SPRAWOZDANIE 5

Zajęcia nr 6 - Całkowe przekształcanie Fouriera - Michał Midor, gr. 5 - 5.11.2025, godz: 16:45

Wstęp teoretyczny:

Celem niniejszego ćwiczenia jest praktyczna analiza ciągłego przekształcenia Fouriera (CFT). W odróżnieniu od szeregów Fouriera, które stosuje się wyłącznie do sygnałów okresowych , całkowe przekształcenie Fouriera pozwala na analizę częstotliwościową sygnałów nieokresowych lub o ograniczonym czasie trwania.

W ramach ćwiczeń, do obliczeń symbolicznych wykorzystany został pakiet MATLAB. Analizie poddano podstawowe sygnały (sinusoida, kosinusoida, impulsy), zjawisko modulacji amplitudowej oraz wpływ okienkowania sygnału na jego widmo częstotliwościowe.

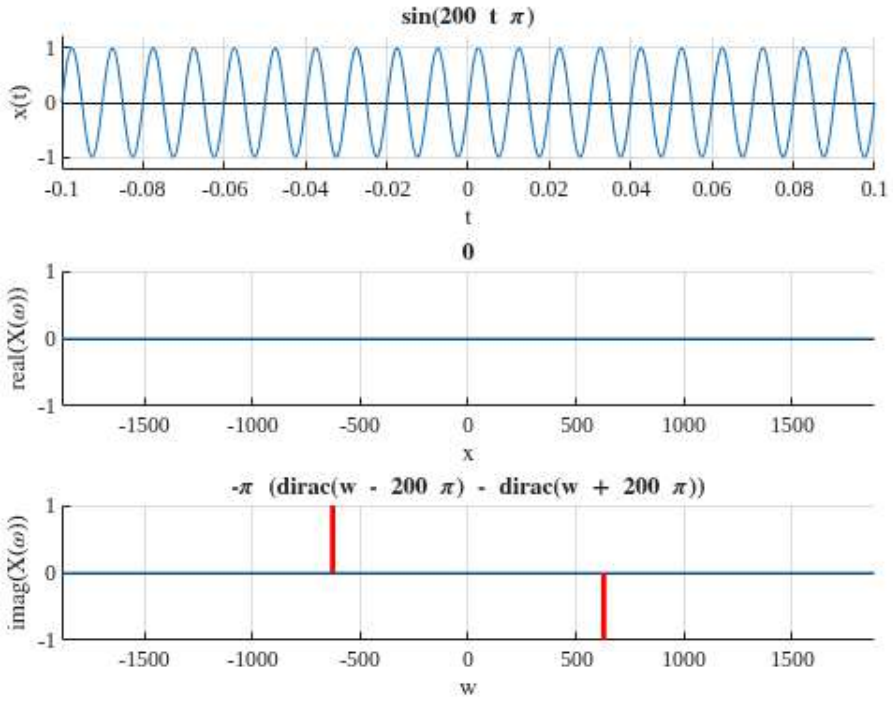
Rozwiązanie zadań:

Zadanie 1)

Pierwszym krokiem było wyznaczenie transformaty Fouriera dla podstawowego sygnału sinusoidalnego. Zgodnie z instrukcją, wykreślono sygnał w dziedzinie czasu oraz część rzeczywistą i urojoną jego widma.

```
clear all; close all;
syms t x f0 w w0 X_FT

f0 = 100; %Hz
w0 = 2*pi*f0;

BND_t = [-10/f0;10/f0]; %20 okresow
t_SMP = [BND_t(1):1/(10*f0):BND_t(2) ];
BND_w = [-3*w0;3*w0];
w_SMP = [BND_w(1):w0/10:BND_w(2) ];

x = sin(w0*t);
X_FT = fourier(x); % symboliczne obliczanie transformaty

figure
subplot(3,1,1); ylabel('x(t)'); hold on
ezplot(x,BND_t); hold on; grid on;
v_num = subs(x, t, t_SMP);
n = find(abs(v_num) == inf); % plot dirac (inf) - nieskończone "szpilli" są
rysowane jako słupki
stem(t_SMP(n),sign(v_num(n)),'r*', 'LineWidth', 2);

subplot(3,1,2); ylabel('real(X(\omega))'); hold on
ezplot(real(X_FT), BND_w); hold on; grid on;
v_num = subs(real(X_FT), w, w_SMP);
n = find( abs(v_num) == inf); % plot dirac (inf)
stem(w_SMP(n),sign(v_num(n)),'r*', 'LineWidth', 2);

subplot(3,1,3); ylabel('imag(X(\omega))'); hold on
ezplot(imag(X_FT), BND_w); hold on; grid on
v_num = subs(imag(X_FT), w, w_SMP);
```

```
n = find( abs(v_num) == inf ); % plot dirac (inf)
stem(w_SMP(n),pi*sign(v_num(n)),'r*', 'LineWidth', 2);
```




Zgodnie z teorią i wynikami działania skryptu, transformata sygnału sinusoidalnego jest czysto urojona. Składa się z dwóch delt Diraca (widocznych jako "szpilki") przy pulsacjach równych dodatniej i ujemnej pulsacji podstawowej, co potwierdza wykres części urojonej. Wykres części rzeczywistej jest tożsamościowo równy zero.

