

Przetwarzanie Sygnałów Cyfrowych

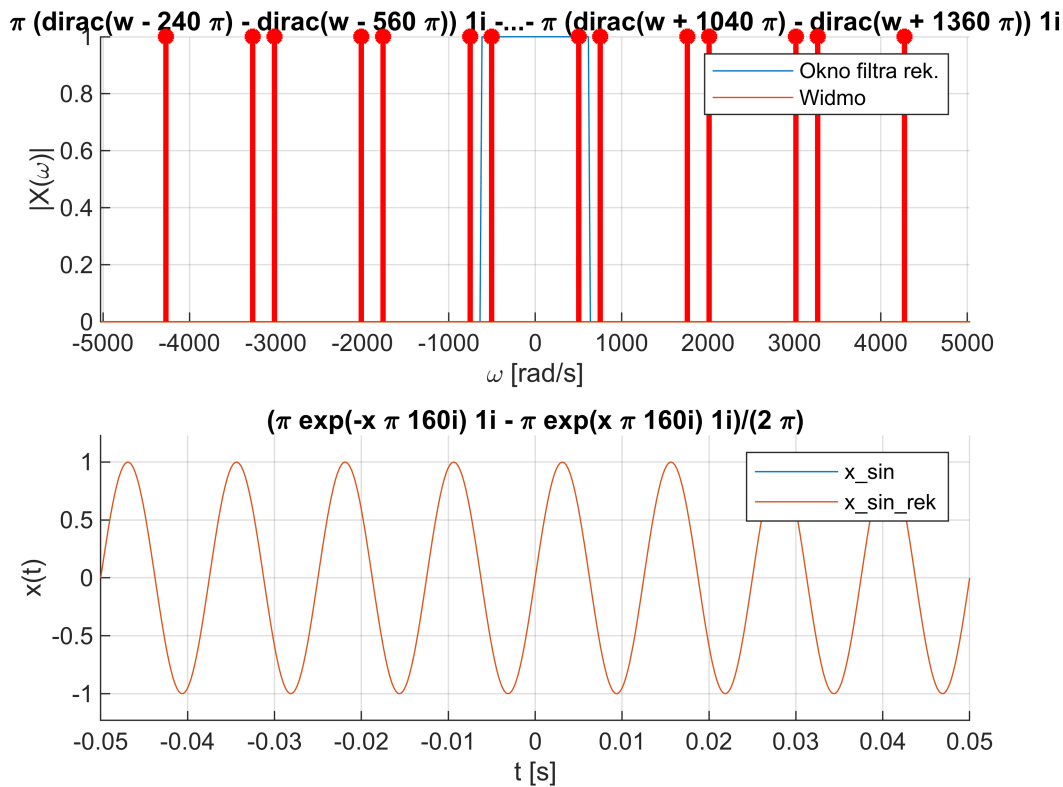
Próbkowanie sygnałów ciągłych

Jan Rosa 410269 AiR

Ćwiczenie 1

Przeprowadź rekonstrukcję sygnału ciągłego $\sin(\omega_k t)$ próbkowanego z częstotliwością 200Hz - szkic skryptu przedstawiono na rysunku 1.

```
clear all; close all;
syms t x w K
fp = 200; fg = fp/2; %Hz
wp = 2*pi*fp; wg = 2*pi*fg;
s = 4/5; ws = s*wg;
x_sin = sin(ws*t);
X_FT_sin_org = fourier(x_sin);
X_FT_sin = X_FT_sin_org + ... % oryginal widma
    symsum((subs(X_FT_sin_org, w, w - K*wp ) + ... % 3 aliasy lewe
    subs(X_FT_sin_org, w, w + K*wp)), K , 1, 3); % 3 aliasy prawe
FILT_FT = rectangularPulse(-wg,wg,w); % filtr rekonstruujący
x_sin_rek = ifourier(X_FT_sin*FILT_FT); % odwr. tarnsf. Fouriera
BND_t = [-10/fp;10/fp];
t_SMP = [BND_t(1):1/(10*fp):BND_t(2) ];
BND_w = [-4*wp;4*wp];
w_SMP = [BND_w(1):wp/10:BND_w(2)];
figure; subplot(2,1,1); hold on; grid on;
ezplot(FILT_FT,BND_w); %okno filtru rek.
ezplot(X_FT_sin,BND_w)
v_num = abs(double(subs(X_FT_sin, w, w_SMP)));
n = find(abs(v_num) == Inf);
stem(w_SMP(n),sign(v_num(n)), 'r*', 'LineWidth', 2);
xlabel('\omega [rad/s]'); ylabel('|X(\omega)|')
legend('Okno filtra rek.','Widmo');
subplot(2,1,2); hold on; grid on;
ezplot(x_sin, BND_t);
% syg. próbkowany
ezplot(x_sin_rek, BND_t) % syg. odtworzony
xlabel('t [s]'); ylabel('x(t)')
legend('x\_sin','x\_sin\_rek');
```



