Przetwarzanie Sygnałów Cyfrowych

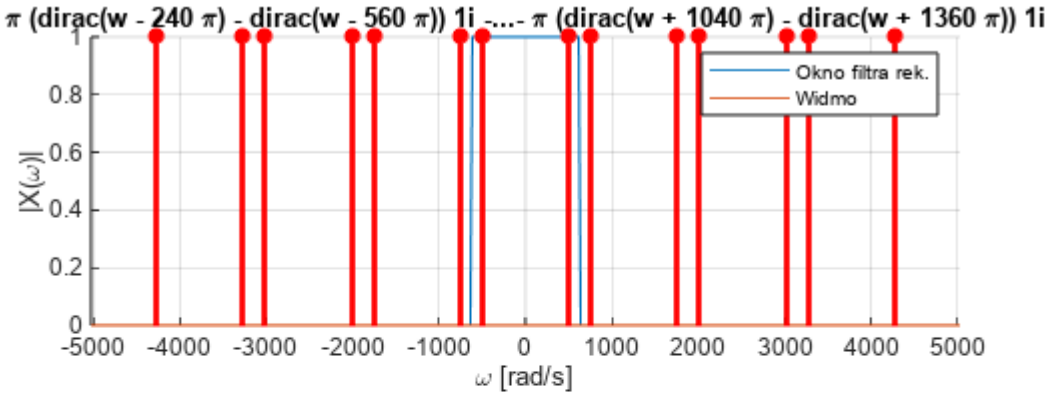
Próbkowanie sygnałów ciągłych

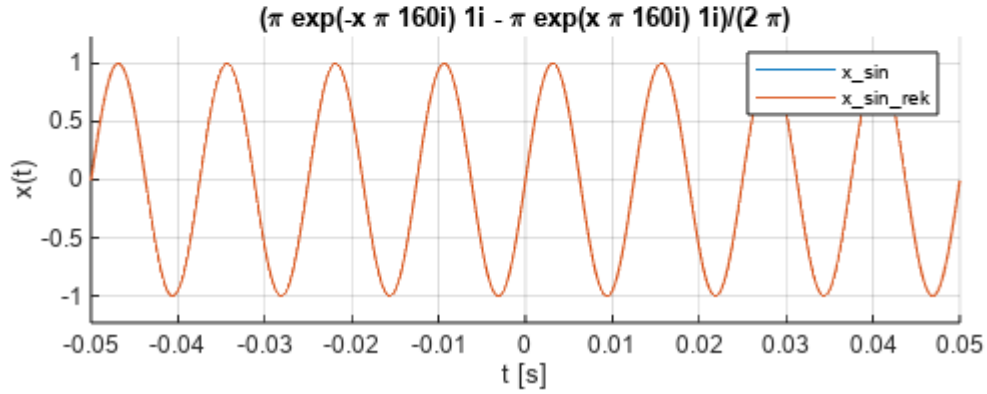
Jan Rosa 410269 AiR

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 rekonstruuj¡cy
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');
```



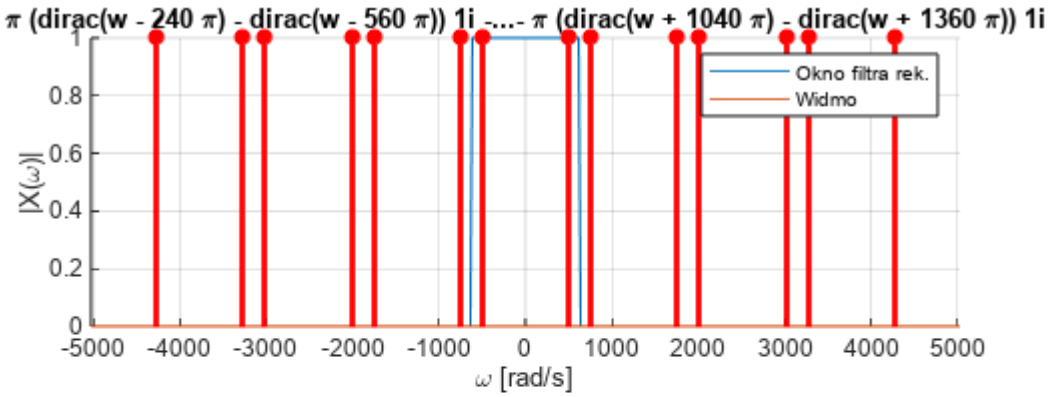


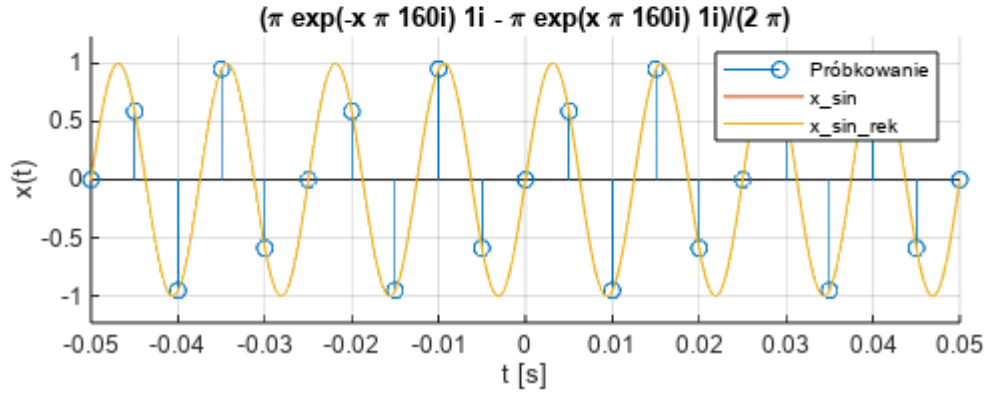
Zadanie 2

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 + ... % 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 rekonstruuj¡cy
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');
```





Zadanie 3

Wykonaj rekonstrukcję sygnału sinusoidalnego o następujących częstotliwościach:

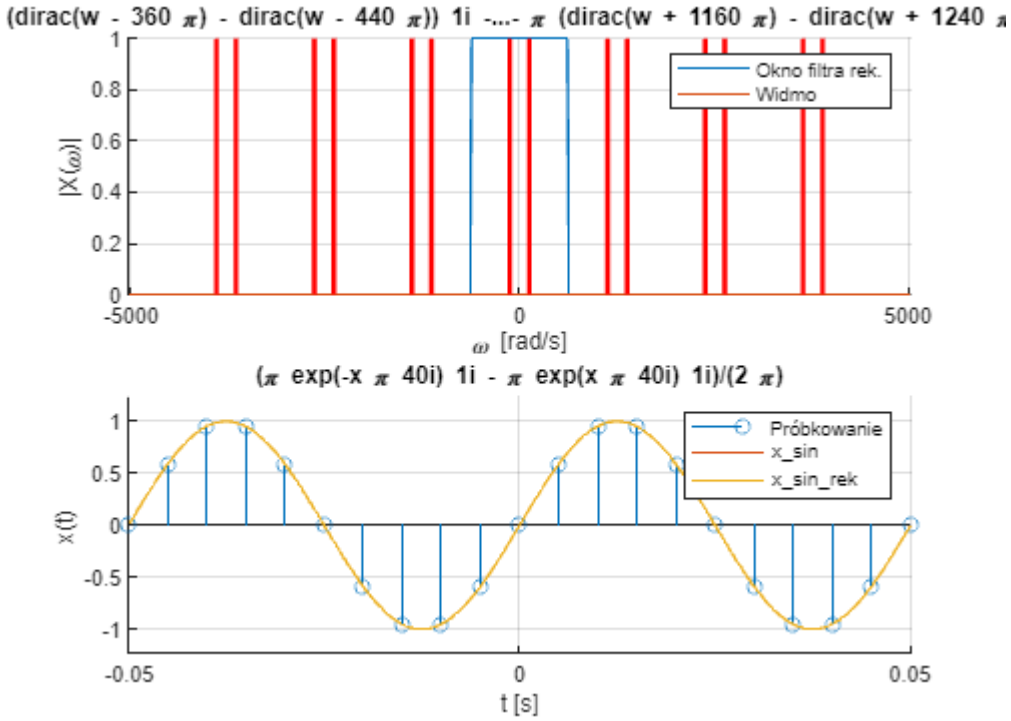
- a) $\frac{1}{5}f_g$, b) $\frac{6}{5}f_g$, c) $\frac{11}{5}f_g$, d) $\frac{16}{5}f_g$,
- e) $\frac{4}{5}f_g$, f) $\frac{9}{5}f_g$, g) $\frac{14}{5}f_g$.