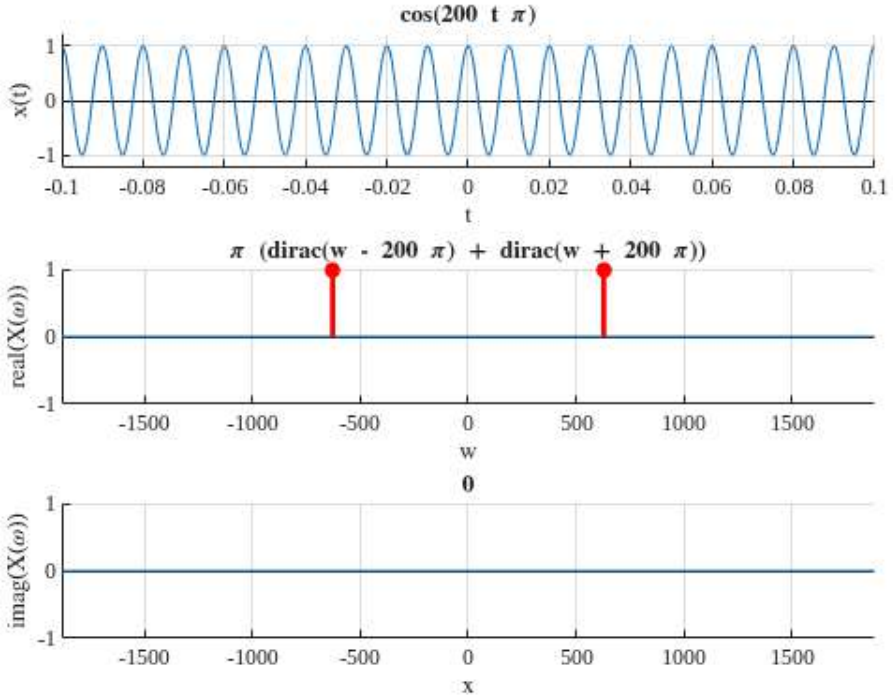
Zadanie 2)

Celem zadania była analiza i porównanie widm częstotliwościowych dla zbioru sygnałów podstawowych: kosinusoidy, sygnału stałego, skoku jednostkowego, impulsu prostokątnego oraz impulsu trójkątnego .

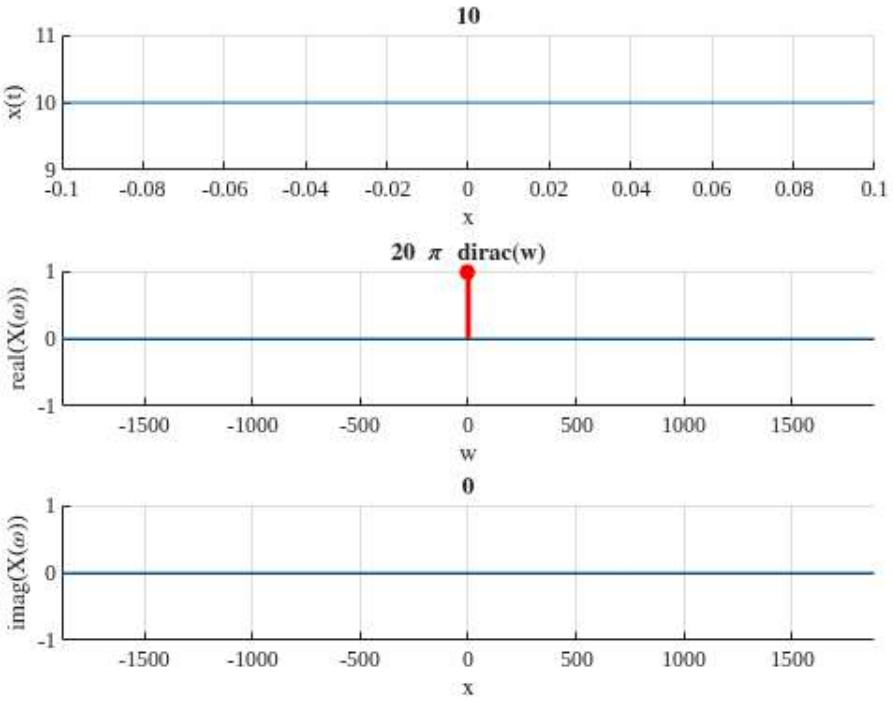
```
x_tri = (1 - f0*abs(t)) * (heaviside(t + 1/f0) - heaviside(t - 1/f0));
sygnaly = {cos(w0*t), sym(10), heaviside(t), heaviside(t + 1/f0) -
heaviside(t - 1/f0), x_tri};

for sygnal=1:length(sygnaly)
    x = sygnaly{sygnal}
    X_FT = fourier(x);
    figure
    subplot(3,1,1); ylabel('x(t)'); hold on
    ezplot(x,BND_t); hold on; grid on;
```

```
    v_num = subs(x, t, t_SMP); % Podstawienie do zamiany wartości
symbolicznych na numeryczne, dla każdej wartości w wektorze t_SMP
    n = find(abs(v_num) == inf); % plot dirac (inf) - nieskończone
"szpilli" są rysowane jako słupki, szukanie indeksów gdzie wartość ucieka
do nieskończoności
    stem(t_SMP(n),sign(v_num(n)),'r*', 'LineWidth', 2); % x to tylko indeksy
znalezione powyżej, y to ich znak, by długość słupka miała amplitudę równą 1

    subplot(3,1,2); ylabel('real(X(\omega))'); hold on
    ezplot(real(X_FT), BND_w); hold on; grid on;
    try
        v_num = subs(real(X_FT), w, w_SMP);
        n = find( abs(v_num) == inf); % plot dirac (inf)
        stem(w_SMP(n),sign(v_num(n)),'r*', 'LineWidth', 2);
    catch
    end

    subplot(3,1,3); ylabel('imag(X(\omega))'); hold on
    ezplot(imag(X_FT), BND_w); hold on; grid on
    try % program się nie zatrzyma, gdy część urojona lub rzeczywista to np.
1/w, a w=0
        v_num = subs(imag(X_FT), w, w_SMP);
        n = find( abs(v_num) == inf ); % plot dirac (inf)
        stem(w_SMP(n),pi*sign(v_num(n)),'r*', 'LineWidth', 2);
    catch
    end
end
```