Zadanie 2

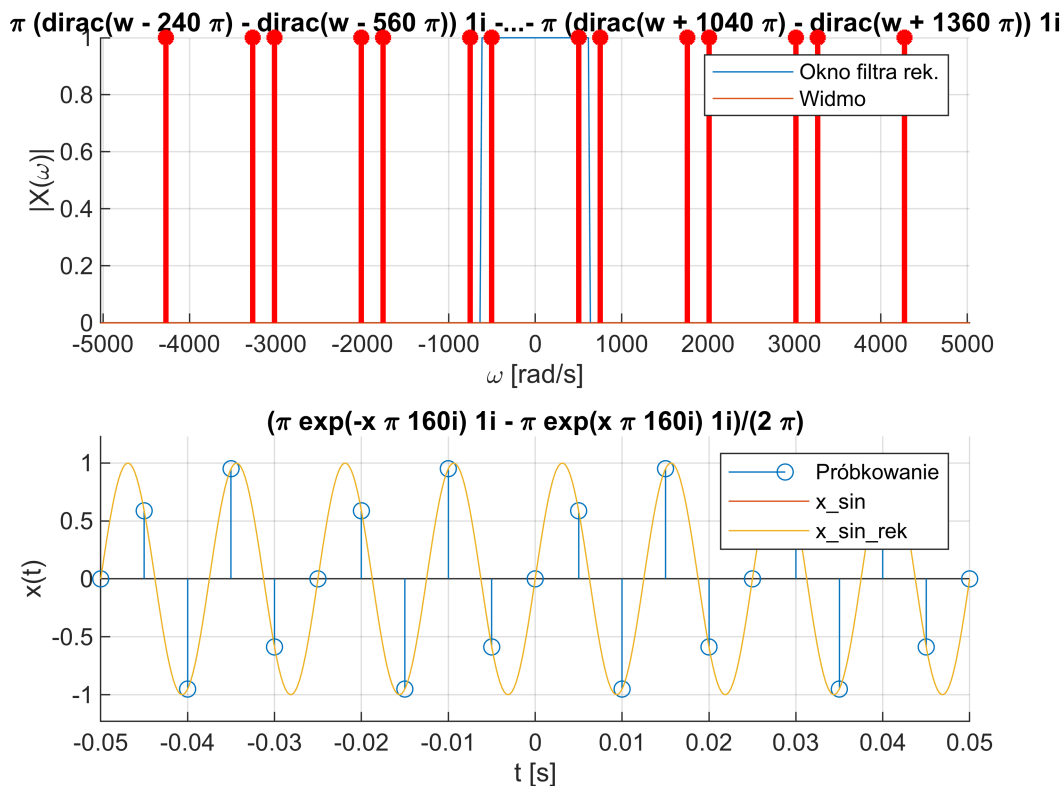
Oznacz na wykresie czasowym węzły próbkowania wyznaczone przez okres próbkującej funkcji grzebieniowej $\delta_{T_p}(t)$, gdzie $T_p = \frac{1}{f_p}$.

```
clear all; close all;
syms t x w K
fp = 200; fg = fp/2; %Hz
wp = 2*pi*fp; wg = 2*pi*fg;
s = 4/5; ws = s*wg;
x_sin = sin(ws*t);
X_FT_sin_org = fourier(x_sin);
X_FT_sin = X_FT_sin_org + ... % oryginalne widma
    symsum((subs(X_FT_sin_org, w, w - K*wp) + ... % 3 aliasy lewe
    subs(X_FT_sin_org, w, w + K*wp)), K, 1, 3); % 3 aliasy prawe
FILT_FT = rectangularPulse(-wg, wg, w); % filtr rekonstruujący
x_sin_rek = ifourier(X_FT_sin * FILT_FT); % odwr. transf. Fouriera
BND_t = [-10/fp; 10/fp];
t_SMP = [BND_t(1):1/(10*fp):BND_t(2)];
BND_w = [-4*wp; 4*wp];
w_SMP = [BND_w(1):wp/10:BND_w(2)];
figure; subplot(2,1,1); hold on; grid on;
ezplot(FILT_FT, BND_w); % okno filtru rek.
ezplot(X_FT_sin, BND_w)
v_num = abs(double(subs(X_FT_sin, w, w_SMP)));
```

```

n = find(abs(v_num) == Inf);
stem(w_SMP(n), sign(v_num(n)), 'r*', 'LineWidth', 2);
xlabel('\omega [rad/s]'); ylabel('|X(\omega)|')
legend('Okno filtra rek.', 'Widmo');
subplot(2,1,2); hold on; grid on;
nodex = BND_t(1):1/fp:BND_t(2);
nodey = subs(x_sin, t, nodex);
stem(nodex, nodey);
ezplot(x_sin, BND_t);
% syg. próbkowany
ezplot(x_sin_rek, BND_t) % syg. odtworzony
xlabel('t [s]'); ylabel('x(t)')
legend('Próbkowanie', 'x_sin', 'x_sin_rek');

```



Zadanie 3

Wykonaj rekonstrukcję sygnału sinusoidalnego o następujących częstotliwościach:

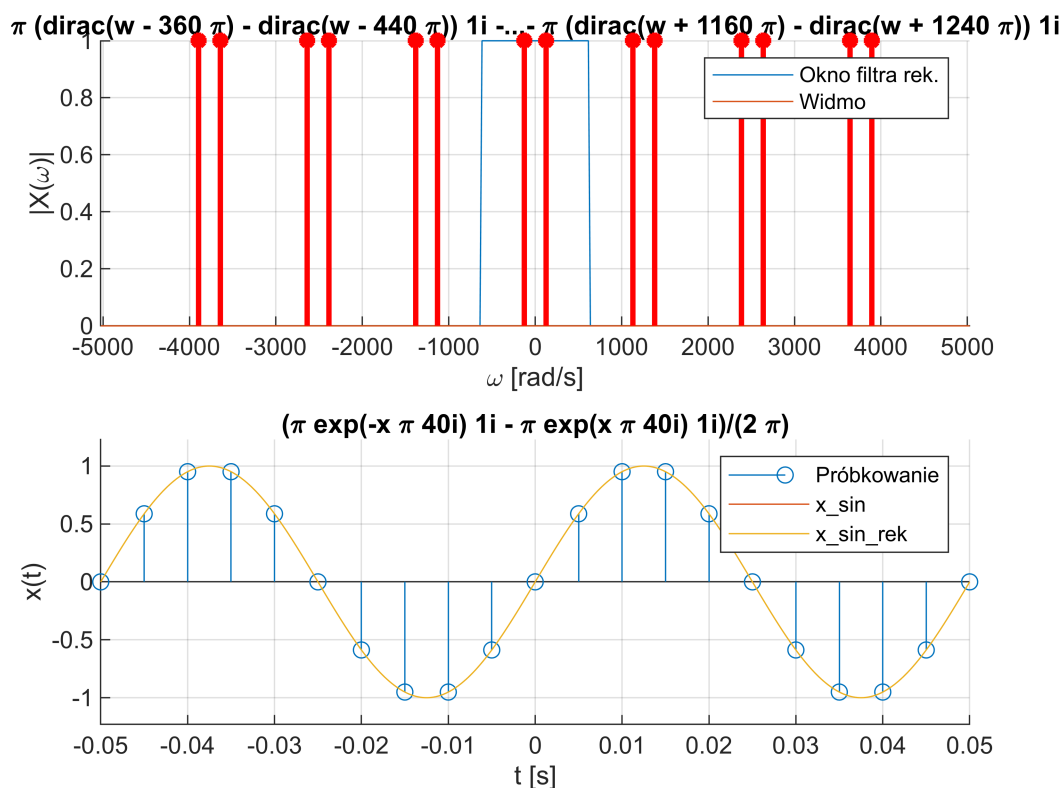
- a) $\frac{1}{5}f_g$, b) $\frac{6}{5}f_g$, c) $\frac{11}{5}f_g$, d) $\frac{16}{5}f_g$,
- e) $\frac{4}{5}f_g$, f) $\frac{9}{5}f_g$, g) $\frac{14}{5}f_g$.