Uwaga: aby ustawić częstotliwość sinusoidy należy zmodyfikować wartość współczynnika `s` który wyznacza pulsację `ws` sygnału `x_sin=sin(ws*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.

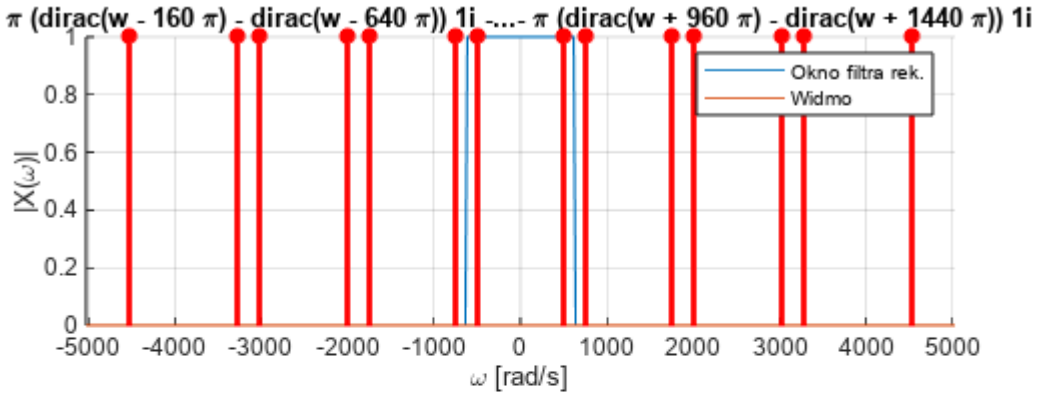
a)

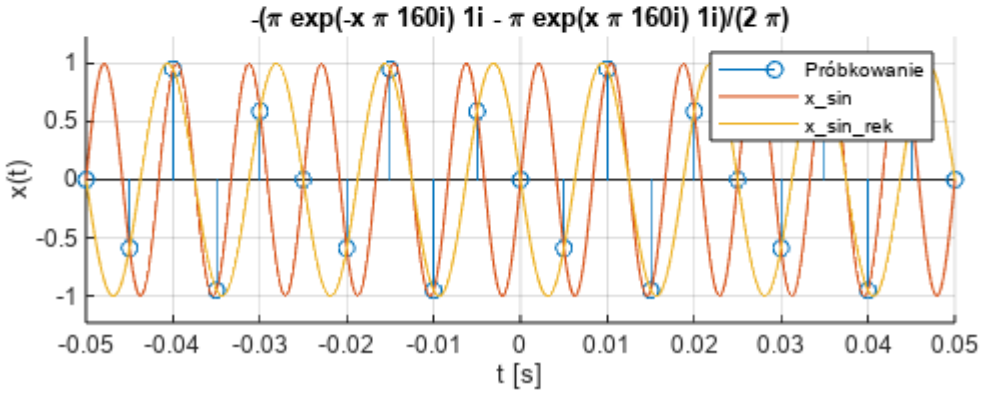
```
fgplot(1/5)
```



b)

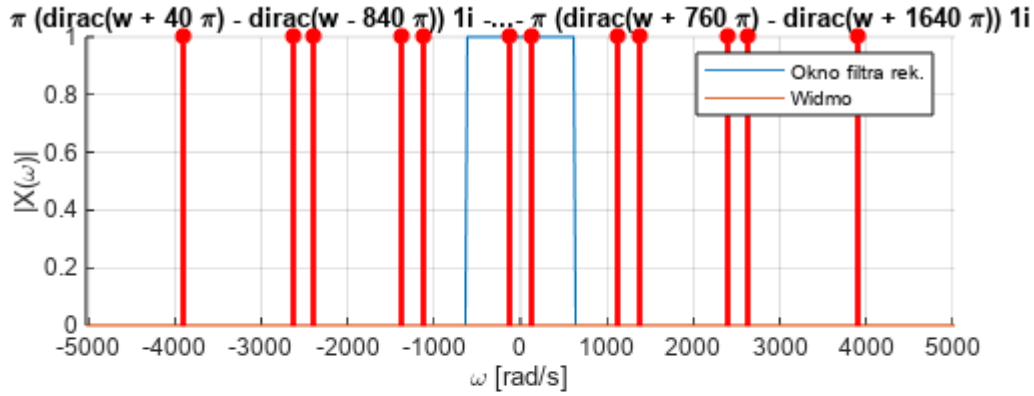
```
fgplot(6/5)
```

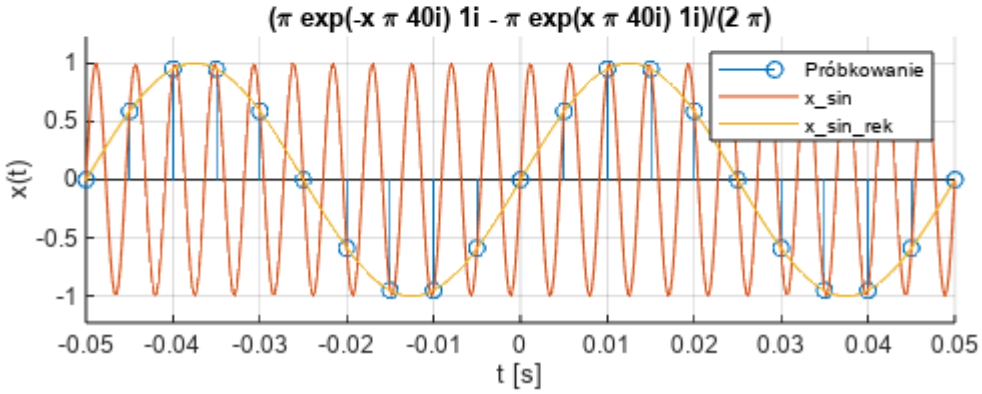

π (dirac(w - 160 π) - dirac(w - 640 π)) 1i -...- π (dirac(w + 960 π) - dirac(w + 1440 π)) 1i
Okno filtra rek.
Widmo
|X(ω)|
ω [rad/s]


-(π exp(-x π 160i) 1i - π exp(x π 160i) 1i)/(2 π)
Próbkowanie
x_sin
x_sin_rek
x(t)
t [s]

c)