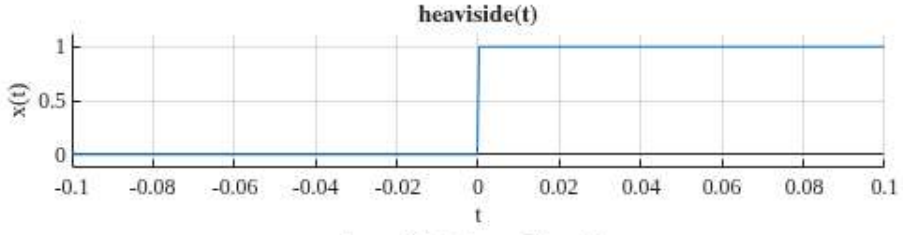
x = $\cos(200\,\pi\,t)$

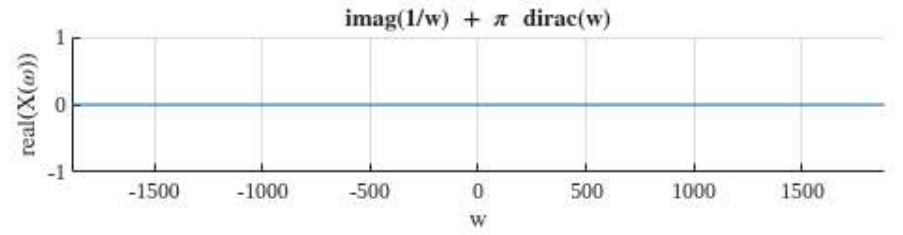


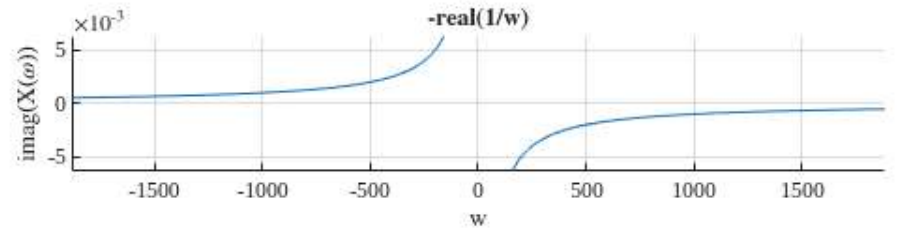

x = 10




x = heaviside($t$)

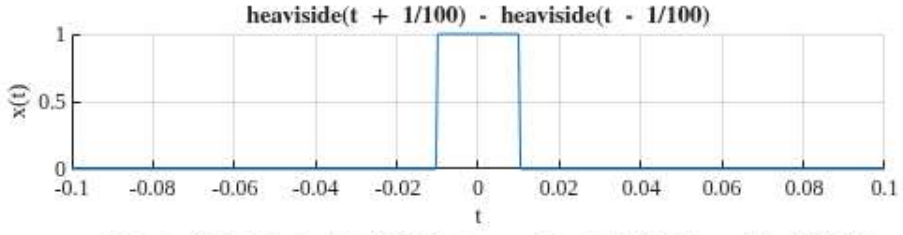


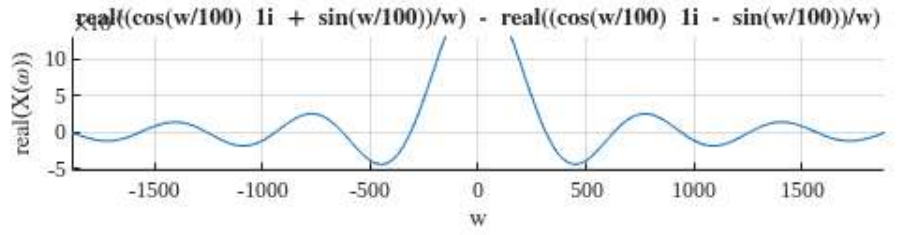



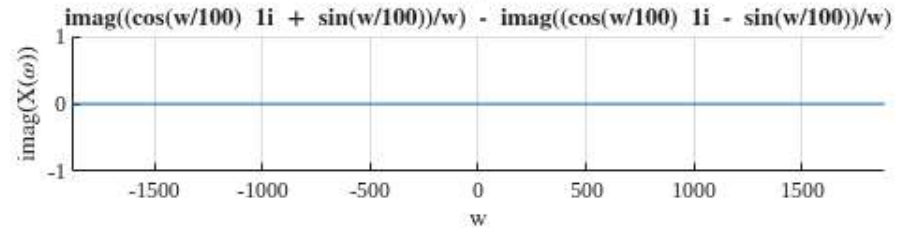




x =

$$\text{heaviside}\left(t+\frac{1}{100}\right)-\text{heaviside}\left(t-\frac{1}{100}\right)$$