Uwaga: aby ustawić częstotliwość sinusoidy należy zmodyfikować wartość współczynnika s który wyznacza pulsację ws sygnału $x_{\sin} = \sin(ws \cdot t)$.

Zaobserwuj i zanotuj w sprawozdaniu podobieństwa oraz różnice poszczególnych przebiegów czasowych. Wyznacz częstotliwość i fazę sygnału zrekonstruowanego dla każdego przypadku. Wyniki i spostrzeżenia umieść w sprawozdaniu. Zastanów się, czy w wyniku rekonstrukcji można uzyskać funkcję stałą. Jeśli tak, napisz w sprawozdaniu jakie warunki muszą być spełnione i podaj przepis na parametry rekonstrukcji.

a)

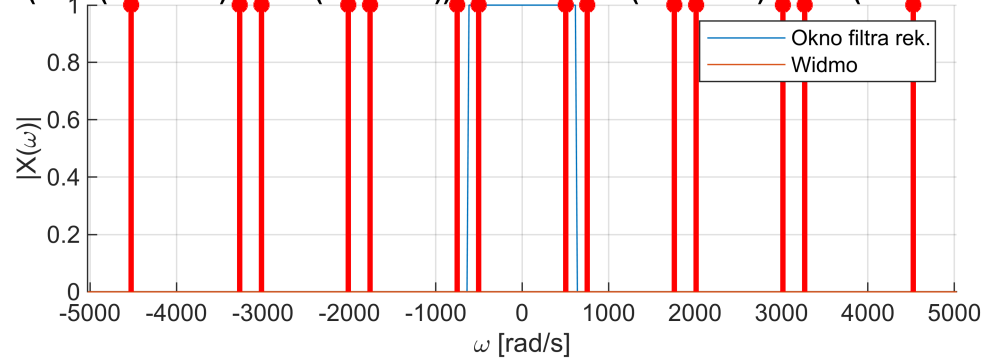
fgplot(1/5)



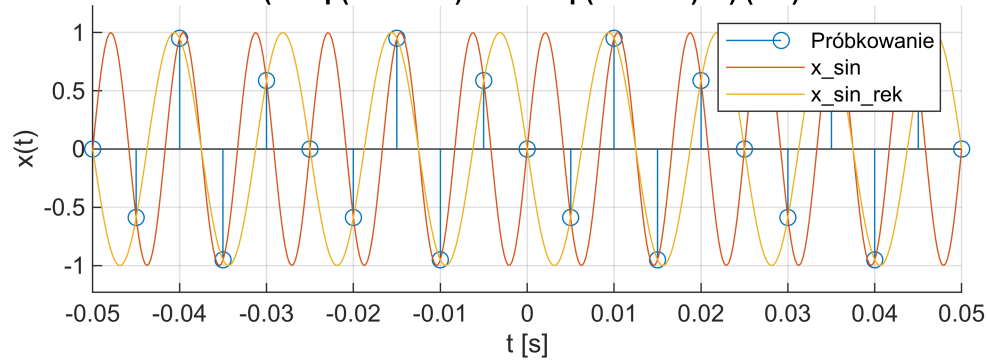
b)

fgplot(6/5)

$$\pi (\text{dirac}(w - 160 \pi) - \text{dirac}(w - 640 \pi)) 1i - \dots - \pi (\text{dirac}(w + 960 \pi) - \text{dirac}(w + 1440 \pi)) 1i$$