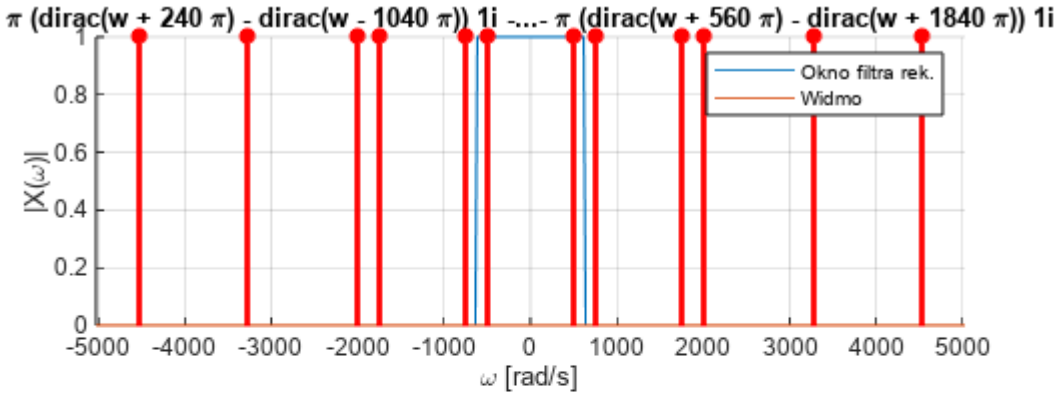
```
fgplot(11/5)
```

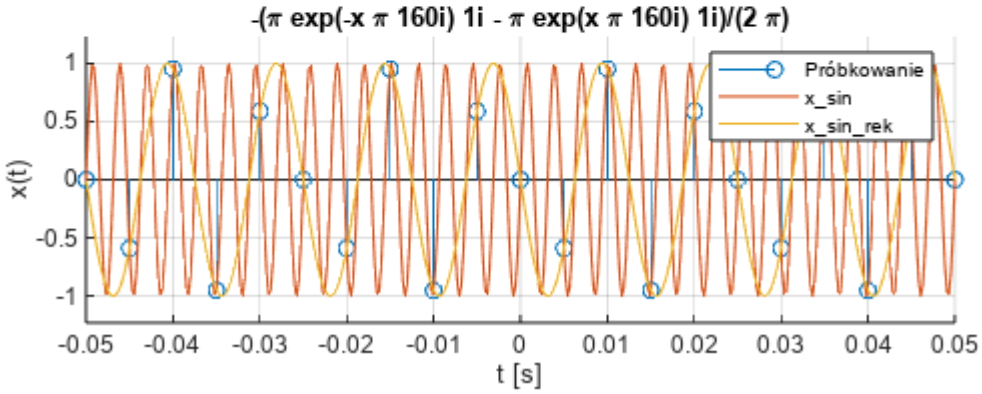

π (dirac(w + 40 π) - dirac(w - 840 π)) 1i -...- π (dirac(w + 760 π) - dirac(w + 1640 π)) 1i
Okno filtra rek.
Widmo
|X(ω)|
ω [rad/s]


(π exp(-x π 40i) 1i - π exp(x π 40i) 1i)/(2 π)
Próbkowanie
x_sin
x_sin_rek
x(t)
t [s]

d)

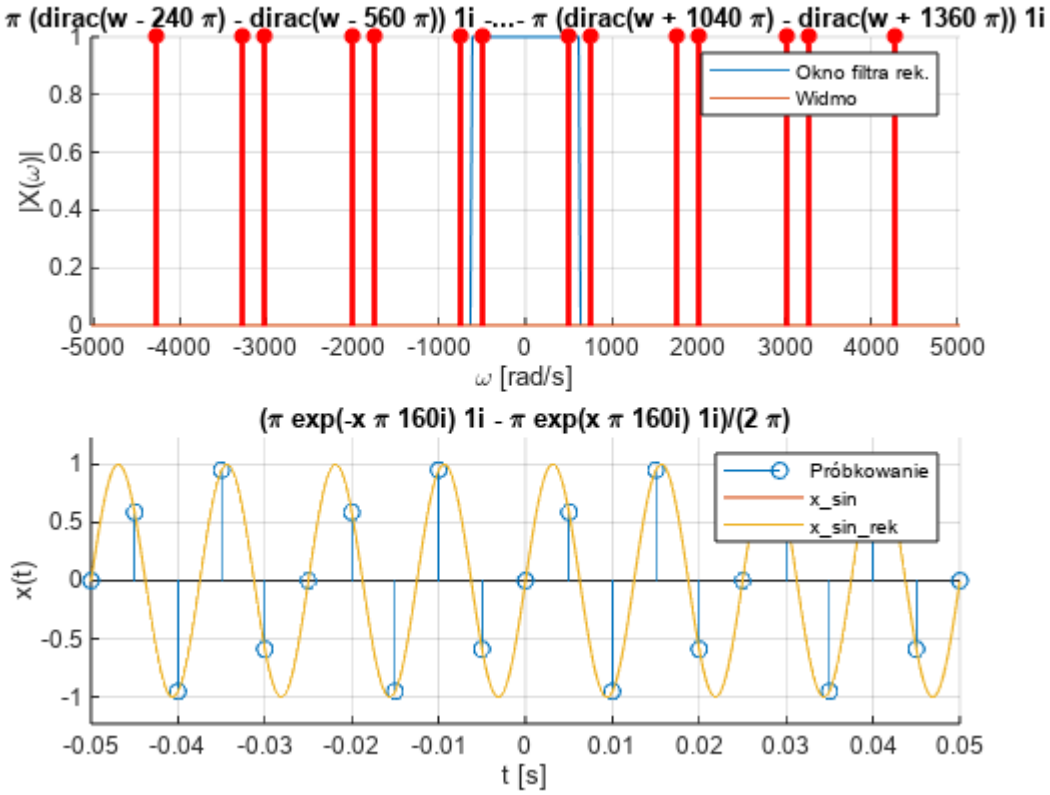
```
fgplot(16/5)
```


π (dirac(w + 240 π) - dirac(w - 1040 π)) 1i -...- π (dirac(w + 560 π) - dirac(w + 1840 π)) 1i
Okno filtra rek.
Widmo
|X(ω)|
ω [rad/s]


-(π exp(-x π 160i) 1i - π exp(x π 160i) 1i)/(2 π)
Próbkowanie
x_sin
x_sin_rek
x(t)
t [s]

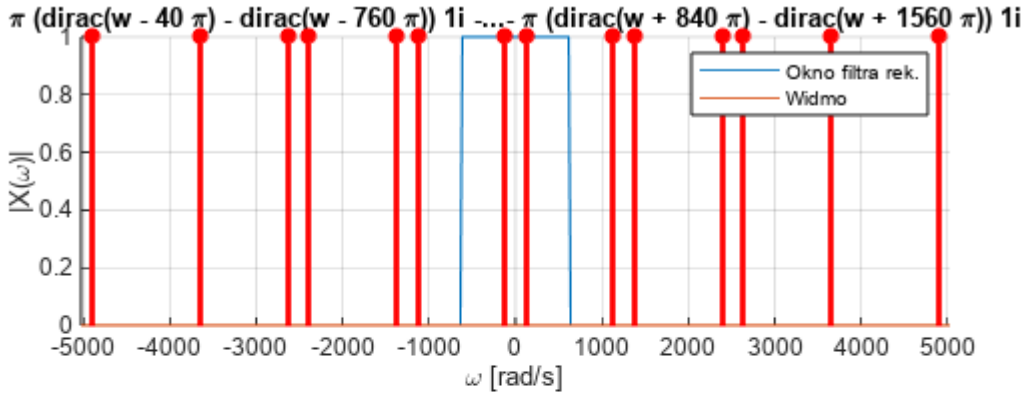
e)

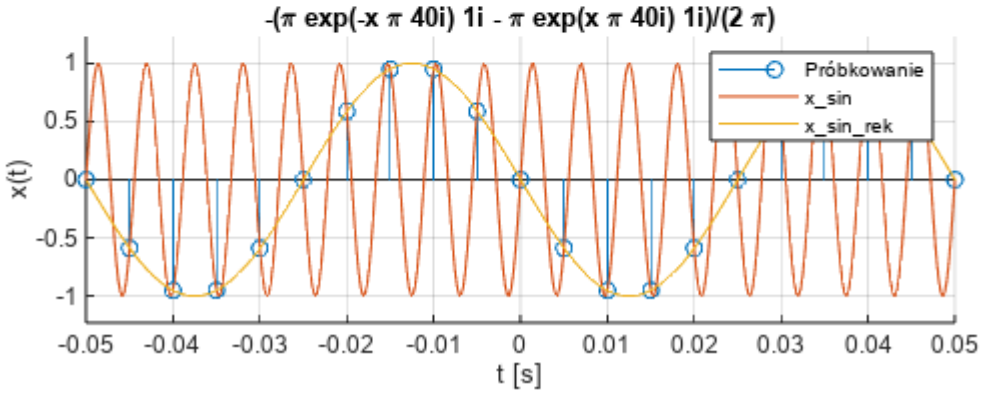
```
fgplot(4/5)
```


π (dirac(w - 240 π) - dirac(w - 560 π)) 1i -...- π (dirac(w + 1040 π) - dirac(w + 1360 π)) 1i
Okno filtra rek.
Widmo
|X(ω)|
ω [rad/s]
(π exp(-x π 160i) 1i - π exp(x π 160i) 1i)/(2 π)
Próbkowanie
x_sin
x_sin_rek
x(t)
t [s]

f)

