 4.

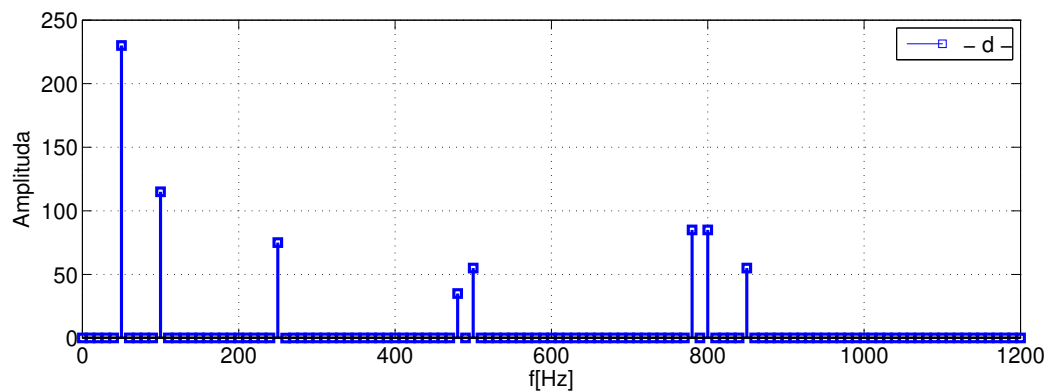
2. Analogicznie do poprzedniego zadania wygenerować sumę przebiegów sinusoidalnych dla liczby próbek $L = 300$. Wykonać wykres widma amplitudowego. Wynikiem ma być wykres 5 oraz 6.
3. Przedstawić na jednym wykresie widmo amplitudowe z zadania 1 oraz 2. Wynikiem ma być wykres 7.
4. Wykonać nakładanie okna czasowego w celu zmniejszenia przecieku widma według poniższej tabeli. Dobrać tak rodzaj okna aby widmo sygnału z zadania 2 jak najlepiej odpowiadało widmu sygnału z zadania 1. Skomentować wyniki oraz uzasadnić wybór okna. Wykres 8 przedstawia przykładowe wyniki dla okna trójkątnego (triang).

scipy.signal	Współczynnik korekcji
triang	2
blackman	2.4
chebwin	2.7 (at = 100)
hamming	1.8
hann	2

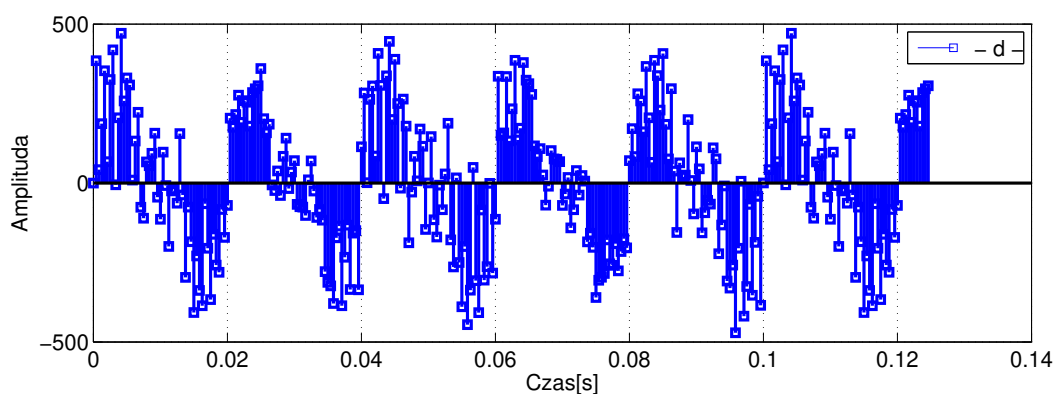
5. Dla wybranego okna z poprzedniego zadania wykonać analizę widma rozszerzając próbki sygnału o wartości zerowe. Aby to wykonać należy podać jako drugi argument funkcji `fft` całkowitą długość sygnału. Przykładowo jeżeli ilość próbek w sygnale wynosi 400 a drugi argument funkcji `fft` 500 to zostanie dodane 100 próbek o wartości zero do oryginalnego sygnału. W praktyce jako drugi argument podaje się wartości będącą potęgą liczby 2. Analizę wykonać dla ilości próbek zerowych odpowiednio: 512, 1024, 2048, 4096, 8192, 16384. Wykres 9 przedstawia przykładowy wynik.
6. Skomentować wyniki uzyskane w poszczególnych zadaniach.