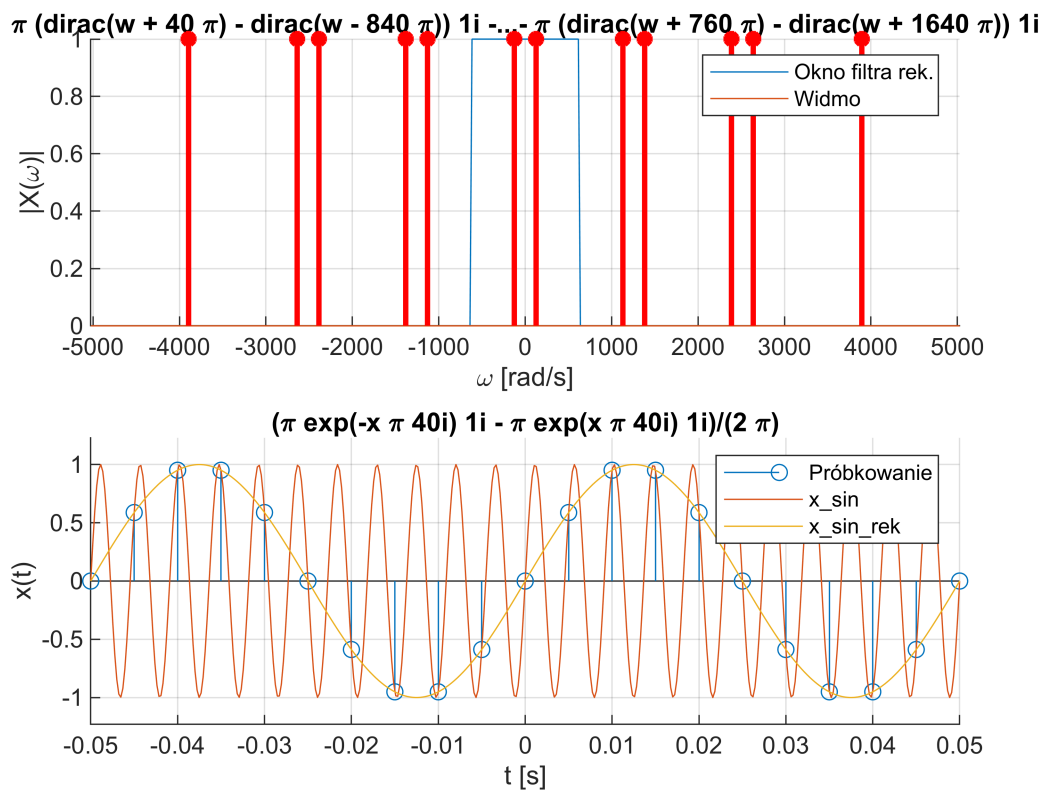


$$-(\pi \exp(-x \pi 160i) 1i - \pi \exp(x \pi 160i) 1i)/(2 \pi)$$



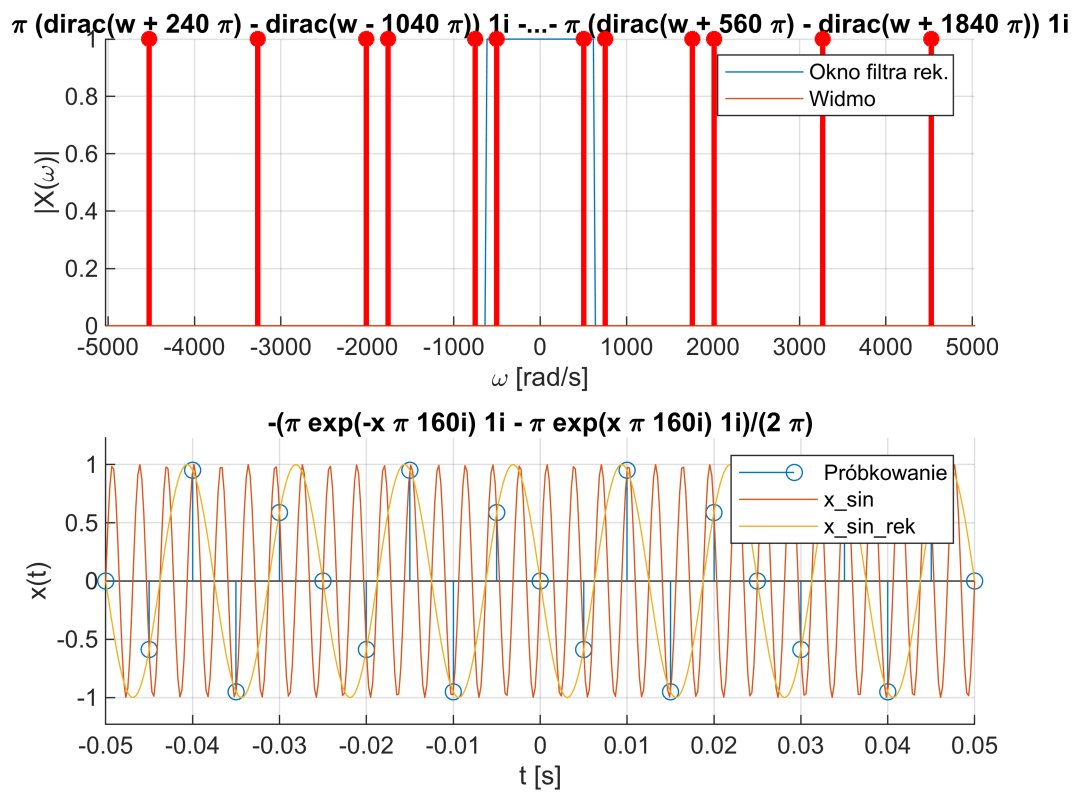
c)

fgplot(11/5)



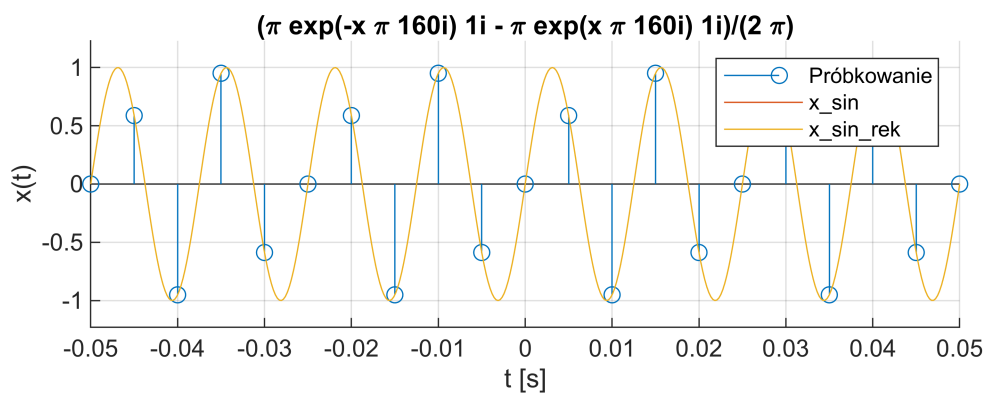
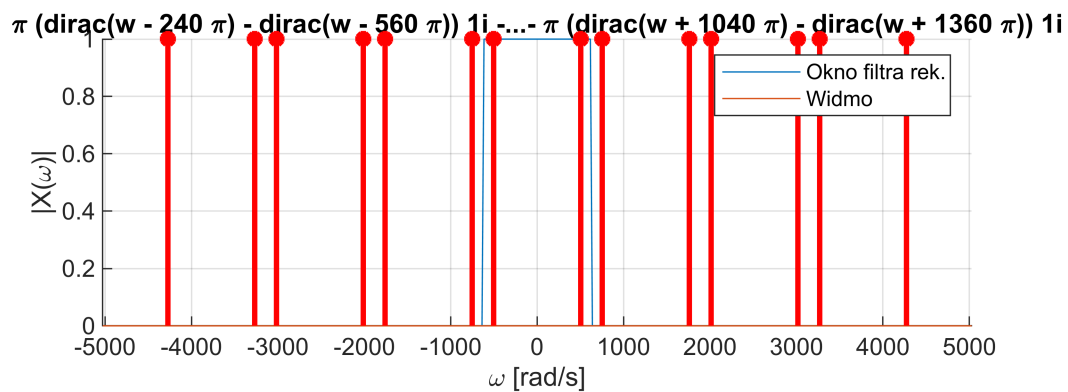
d)

