

DFT II

Cele ćwiczenia

Celem ćwiczenia jest zapoznanie się z dyskretnym przekształceniem Fouriera oraz rozkładem sumy sygnałów sinusoidalnych na składowe podstawowe.

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

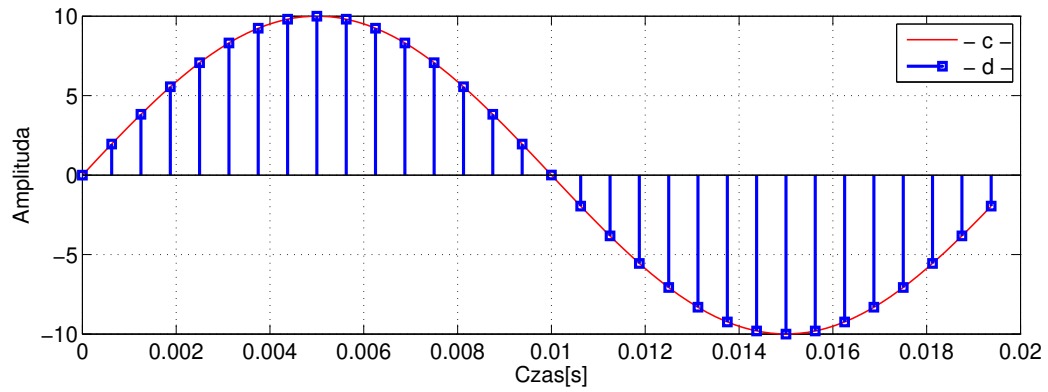
Przykład pomocniczy do zadań

Poniższy przykład pozwala na wygenerowanie widma amplitudowego sygnału sinusoidalnego (wykres. 1 oraz 2). Kod można wykorzystać do rozwiązania zadań w ramach tej listy.