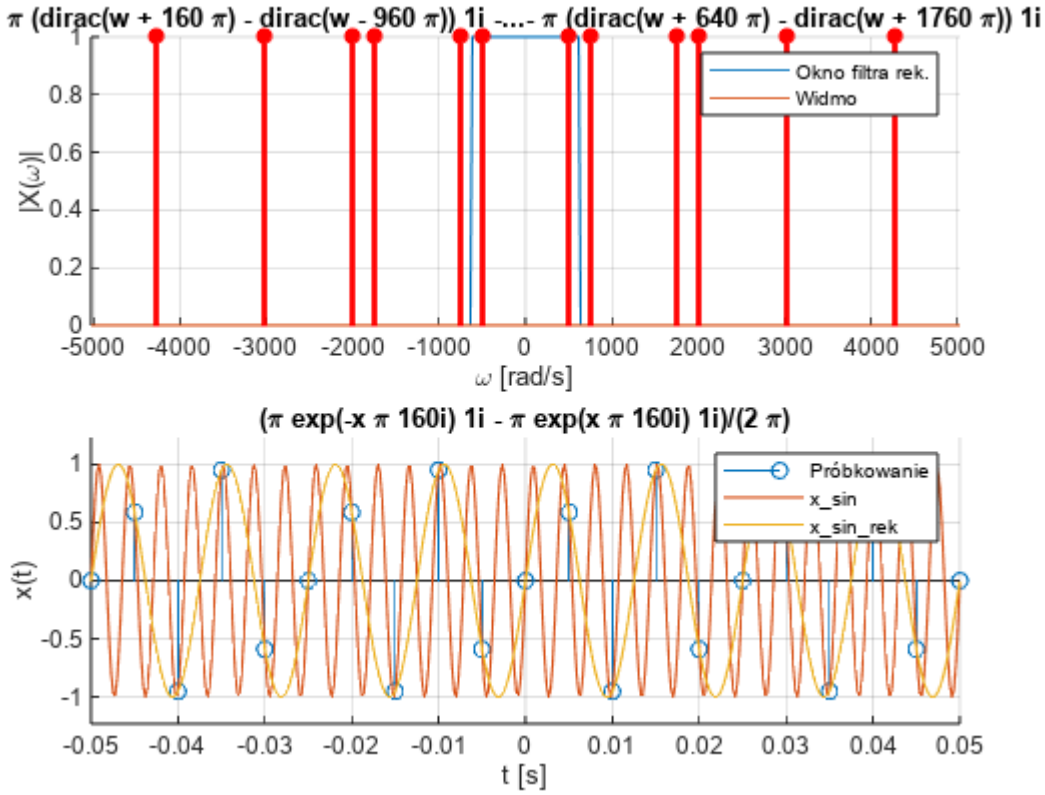
```
fgplot(9/5)
```


π (dirac(w - 40 π) - dirac(w - 760 π)) 1i -...- π (dirac(w + 840 π) - dirac(w + 1560 π)) 1i
Okno filtra rek.
Widmo
|X(ω)|
ω [rad/s]


-(π exp(-x π 40i) 1i - π exp(x π 40i) 1i)/(2 π)
Próbkowanie
x_sin
x_sin_rek
x(t)
t [s]

g)

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

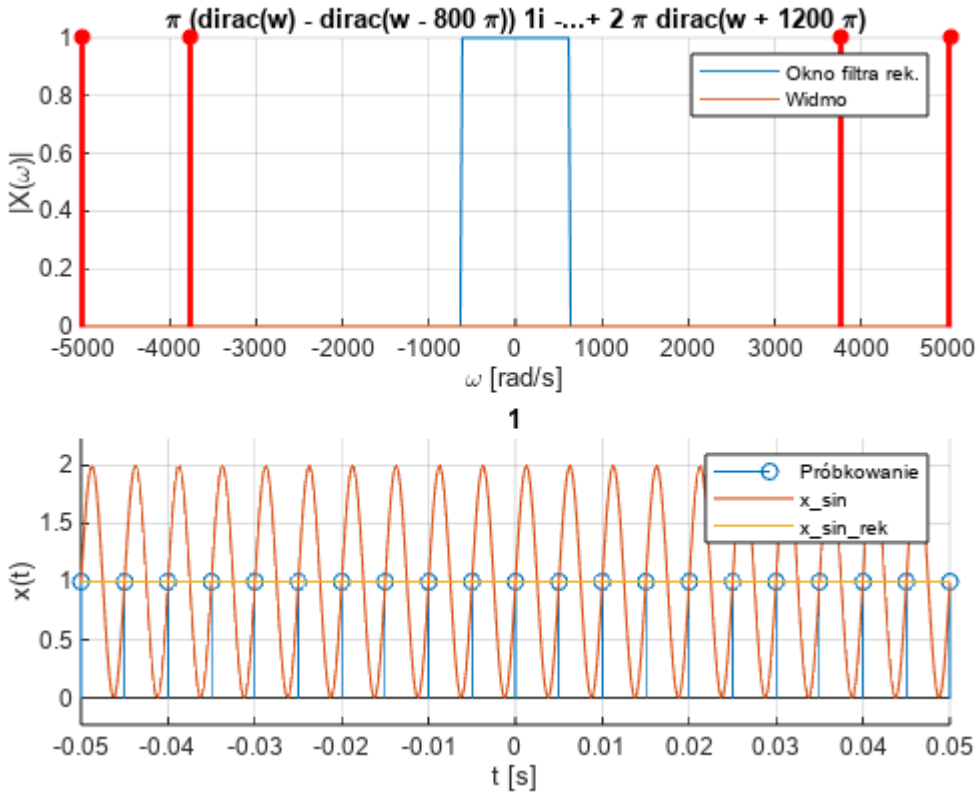


Najlepeij odtworzona jest funkcja gdy s<1 tj gdy częstotliwość próbkowania jest co najmniej dwa razy większa niż najmniejsza składowa harmoniczna, **Twierdzenie o próbkowaniu**, **twierdzenie Kotelnikowa Shanona.**

Funkcja jest stała wtedy gdy częstowliwość próbkownia pozostaje w stosunku 1/(k+1) do częstotliwości sygnału próbkowaniego gdzie k jest liczbą całkowitą.

```
syms t x w K
s = 2;
fp = 200; fg = fp/2; %Hz
wp = 2*pi*fp; wg = 2*pi*fg;
ws = s*wg;
x_sin = sin(ws*t) + 1;
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 rekonstruuj¡cy
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');
```