fgplot(16/5)



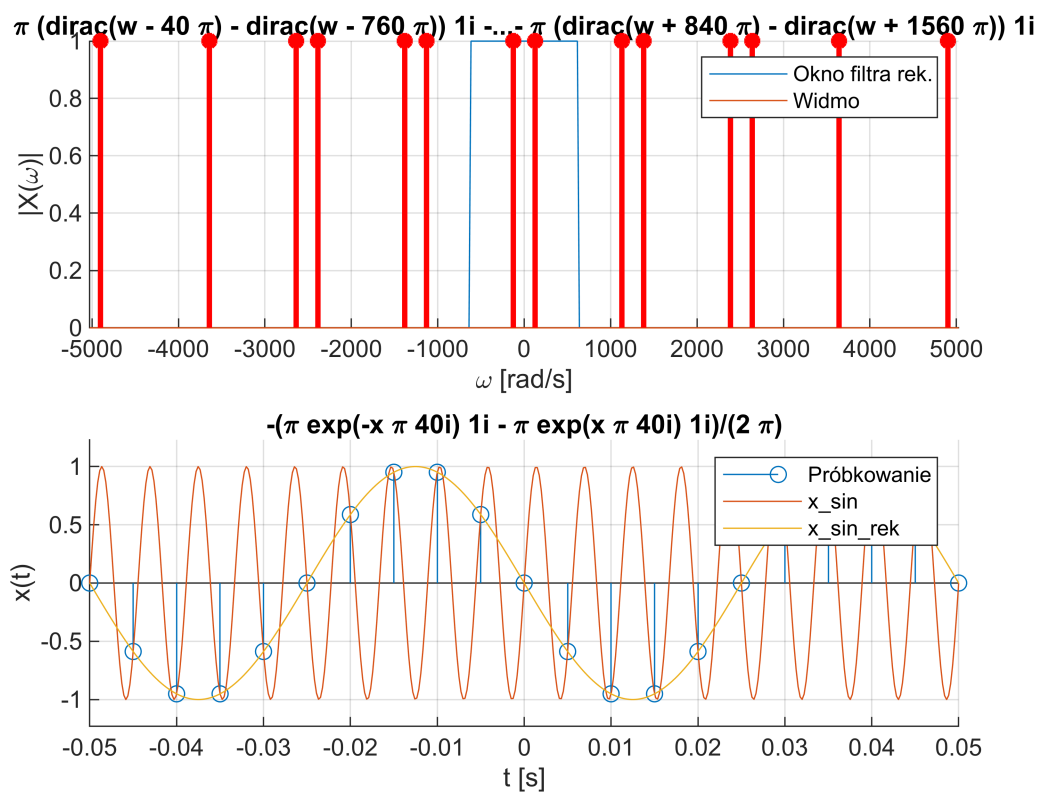
e)

fgplot(4/5)



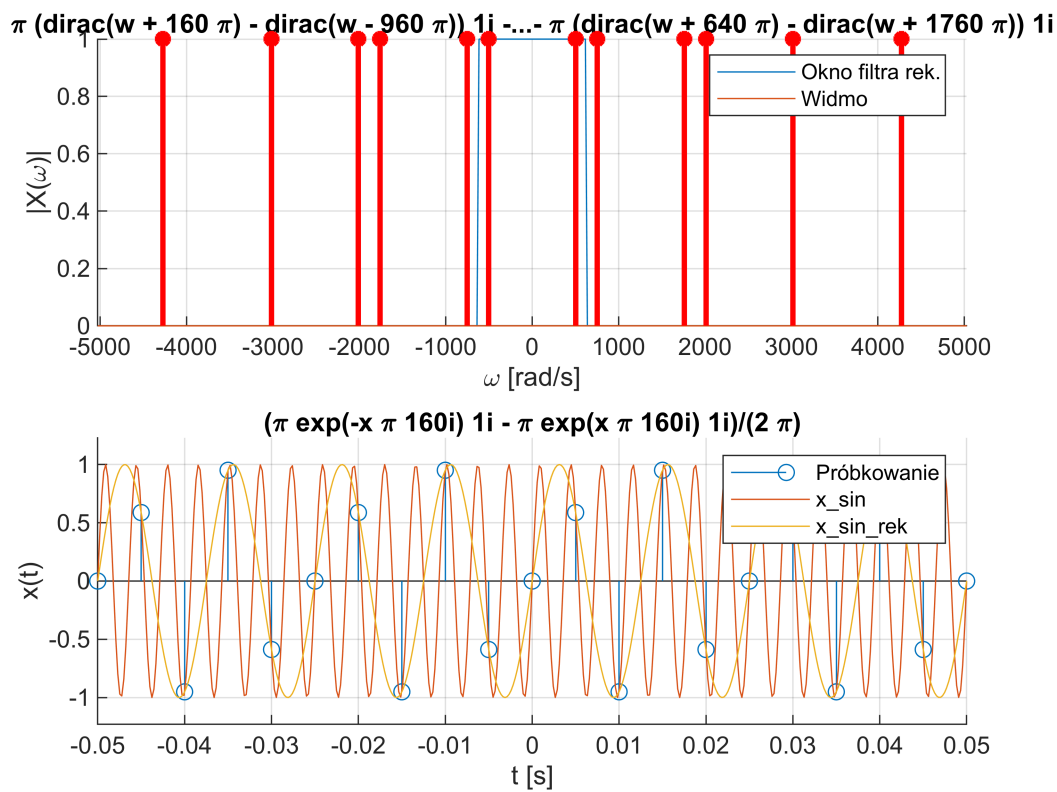
f)

fgplot(9/5)