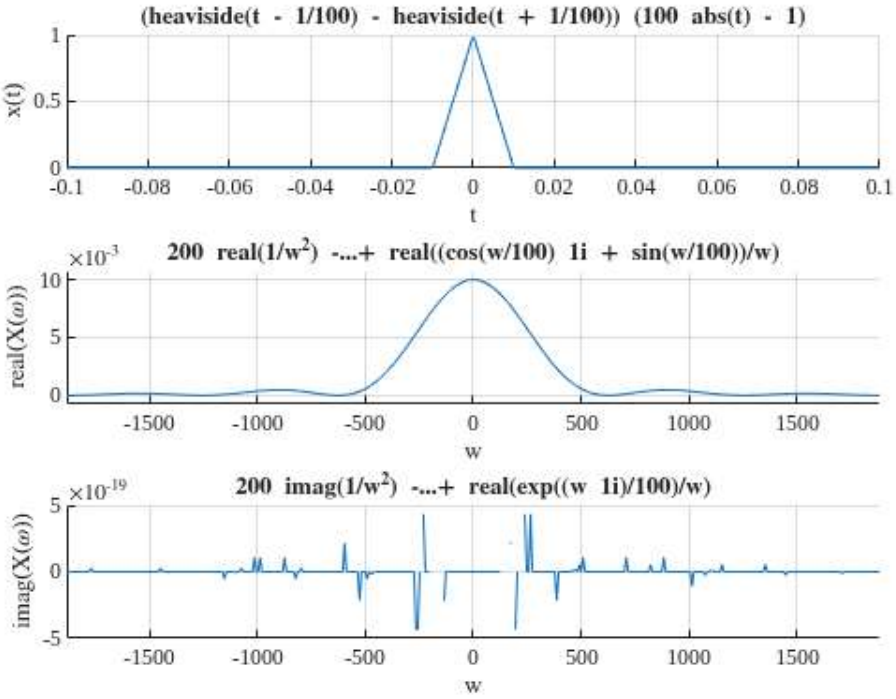



x =

$$\left(\text{heaviside}\left(t-\frac{1}{100}\right)-\text{heaviside}\left(t+\frac{1}{100}\right)\right)\,(100\,|t|-1)$$




Z wykresów wygenerowanych w tym zadaniu można wysnuć wniosek, że "szpilki" występują tam, gdzie sygnał posiada "czyste", niezmienne częstotliwości i cała energia sygnału jest skupiona właśnie w nich.

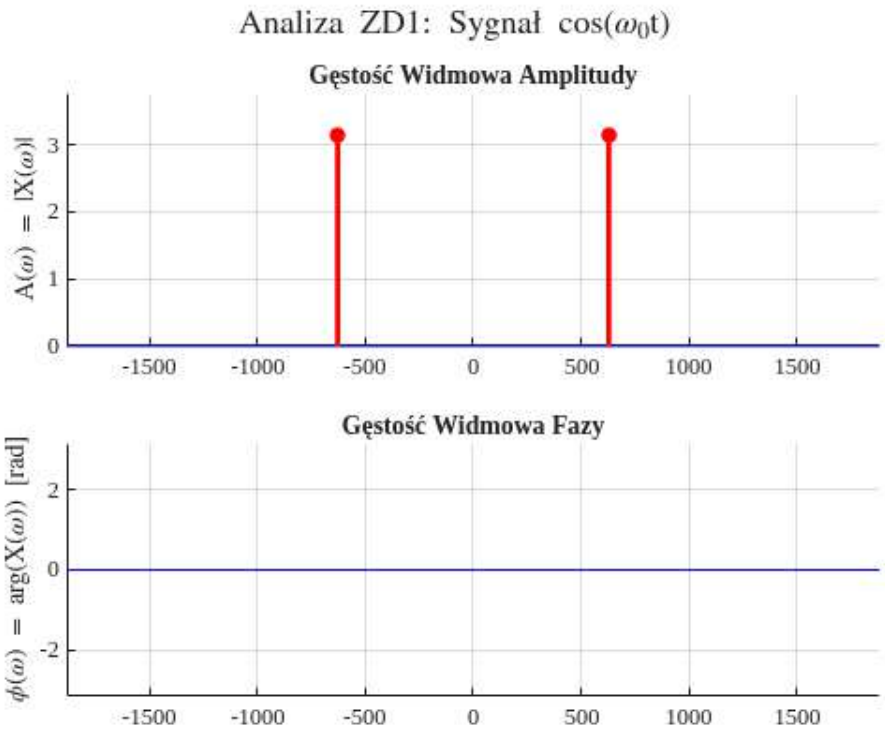
Natomiast dla sygnałów trwających skończony czas widmo jest rozmyte po całej osi częstotliwości, jest ich nieskończenie wiele. Nagłe, ostre zmiany sygnału możliwe są tylko jeśli "zmiesza" się ze sobą ogromną ilość różnych częstotliwości sygnału sinusoidalnego.