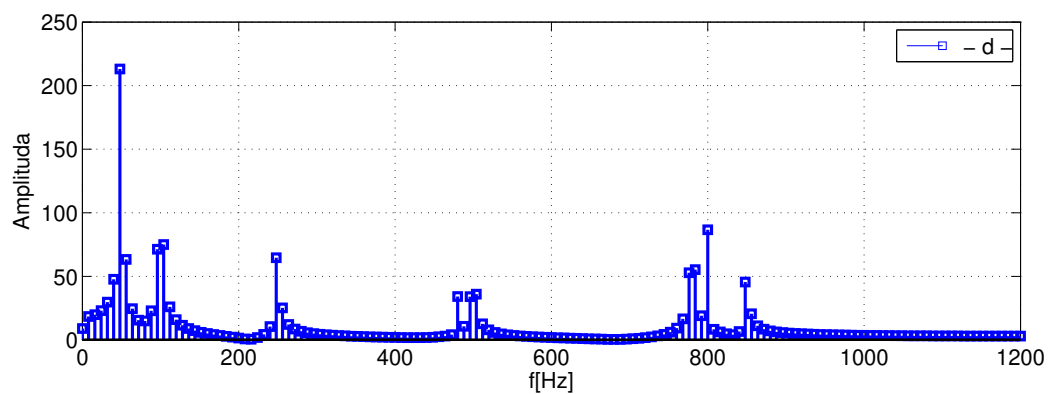
Wykres. 3: d – przebieg dyskretny



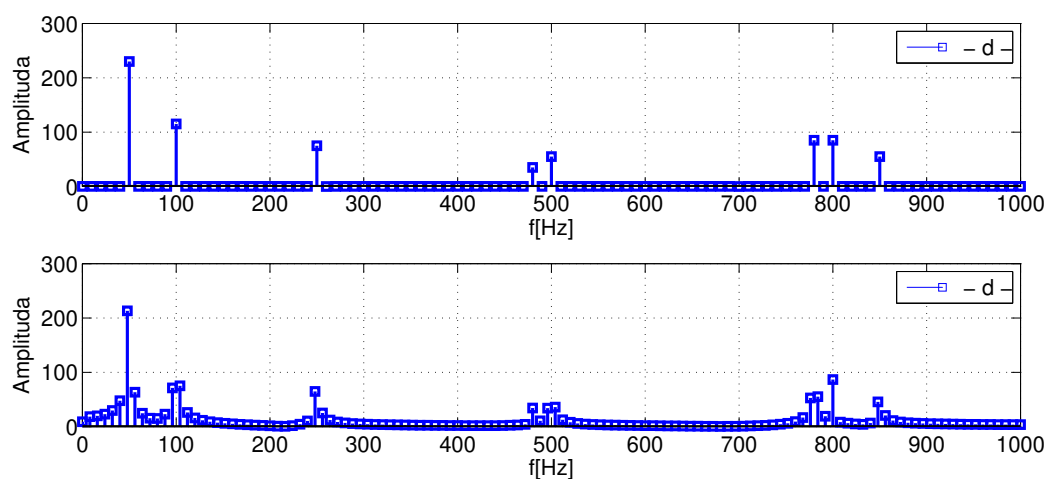
Wykres. 4: d - przebieg dyskretny



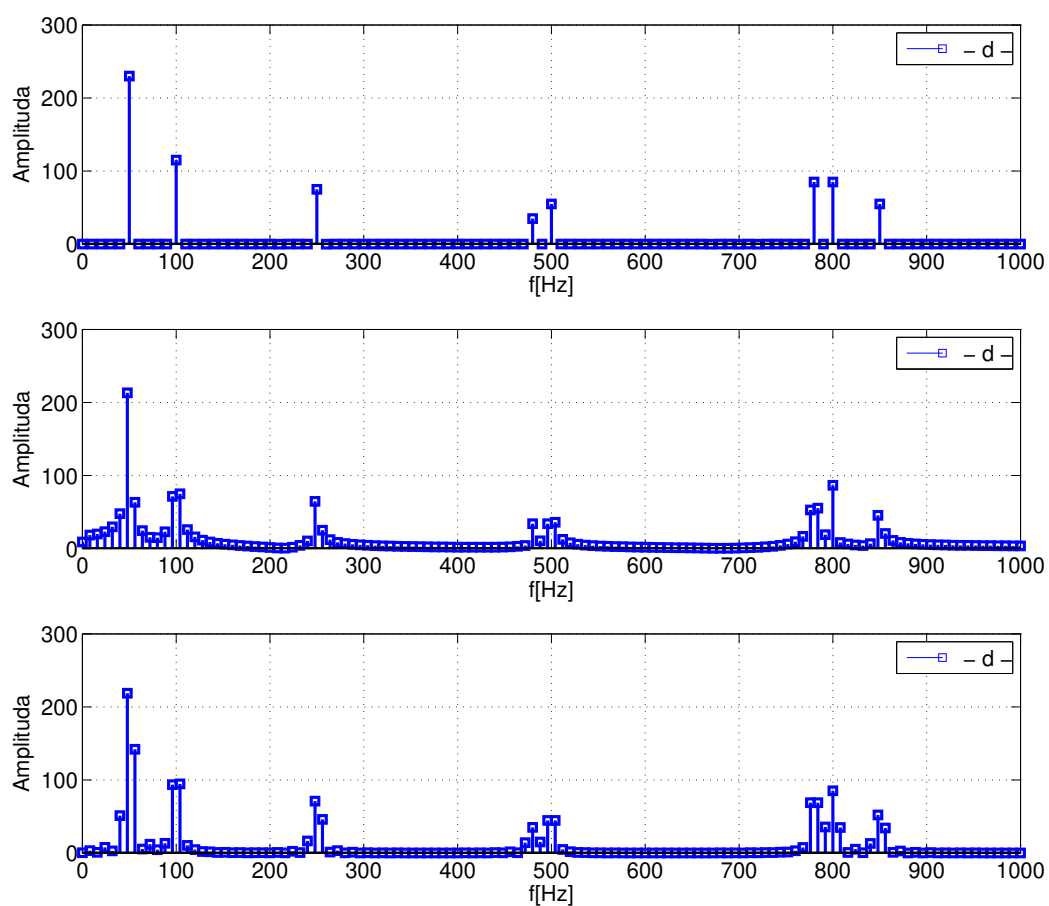