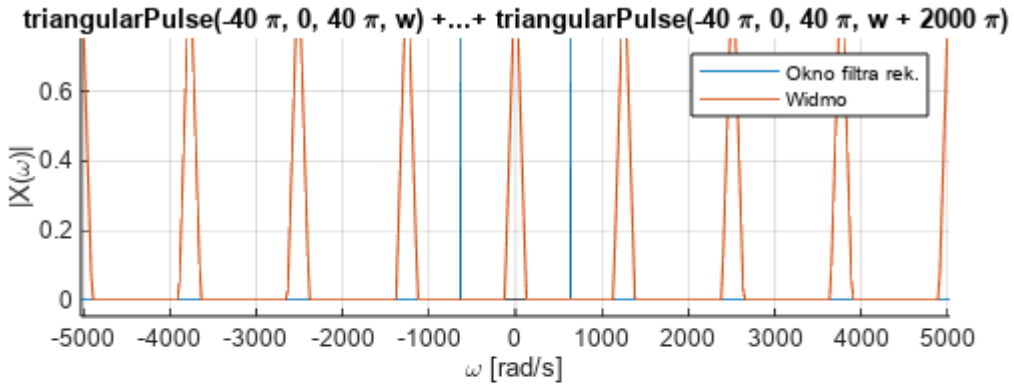


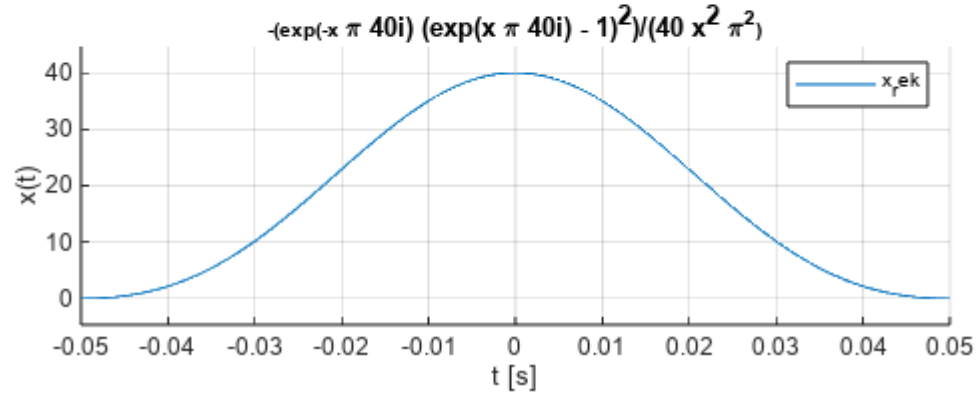
Zadanie 4

Zastąp widmo sygnału sinusiodalnego `X_FT_sin` symetrycznym widmem o kształcie trójkątnym $X_\Lambda(j\omega)$ którego częstotliwość graniczna jest równa f_g, wartość minimalna wynosi 0.0 a maksymlana 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.

a)

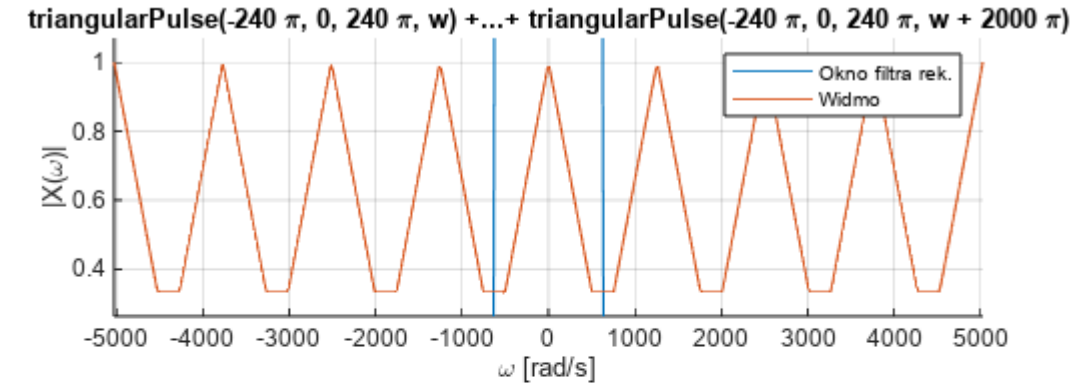
```
trianwave(1/5)
```

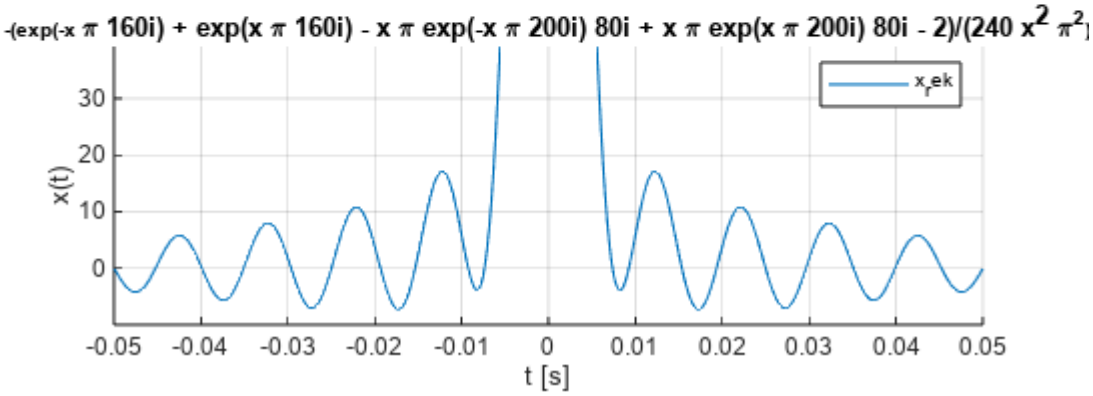




b)

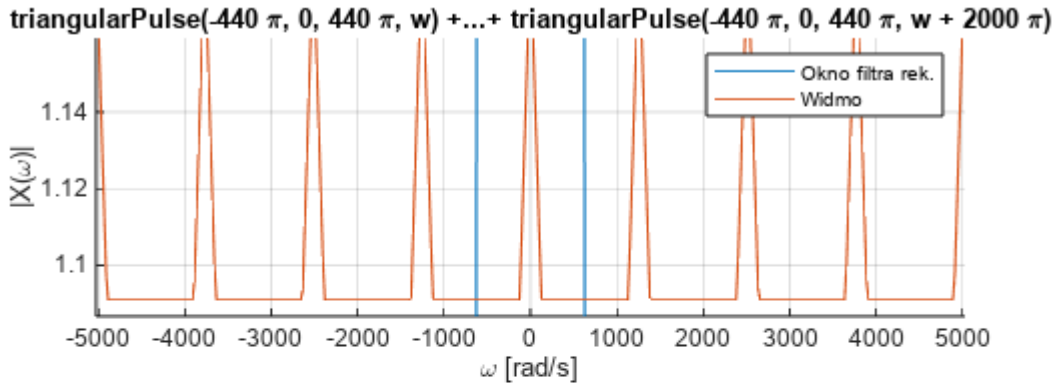
```
trianwave(6/5)
```

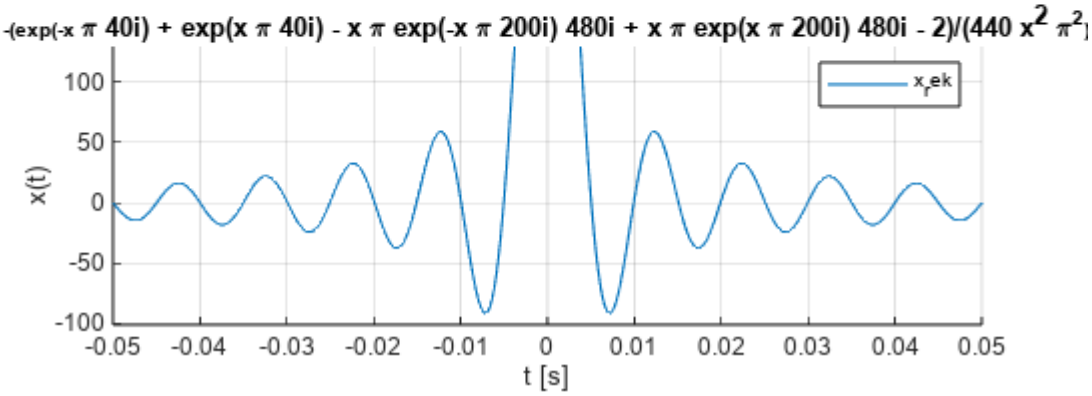




c)