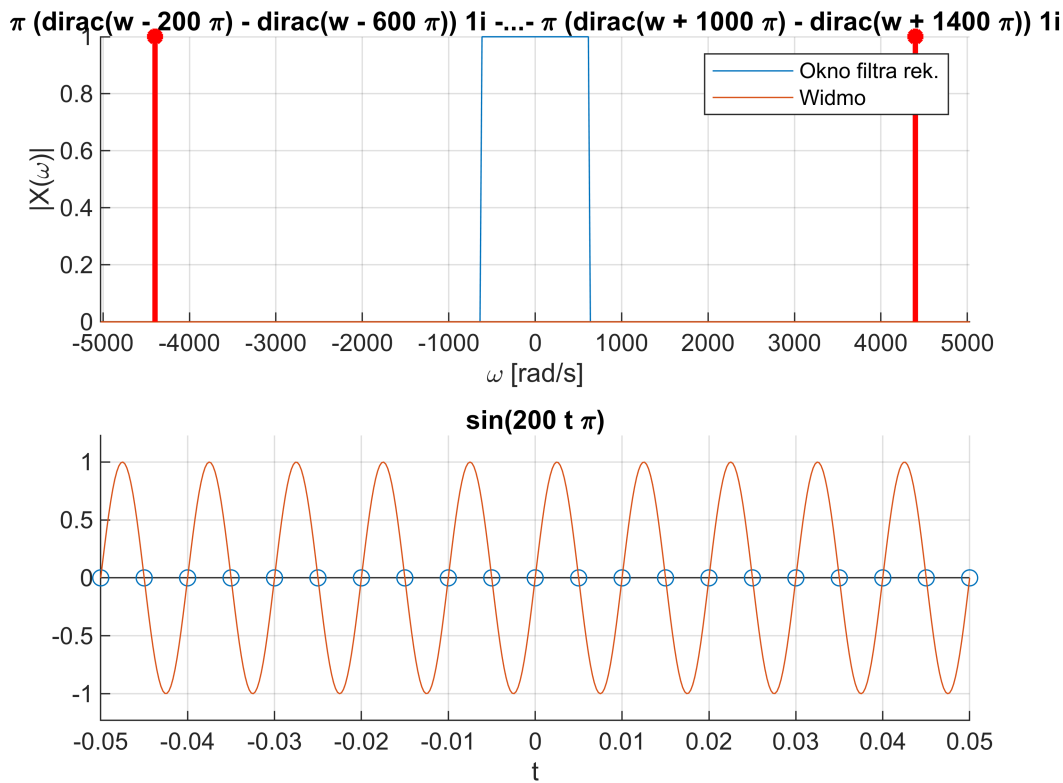
g)

```
fgplot(14/5)
```



Funkcja jest stała wtedy gdy częstotliwość próbkowania pozostaje w stosunku $1/k$ do częstotliwości sygnału próbkowanego gdzie k jest liczbą całkowitą.

```
fgplot(1)
```



```
Error using inlineeval
Error in inline expression ==> fourier(-rectangularPulse(-200.*pi, 200.*pi, w).*(pi.*(dirac(w - 200.*pi) -
dirac(w + 200.*pi)).*1i - pi.*(dirac(w - 200.*pi) - dirac(w - 600.*pi)).*1i +
pi.*(dirac(w + 200.*pi) - dirac(w + 600.*pi)).*1i - pi.*(dirac(w - 600.*pi) - dirac(w - 1000.*pi)).*1i +
pi.*(dirac(w + 600.*pi) - dirac(w + 1000.*pi)).*1i - pi.*(dirac(w - 1000.*pi) - dirac(w -
1400.*pi)).*1i + pi.*(dirac(w + 1000.*pi) - dirac(w + 1400.*pi)).*1i), w, -x)./(2.*pi)
Undefined function 'fourier' for input arguments of type 'double'.
```

```
Error in inline/feval (line 33)
    INLINE_OUT_ = inlineeval(INLINE_INPUTS_, INLINE_OBJ_.inputExpr, INLINE_OBJ_.expr); %#ok<DILEVAL>
```

```
Error in ezplotfeval (line 53)
    z = feval(f,x(1),y(1));
```

```
Error in ezplot>ezimplicit (line 266)
    u = ezplotfeval(f, X, Y);
```

```
Error in ezplot (line 162)
    hp = ezimplicit(cax, f{1}, vars, labels, args{:});
```

```
Error in sym/ezplot (line 78)
    h = ezplot(fhandle(f),varargin{:});%#ok<EZPLT>
```

```
Error in Lab_6>fgplot (line 150)
ezplot(x_sin_rek, BND_t) % syg. odworzony
```

Zadanie 4

Zastąp widmo sygnału sinusoidalnego X_{FT_sin} symetrycznym widmem o kształcie trójkątnym $X_A(j\omega)$ którego częstotliwość graniczna jest równa f_g , wartość minimalna wynosi 0.0 a maksymalna 1.0. Pomijamy wówczas obliczenia transformaty za pomocą funkcji `fourier()` ale musimy pamiętać o dodaniu aliasów po prawej i lewej stronie. Przeprowadź analizę jak w **Zad. 3 1.3**.