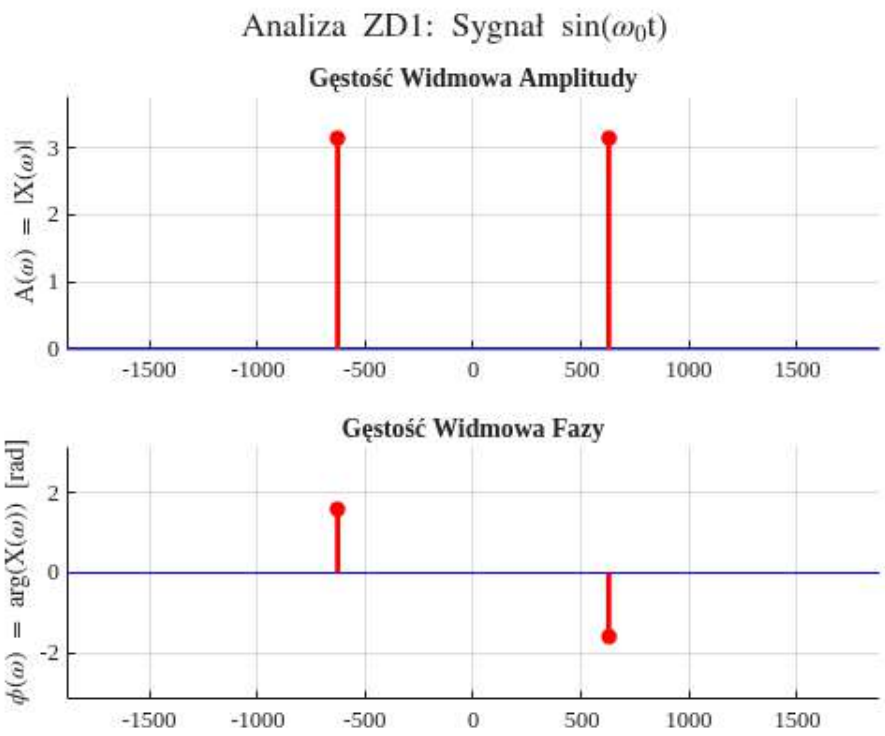
Zadanie domowe 1)

Zadanie polegało na wygenerowaniu i porównaniu wykresów gęstości widmowej amplitudy oraz gęstości widmowej fazy dla sygnałów kosinusa i sinusa.

```
sygnaly_ZD1 = {cos(w0*t), sin(w0*t)};
nazwy_sygnalow = {'cos(\omega_0t)', 'sin(\omega_0t)'};

for i = 1:length(sygnaly_ZD1)
    x = sygnaly_ZD1{i};
    X_FT = fourier(x);

    figure;
    sgtitle(sprintf('Analiza ZD1: Sygnał %s', nazwy_sygnalow{i}));

    subplot(2,1,1);
    ylabel('A(\omega) = |X(\omega)|');
    hold on;
    grid on;
```

```
    title('Gęstość Widmowa Amplitudy');

    plot(BND_w, [0 0], 'b-'); % Linia bazowa y=0
    xlim(BND_w);
    ylim([0 pi*1.2]); % Ustawienie limitu osi Y dla czytelności

    % Ręczne rysowanie delt
    % Amplituda delty = pi
    try
        A_FT = abs(X_FT); % Potrzebne do detekcji
        v_num_A = subs(A_FT, w, w_SMP);
        n_A = find(abs(v_num_A) == inf);
        if ~isempty(n_A)
            stem(w_SMP(n_A), pi * ones(size(n_A)), 'r*', 'LineWidth', 2);
        end
    catch
    end


    subplot(2,1,2);
    ylabel('\phi(\omega) = arg(X(\omega)) [rad]');
    hold on;
    grid on;
    title('Gęstość Widmowa Fazy');

    plot(BND_w, [0 0], 'b-');
    xlim(BND_w);
    ylim([-pi pi]); % Ustawiamy zakres osi Y dla fazy

    % Ręczne rysowanie fazy dla delt
    try
        v_num_imag = subs(imag(X_FT), w, w_SMP);
        n_imag = find(abs(v_num_imag) == inf);

        if ~isempty(n_imag) % Wykona się tylko dla sin(t)
            znaki = sign(v_num_imag(n_imag));
            fazy = znaki * (pi/2); % faza = +pi/2 lub -pi/2
            stem(w_SMP(n_imag), fazy, 'r*', 'LineWidth', 2);
        end
    catch
    end
end
```




Oba sygnały mają dokładnie takie samo widmo amplitudowe - zbudowane są z tych samych częstotliwości, a ich moc (amplituda) jest identyczna.