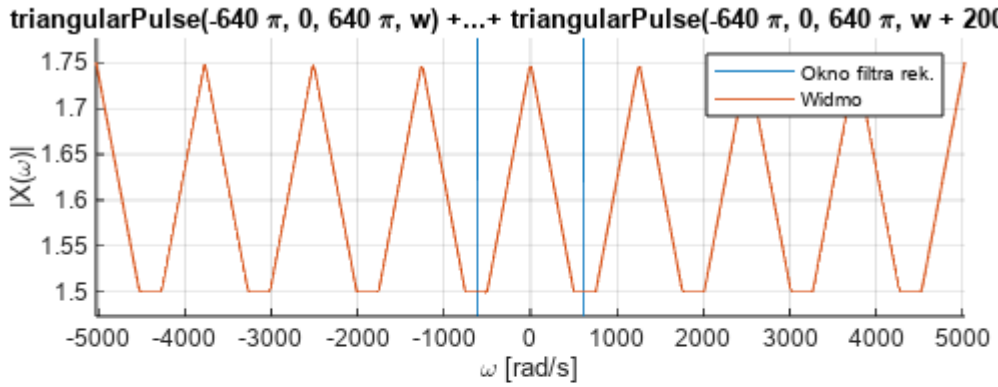
```
trianwave(11/5)
```

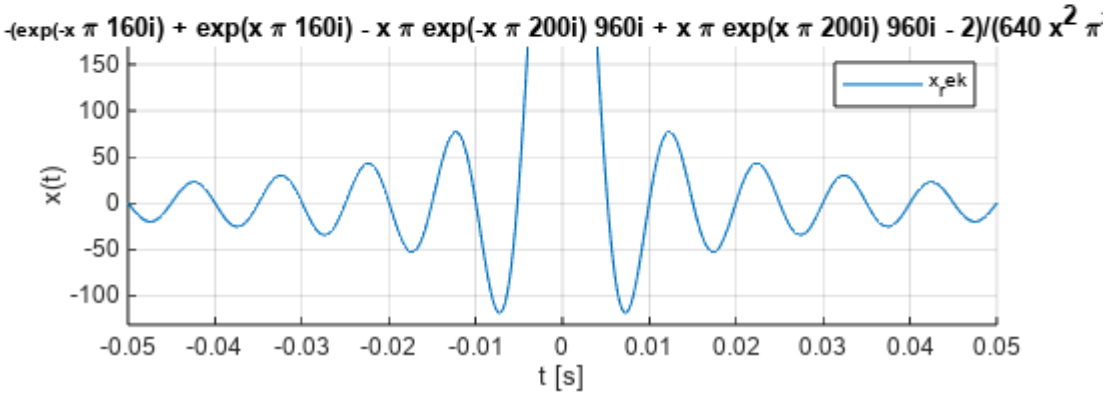




d)

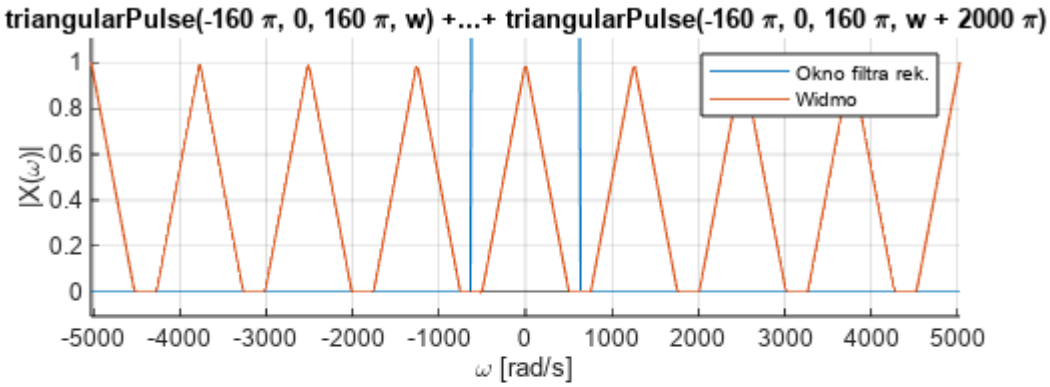
```
trianwave(16/5)
```

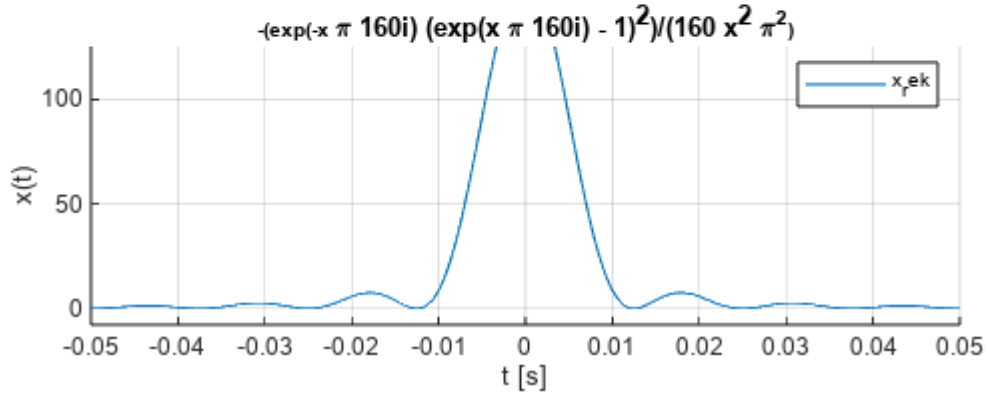




e)