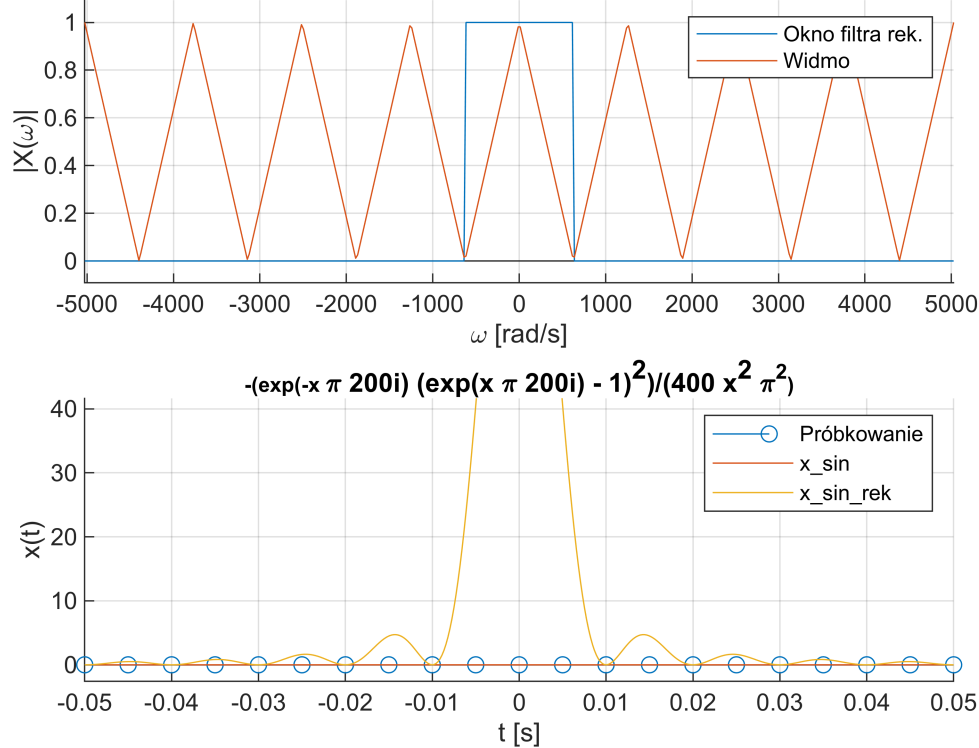
```
clear all; close all;
syms t x w K
fp = 200; fg = fp/2; %Hz
wp = 2*pi*fp; wg = 2*pi*fg;
s = 4/5; ws = s*wg;
x_sin = sym(0);

X_FT_sin_org = triangularPulse(-wg, wg, w);
X_FT_sin = X_FT_sin_org + symsum((subs(X_FT_sin_org, w, w - K*wp ) + subs(X_FT_sin_org, w, w +
FILT_FT = rectangularPulse(-wg, wg, w); % filtr rekonstruuujący
x_sin_rek = ifourier(X_FT_sin*FILT_FT); % odwr. tarnsf. Fouriera
BND_t = [-10/fp; 10/fp];
t_SMP = [BND_t(1):1/(10*fp):BND_t(2) ];
BND_w = [-4*wp; 4*wp];
w_SMP = [BND_w(1):wp/10:BND_w(2)];

figure; subplot(2,1,1); hold on; grid on;
title("Trójkątnba transformata")
ezplot(FILT_FT, BND_w); %okno filtru rek.
ezplot(X_FT_sin, BND_w)
v_num = abs(double(subs(X_FT_sin, w, w_SMP)));
n = find(abs(v_num) == Inf);
stem(w_SMP(n), sign(v_num(n)), 'r*', 'LineWidth', 2);
xlabel('\omega [rad/s]'); ylabel('|X(\omega)|')
legend('Okno filtra rek.', 'Widmo');

subplot(2,1,2); hold on; grid on;
title('Rekonstrukcja')
nodex = BND_t(1):1/fp:BND_t(2);
nodey = subs(x_sin, t, nodex);
stem(nodex, nodey);
ezplot(x_sin, BND_t);
% syg. próbkowany
ezplot(x_sin_rek, BND_t) % syg. odtworzony
xlabel('t [s]'); ylabel('x(t)')
legend('Próbkowanie', 'x_sin', 'x_sin_rek');
```

`triangularPulse(-200 π , 0, 200 π , w) + ... + triangularPulse(-200 π , 0, 200 π , w + 2000 π)`



Zadanie 5

Zadanie polega na rekonstrukcji próbkowanej, nieskończonej symetrycznej fali prostokątnej o częstotliwości $f_s = \frac{4}{5}f_g$, wartości średniej **0.5**, amplitudzie **1.0** i współczynnika wypełnienia równym **0.5**. Ze względu na nieskończoną reprezentację tego sygnału w dziedzinie czasu, najlepiej zdefiniować go jako obraz częstotliwościowy w dziedzinie pulsacji, stosując formułę [8]. Współczynniki X_n szeregu Fouriera można wyznaczyć komputerowo jak w ćwiczeniu Lab. *Analiza harmoniczna sygnałów* albo korzystając z tablic. W tym przypadku szereg będzie nieskończony, jednak do symulacji można wykorzystać kilkanaście (kilkadziesiąt) pierwszych wyrazów ciągu, np. $n \in (-20, 20)$. Wykonaj rekonstrukcję sygnału sinusoidalnego o następujących częstotliwościach f_s :

- $\frac{1}{5}f_g$,
- $\frac{4}{5}f_g$,
- f_g ,
- $\frac{6}{5}f_g$.

Napisz w sprawozdaniu, w którym przypadku rekonstrukcja daje najlepsze efekty. Odpowiedź uzasadnij.