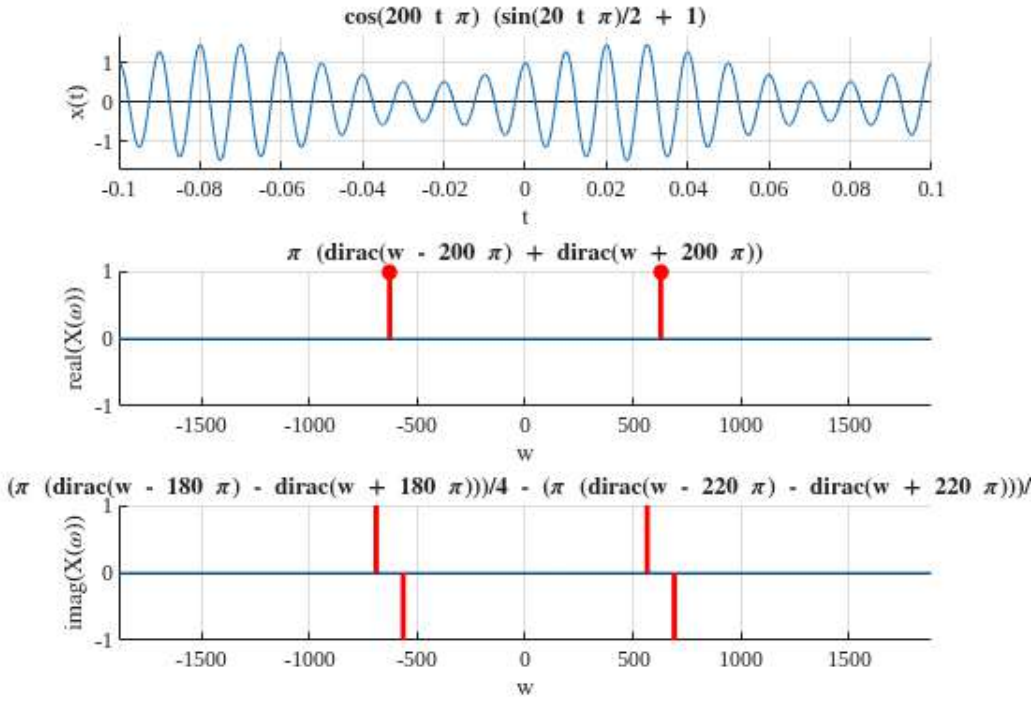
Różnica pojawia się w fazie - cosinus jest funkcją rzeczywistą i parzystą, tak samo jak jego transformata Fouriera, sinus natomiast jest rzeczywisty nieparzysty, więc transformata jest czysto urojona.

Zadanie 3)

W tym zadaniu przeprowadzono symulację modulacji amplitudowej (AM). Sygnał nośny cos(w0*t) pomnożony został przez sygnał modulujący  (1 + m * x_m), gdzie x_m to sinusoida o 10-krotnie mniejszej częstotliwości, a głębokość modulacji m=0.5.

Dodatkowo porównuję moc sygnału nośniego P_nosny z całkowitą mocą sygnału zmodulowanego P_y w celu określenia wpływu parametru m na tę relację.

```
x_n = cos(w0*t);
w_m = w0 / 10;
x_m = sin(w_m * t);
m = 0.5;
x = (1 + m * x_m) * x_n;

X_FT = fourier(x);
figure
subplot(3,1,1); ylabel('x(t)'); hold on
ezplot(x,BND_t); hold on; grid on;
v_num = subs(x, t, t_SMP); % Podstawienie do zamiany wartości symbolicznych
na numeryczne, dla każdej wartości w wektorze t_SMP
n = find(abs(v_num) == inf); % plot dirac (inf) - nieskończone "szpilli"
są rysowane jako słupki, szukanie indeksów gdzie wartość ucieka do
nieskończoności
stem(t_SMP(n),sign(v_num(n)),'r*', 'LineWidth', 2); % x to tylko indeksy
znalezione powyżej, y to ich znak, by długość słupka miała amplitudę równą 1

subplot(3,1,2); ylabel('real(X(\omega))'); hold on
ezplot(real(X_FT), BND_w); hold on; grid on;
try
    v_num = subs(real(X_FT), w, w_SMP);
    n = find( abs(v_num) == inf); % plot dirac (inf)
    stem(w_SMP(n),sign(v_num(n)),'r*', 'LineWidth', 2);
catch
end

subplot(3,1,3); ylabel('imag(X(\omega))'); hold on
ezplot(imag(X_FT), BND_w); hold on; grid on
try % program się nie zatrzyma, gdy część urojona lub rzeczywista to np.
1/w, a w=0
    v_num = subs(imag(X_FT), w, w_SMP);
    n = find( abs(v_num) == inf ); % plot dirac (inf)
    stem(w_SMP(n),pi*sign(v_num(n)),'r*', 'LineWidth', 2);
catch
end
```




```
% Zadanie domowe 2:

A_nosny = 1;
P_nosny = A_nosny*A_nosny/2;
% P_y = P_n + (m^2 / 4)
P_y = P_nosny + (m*m/4);
fprintf('Relacja mocy (Py / Pn): %.4f\n', P_y / P_nosny);
```

```
Relacja mocy (Py / Pn): 1.1250
```

Wykresy widma sygnału zmodulowanego wyraźnie pokazały charakterystyczną strukturę AM. 3 wykres sygnału modulowanego ukazuje dwa symetryczne prążki, w których zawarta jest informacja o sygnale modulującym.