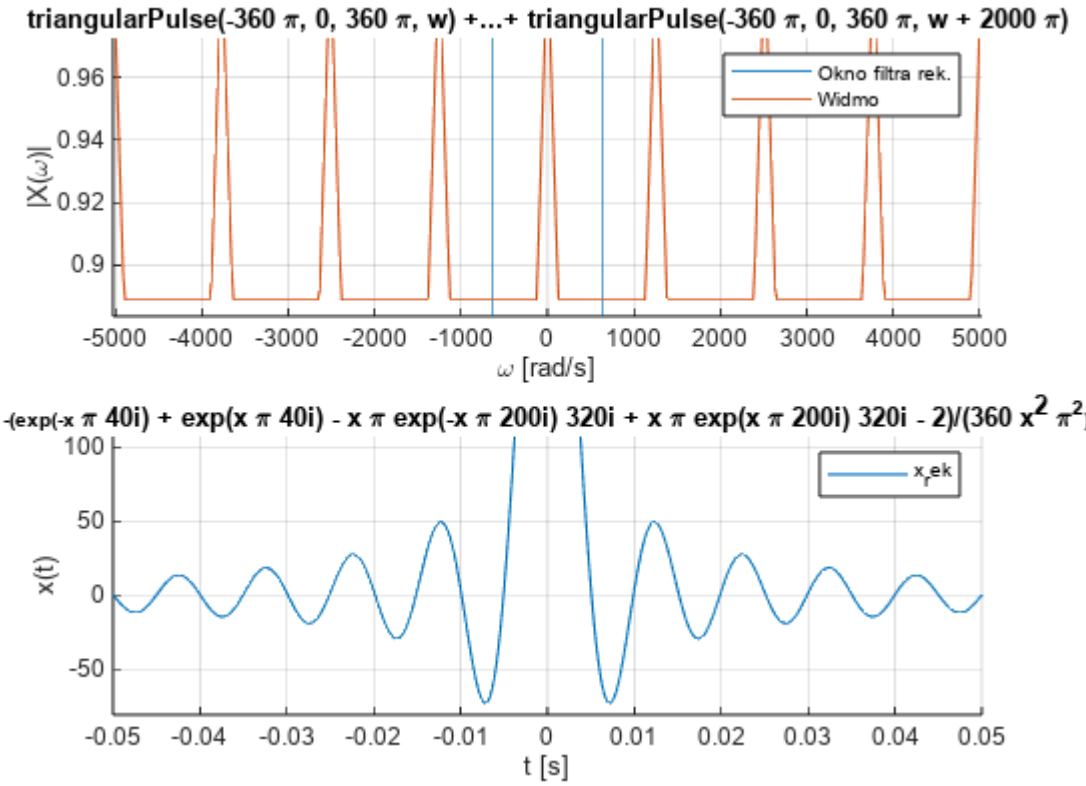
```
trianwave(4/5)
```





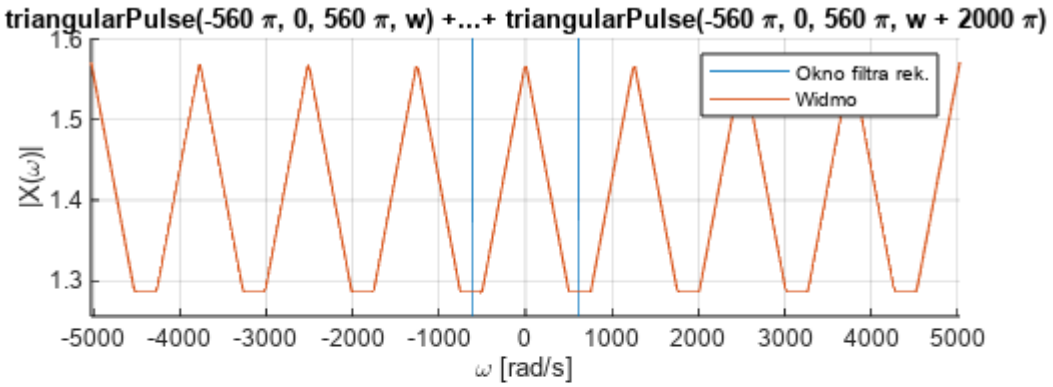
f)

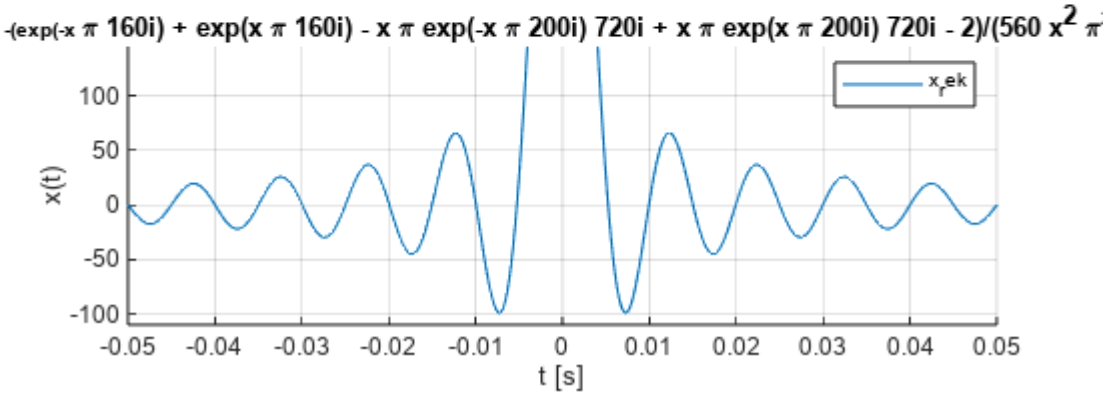
```
trianwave(9/5)
```



g)

```
trianwave(14/5)
```





Zadanie 5

Zadanie polega na rekonstrukcji spróbkowanej, nieskończonej symetrycznej fali prostokątnej o częstotliwości $f_s = \frac{4}{5} f_g$, wartości średniej 0.5, amlplitudzie 1.0 i współczynniku wypełnienienia 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 Xn 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 kilkananaście (kilkadziesiąt) pierwszych wyrazów ciągu, np. $n \in (-20, 20)$. Wykonaj rekonstrukcję sygnału sinusoidalnego o następujących częstotliwościach f_s:

a) $\frac{1}{5} f_g$,

b) $\frac{4}{5} f_g$,

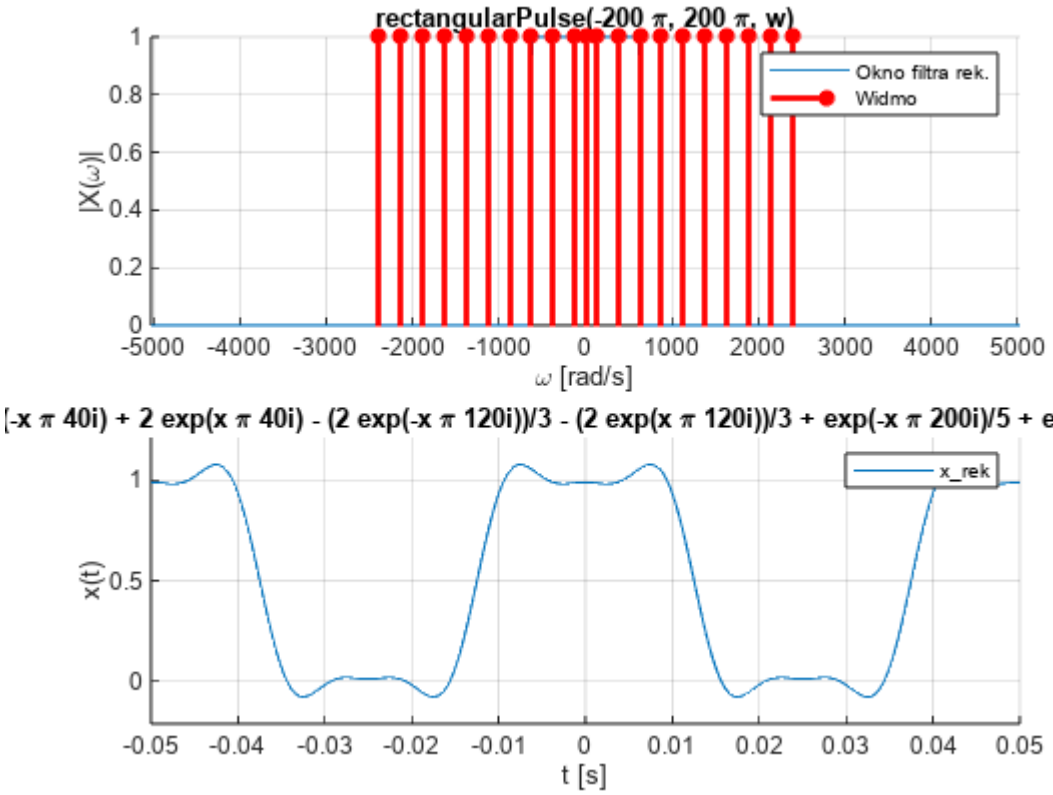
c) f_g,

d) $\frac{6}{5} f_g$.

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

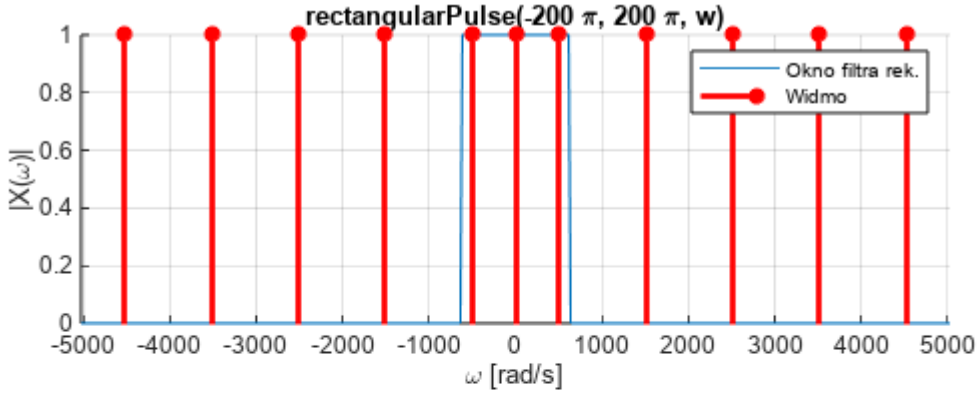
a)