Wykres. 5: d – przebieg dyskretny



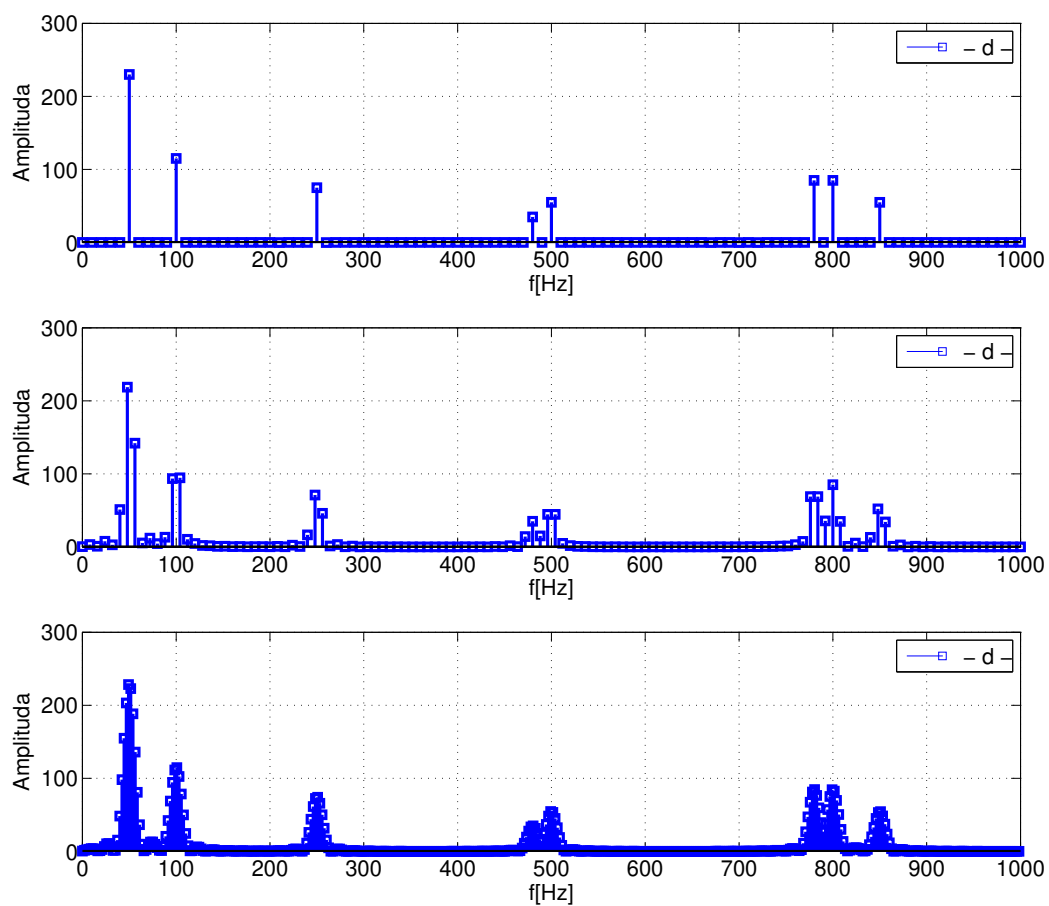
Wykres. 6: d – przebieg dyskretny



Wykres. 7: d – przebieg dyskretny



Wykres. 8: d – przebieg dyskretny



Wykres. 9: d – przebieg dyskretny