Głębokość modulacji ma kluczowy wpływ, ponieważ całkowita moc sygnału smodulowanego rośnie proporcjonalnie do m^2 - oznacza to, że im większa wartość m, tym więcej energii (mocy) umiejscowione zostaje w prążkach bocznych.

Zadanie 4)

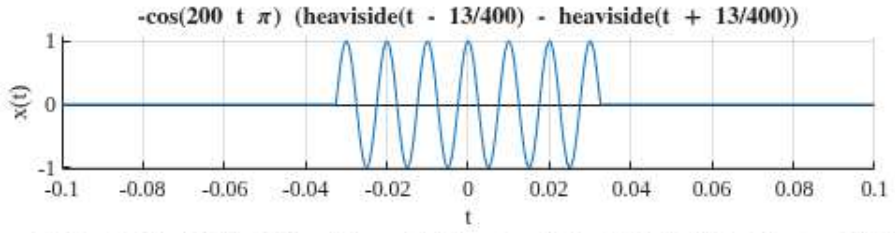
Celem zadania była analiza wpływu okienkowania sygnału na jego widmo częstotliwościowe. Sygnał kosinusoidalny o skończonym czasie trwania (6.5 okresu) został pomnożony przez trzy różne funkcje okna: prostokątne, trójkątne oraz Gaussa.

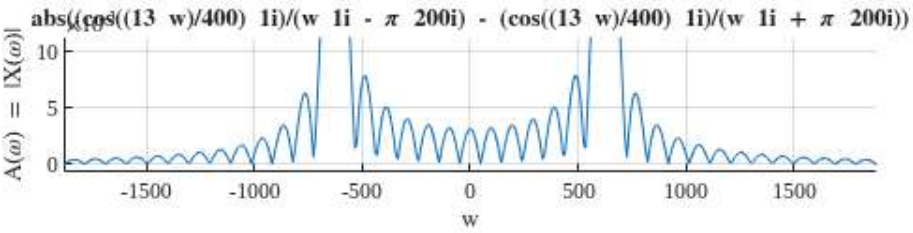
```
To = 6.5/f0;
w_rect = heaviside(t + To/2) - heaviside(t - To/2);
w_tri = (1 - abs(t) / (To/2)) * w_rect;
```

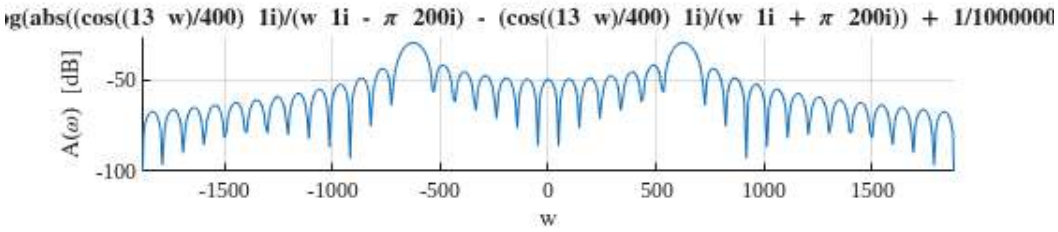
```
c = (To/2) / 3;
w_gauss = exp(-t^2 / (2*c^2));

sygnaly_okno = {cos(w0*t) * w_rect, cos(w0*t) * w_tri, cos(w0*t) * w_gauss};

for sygnal_okno=1:length(sygnaly_okno)
    x = sygnaly_okno{sygnal_okno}
    X_FT = fourier(x);
    figure

    subplot(3,1,1);
    ylabel('x(t)');
    hold on;
    ezplot(x, BND_t);
    hold on;
    grid on;
    try
        v_num = subs(x, t, t_SMP);
        n = find(abs(v_num) == inf); % gęstość widmowa amplitudy
        stem(t_SMP(n), sign(v_num(n)), 'r*', 'LineWidth', 2);
    catch
    end

    subplot(3,1,2);
    ylabel('A(\omega) = |X(\omega)|');
    hold on;
    ezplot(abs(X_FT), BND_w);
    hold on;
    grid on;

    subplot(3,1,3);
    ylabel('A(\omega) [dB]');
    hold on;
    ezplot(20*log10(abs(X_FT) + 1e-6), BND_w); % zwiększenie czytelności
dzięki skali dB
    hold on;
    grid on;
end
```

x =

$$-\cos(200\,\pi\,t)\left(\text{heaviside}\left(t-\frac{13}{400}\right)-\text{heaviside}\left(t+\frac{13}{400}\right)\right)$$