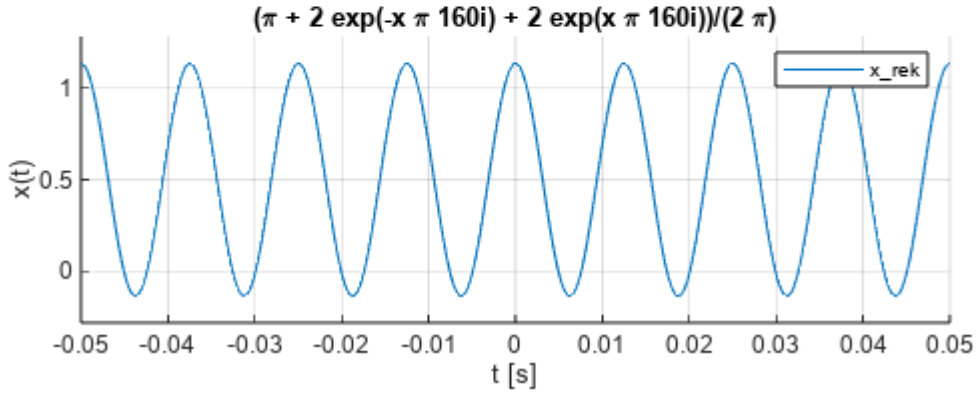
```
rectwave(1/5)
```



b)

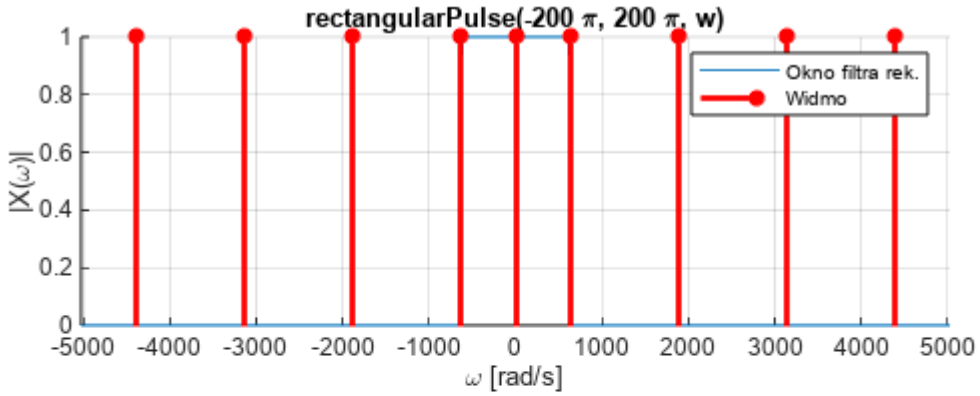
```
rectwave(4/5)
```

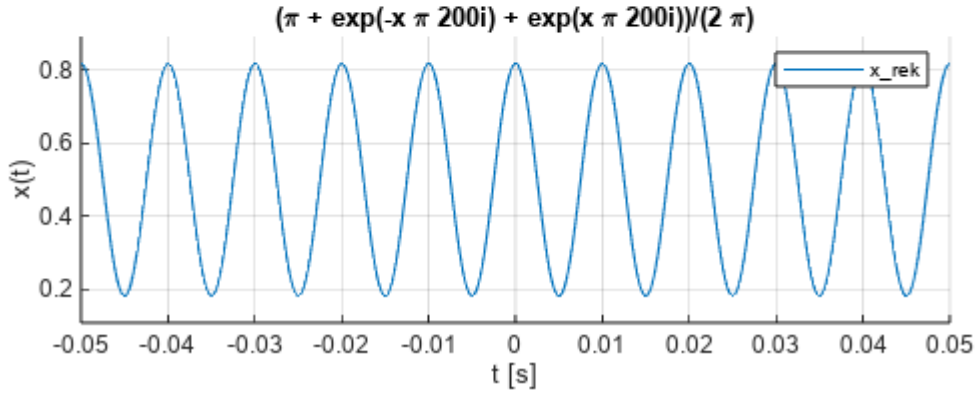

rectangularPulse(-200 π, 200 π, w)
Okno filtra rek.
Widmo
|X(ω)|
ω [rad/s]


(π + 2 exp(-x π 160i) + 2 exp(x π 160i))/(2 π)
x_rek
x(t)
t [s]

c)

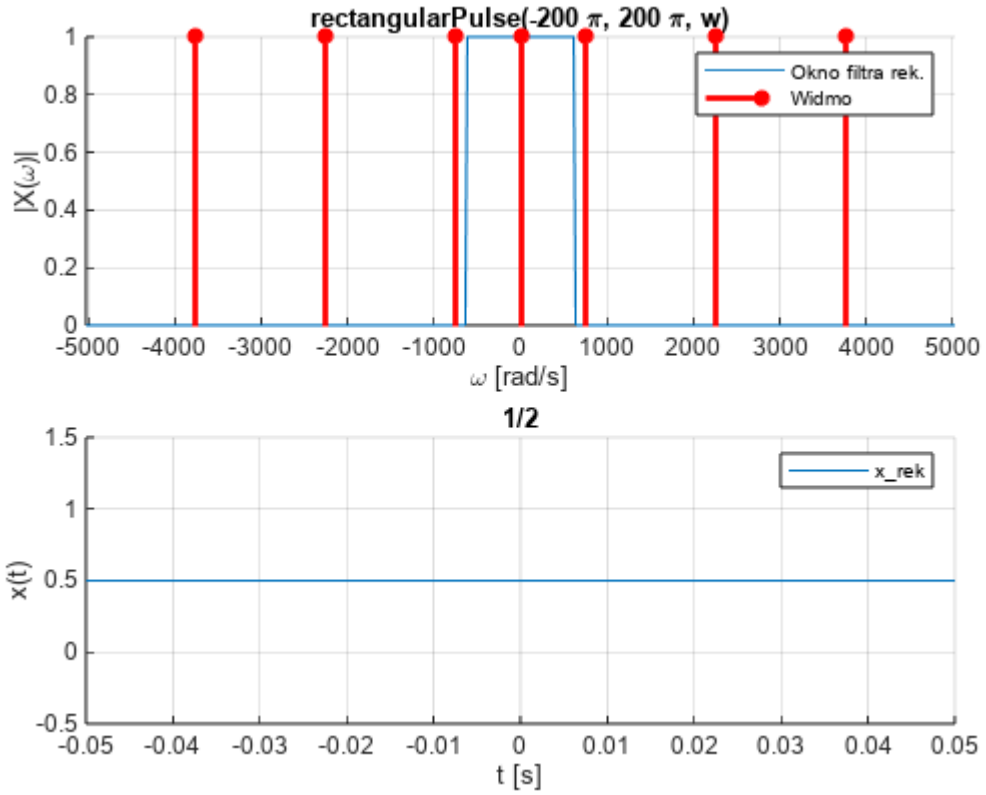
```
rectwave(1)
```





d)

```
rectwave(6/5)
```



Rekonstrukcja daje najlepsze efekty gdy okno filtra pokrywa wszystkie składowe częstotliwości sygnału. tj im s mniejsze tym rekonstrukcja lepsza

```
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 rekonstruuj¡cy
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 [] = trianwave(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 = sym(0);

X_FT_sin_org = triangularPulse(-ws, 0, ws, 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 = 2*rectangularPulse(-wg,wg,w); % filtr rekonstruuj¡cy
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('Rekonstukcja')
%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('x_rek');
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [] = rectwave(s)
syms t x w K
fp = 200; fg = fp/2; %Hz
wp = 2*pi*fp; wg = 2*pi*fg;
ws = s*wg;
Ts = 1/s/fg;
xTT = rectangularPulse(-Ts/4, Ts/4, t);

NT = 20;
ind = -NT : NT;
X = sym(0);
for n = ind
    Xn = (ws)*int(xTT*exp(-1i*ws*n*t),t, [-1/4*Ts, 3/4*Ts]);
    %Xn to współczynnik
    X = X + sym(Xn)*dirac(w-n*ws);
end

X_FT_sin = X;
x_sin = xTT;

FILT_FT = rectangularPulse(-wg,wg,w); % filtr rekonstruuj¡cy
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_rek, BND_t); % syg. próbkowany
xlabel('t [s]'); ylabel('x(t)')
legend('x\_rek');
end
```