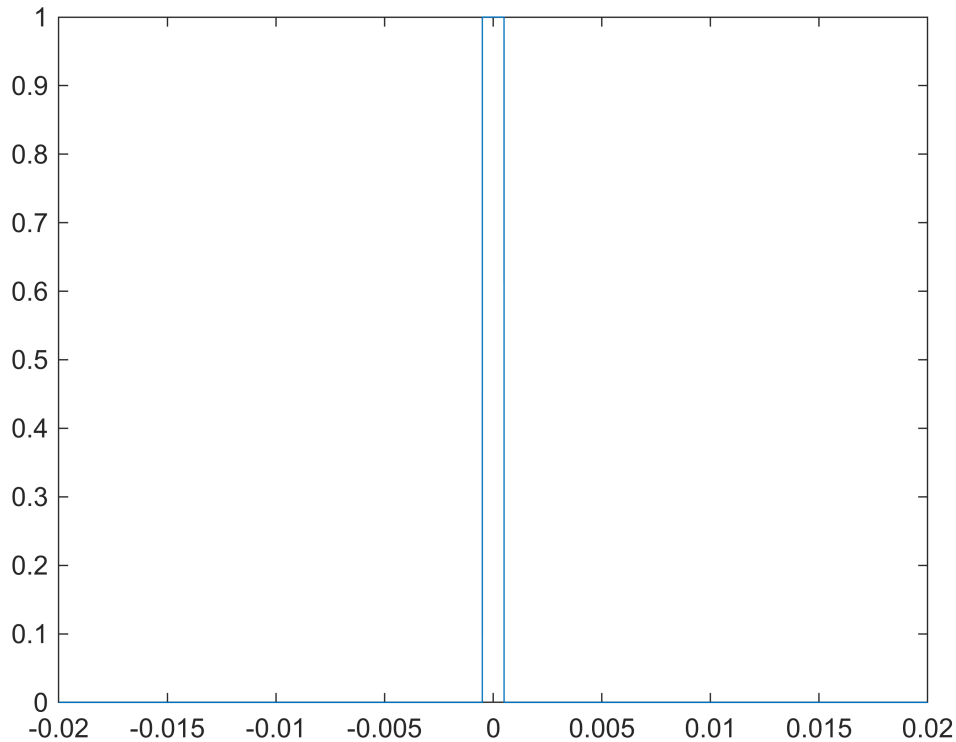
$s = 1/5$;

```

syms t x w K
fp = 200; fg = fp/2; %Hz
wp = 2*pi*fp; wg = 2*pi*fg;
ws = s*wg;
Ts = s/fg;
x_sin = rectangularPulse(-Ts/4, Ts/4, t);

X_sin = x_sin + symsum(subs(x_sin, t, t - K/Ts ) + (subs(x_sin, t, t + K/Ts )), K , 1, 3);
figure
fplot(X_sin,[-0.02,0.02])

```



```

function [] = fgplot(s)
syms t x w K
fp = 200; fg = fp/2; %Hz
wp = 2*pi*fp; wg = 2*pi*fg;
ws = s*wg;
x_sin = sin(ws*t);
X_FT_sin_org = fourier(x_sin);
X_FT_sin = X_FT_sin_org + ... % oryginal widma
    symsum((subs(X_FT_sin_org, w, w - K*wp ) + ... % 3 aliasy lewe
    subs(X_FT_sin_org, w, w + K*wp)), K , 1, 3); % 3 aliasy prawe
FILT_FT = rectangularPulse(-wg,wg,w); % filtr rekonstruujący
x_sin_rek = ifourier(X_FT_sin*FILT_FT); % odwr. tarnsf. Fouriera
BND_t = [-10/fp;10/fp];
t_SMP = [BND_t(1):1/(10*fp):BND_t(2) ];
BND_w = [-4*wp;4*wp];

```

```

w_SMP = [BND_w(1):wp/10:BND_w(2)];
figure; subplot(2,1,1); hold on; grid on;
ezplot(FILT_FT,BND_w); %okno filtru rek.
ezplot(X_FT_sin,BND_w)
v_num = abs(double(subs(X_FT_sin, w, w_SMP)));
n = find(abs(v_num) == Inf);
stem(w_SMP(n),sign(v_num(n)), 'r*', 'LineWidth', 2);
xlabel('\omega [rad/s]'); ylabel('|X(\omega)|')
legend('Okno filtra rek.', 'Widmo');
subplot(2,1,2); hold on; grid on;
nodex = BND_t(1):1/fp:BND_t(2);
nodey = subs(x_sin, t, nodex);
stem(nodex, nodey);
ezplot(x_sin, BND_t);
% syg. próbkowany
ezplot(x_sin_rek, BND_t) % syg. odtworzony
xlabel('t [s]'); ylabel('x(t)')
legend('Próbkowanie', 'x\_sin', 'x\_sin\_rek');
end

function [] = sqfgplot(s)
syms t x w K
fp = 200; fg = fp/2; %Hz
wp = 2*pi*fp; wg = 2*pi*fg;
ws = s*wg;
x_sin = rectangularPulse(-ws/2, ws/2, t);

X_sin = x_sin + symsum(subs(x_sin, t, t - K*wp ) + (subs(x_sin, t, t + K*wp )), K , 1, 3);
X_FT_sin = fourier(X_sin);
FILT_FT = rectangularPulse(-wg,wg,w); % filtr rekonstruujący
x_sin_rek = ifourier(X_FT_sin*FILT_FT); % odwr. tarnsf. Fouriera
BND_t = [-10/fp;10/fp];
t_SMP = [BND_t(1):1/(10*fp):BND_t(2) ];
BND_w = [-4*wp;4*wp];
w_SMP = [BND_w(1):wp/10:BND_w(2)];
figure; subplot(2,1,1); hold on; grid on;
ezplot(FILT_FT,BND_w); %okno filtru rek.
ezplot(X_FT_sin,BND_w)
v_num = abs(double(subs(X_FT_sin, w, w_SMP)));
n = find(abs(v_num) == Inf);
stem(w_SMP(n),sign(v_num(n)), 'r*', 'LineWidth', 2);
xlabel('\omega [rad/s]'); ylabel('|X(\omega)|')
legend('Okno filtra rek.', 'Widmo');
subplot(2,1,2); hold on; grid on;
nodex = BND_t(1):1/fp:BND_t(2);
nodey = subs(x_sin, t, nodex);
stem(nodex, nodey);
ezplot(x_sin, BND_t);
% syg. próbkowany
ezplot(x_sin_rek, BND_t) % syg. odtworzony

```

```
xlabel('t [s]'); ylabel('x(t)')  
legend('Próbkowanie', 'x_sin', 'x_sin_rek');  
end
```