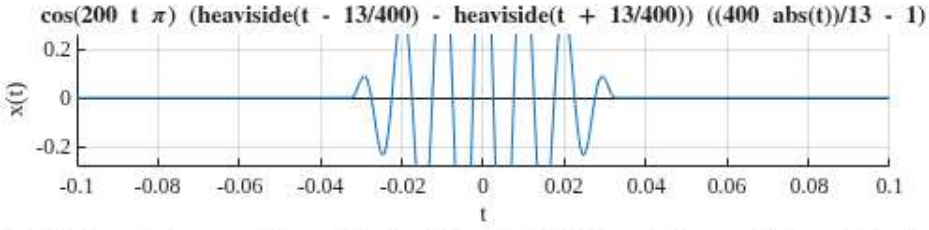


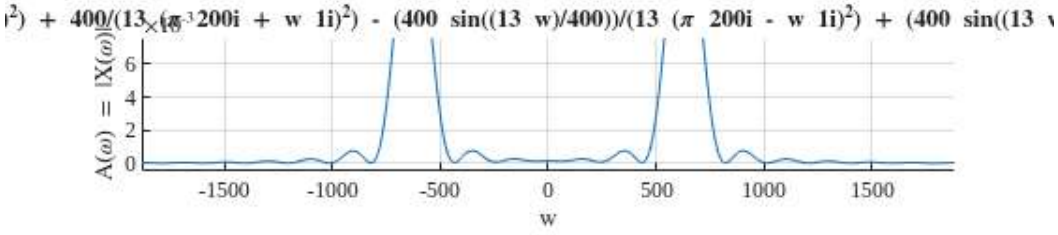



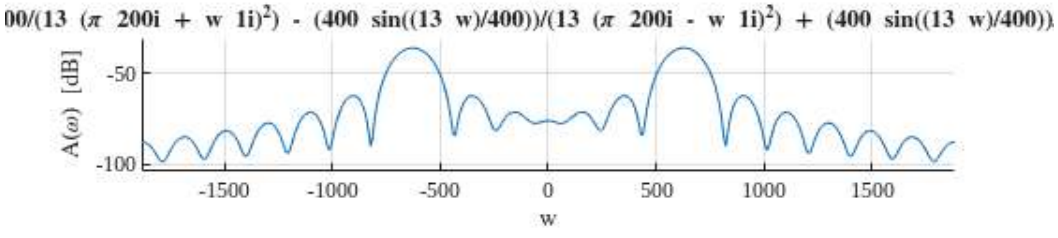




x =

$$\cos(200\,\pi\,t)\,\left(\text{heaviside}\left(t-\frac{13}{400}\right)-\text{heaviside}\left(t+\frac{13}{400}\right)\right)\,\left(\frac{400\,|t|}{13}-1\right)$$

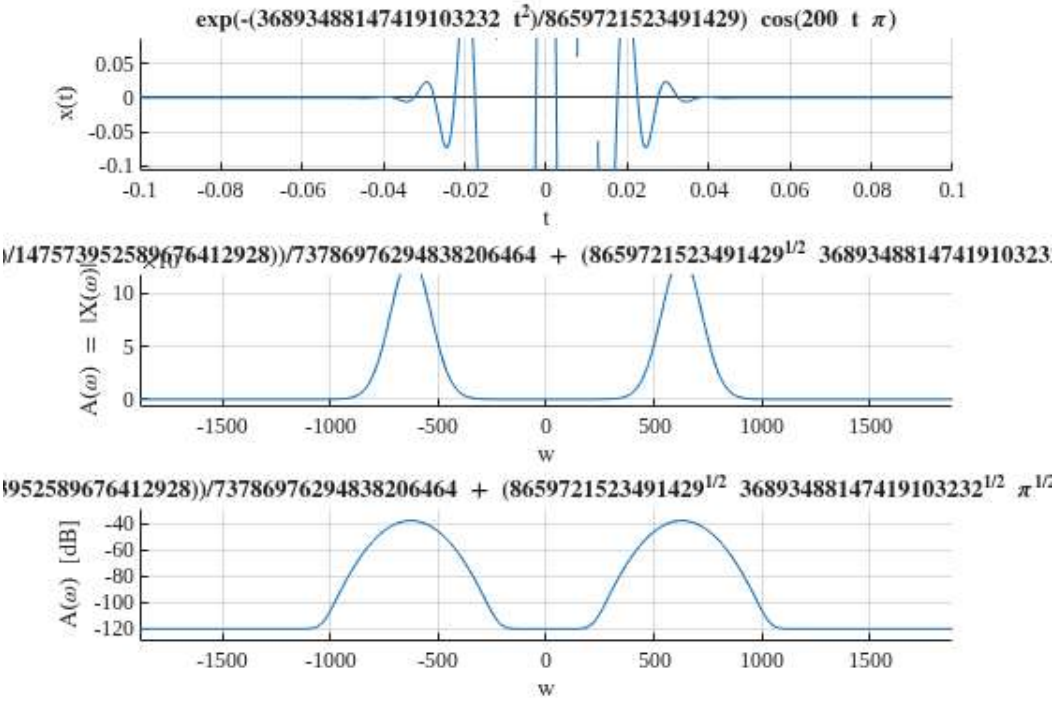



x =

$$e^{-\frac{3689348814741910323 2\,t^2}{8659721523491429}}\cos(200\,\pi\,t)$$




Widzimy, że okienktowanie prostokątne ma najlepszą rozdzielczość - prążek główny jest bardzo wąski. Niestety prąðki boczne są bardzo wysokie, co powoduje największy przeciek widmowy.

Okno trójkątne ma nieco szerszy słupek główny natomiast boczne opadają znacznie szybciej i są dużo mniejsze.

Dla okna Gaussa słupek główny jest najszerszy, ale nie ma żadnego wycieku widmowego.

Najlepszą metodą jest kompromis, czyli okno trójkątne - oferuje bardzo duża redukcję przecieku widmowego, kosztem akceptowalnej utraty rozdzielczości.