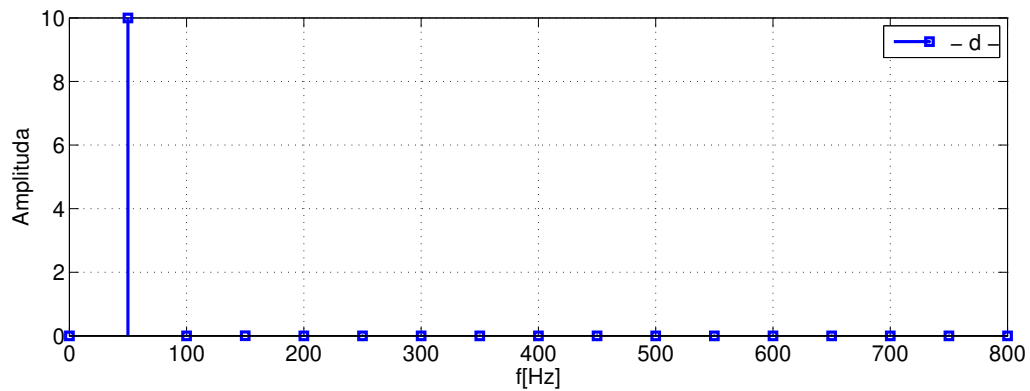
```
Fs = 1600;                % Częstotliwości próbkowania
Ts = 1/Fs;                % Czas próbkowania
L = 32;                   % Liczba próbek
t_d = (0:L-1)*Ts;         % Oś czasu - wartości dyskretne
t_c = linspace(t_d(1),t_d(end),100); % Oś czasu - wartości ciągłe
A = 10;                   % Parametry 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
Y = np.fft.fft(y_d)
Y = np.abs(Y)/L
Y[1:-2] = 2*Y[1:-2]
f = Fs*(np.linspace(0,L,L))/L
```

```
# Wykres
plt.figure(figsize=(15,3))
plt.stem(f[:int(L/2+1)],Y[:int(L/2+1)] , 'b' ,
        markerfmt = 'bs', 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	10
2	115	100	20
3	75	250	30
4	35	400	40
5	25	800	50

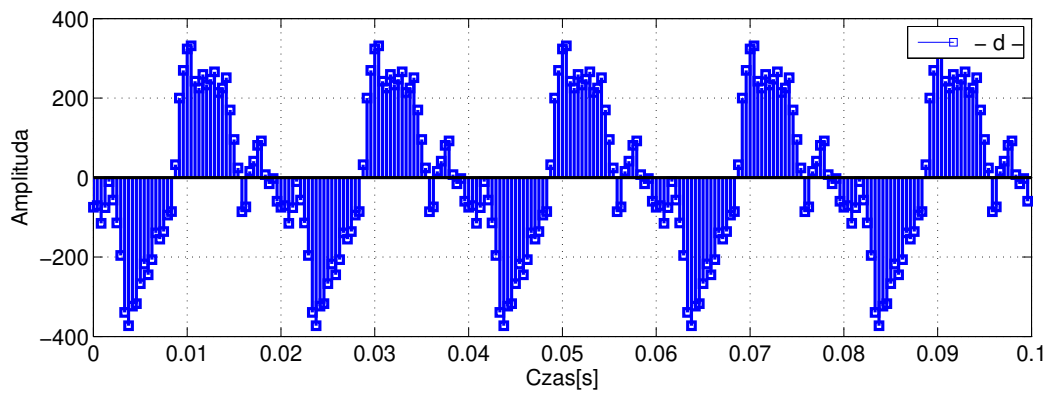
$F_s = 2400[Hz]$, liczba próbek $L = 240$. Wykonać wykres widma amplitudowego. Wynikiem ma być wykres 3 oraz 4.

2. Wygenerować przebieg który jest sumą sygnału wygenerowanego w zadaniu 1 oraz sygnału wygenerowanego na podstawie poniższego kodu

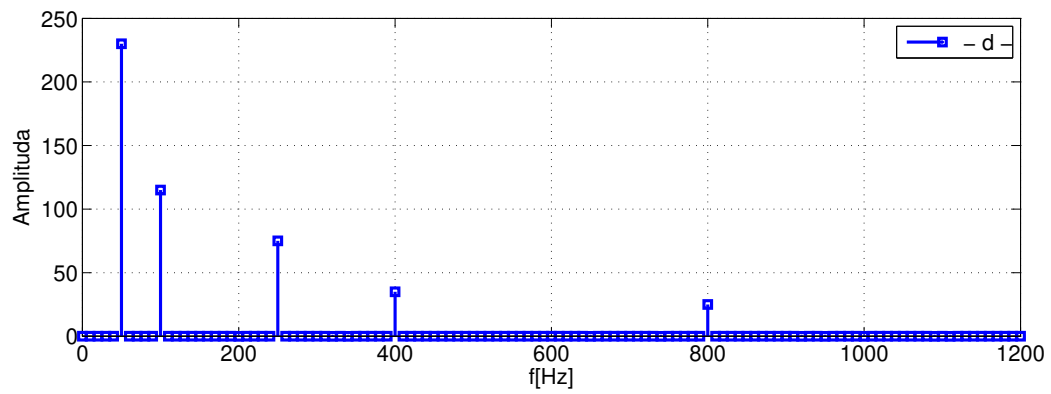
```
Amin = 1
Amax = 300
Fmin = 0
Fmax = 2500
Aa = Amin + (Amax-Amin)*np.random.rand(len(t_d))
ff = np.linspace(Fmin,Fmax,len(t_d))
yy = Aa*np.sin(2*np.pi*ff*t_d)
```

a następnie wykonać analizę analogicznie do zadania pierwszego. Wykonać wykres widma amplitudowego. Wynikiem ma być wykres 5 oraz 6.

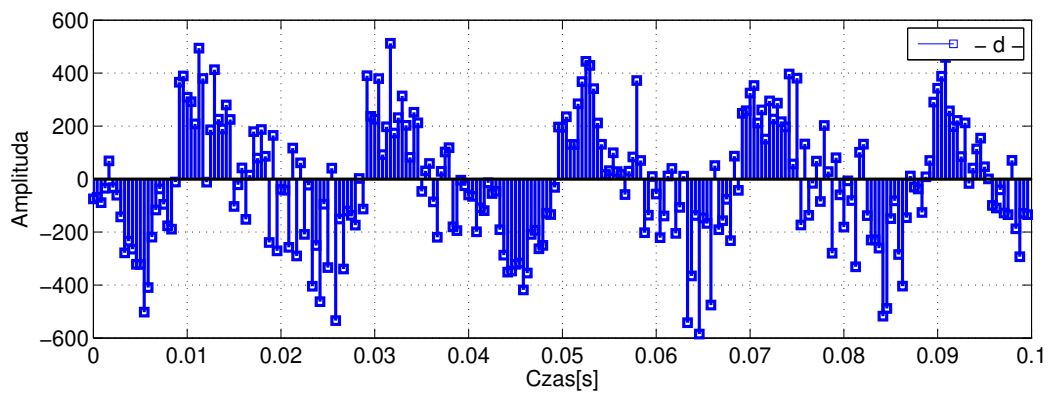
3. Wykonać analizę sygnału z zadania 2 dla ilość próbek L odpowiednio: 240, 480, 720, 960, 1200, 1440, 1680, 1920, 2160, 2400 poprzez wyznaczenie widma amplitudowego. Sprawdzić wpływ ilość próbek na wartości widma amplitudowego. Skomentować wyniki.



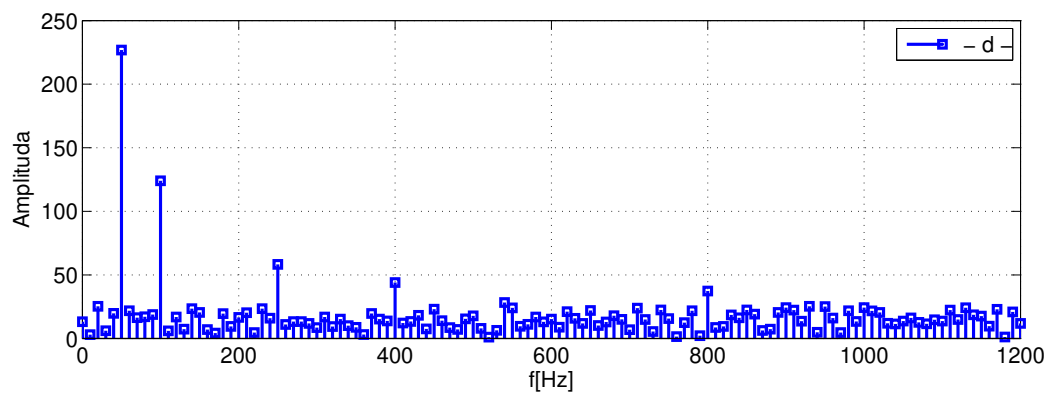
Wykres. 3: d – przebieg dyskretny



Wykres. 4: d - przebieg dyskretny



Wykres. 5: d – przebieg dyskretny



Wykres. 6: d – przebieg dyskretny