

SPRAWOZDANIE 6

Zajęcia nr 7 - Próbkowanie i przekształcanie sygnałów ciągłych,

Michał Midor, gr. 5 - 19.11.2025, godz: 16:45

Wprowadzenie:

Celem laboratorium jest symulacja i analiza procesu cyfrowego przetwarzania sygnałów ciągłych, obejmująca trzy kluczowe etapy: próbkowanie (dyskretyzację), analizę widmową oraz rekonstrukcję sygnału ciągłego (konwersję C/A). Głównym zagadnieniem badanym w ćwiczeniu jest **Twierdzenie o próbkowaniu (Nyquista-Shannona)**.

Symulacje przeprowadzone w środowisku MATLAB mają na celu wizualizację zjawiska **aliasingu** (nakładania się widm), powstającego w wyniku niespełnienia warunku $f_p > 2f_{\max}$, oraz ocenę wpływu filtracji na wierność odtwarzania sygnałów o różnych charakterystykach widmowych.

Ćwiczenie 1) i 2)

W pierwszej części przeprowadzono symulację próbkowania sygnału sinusoidalnego o częstotliwości $f_s = 80$ Hz przy częstotliwości próbkowania $f_p = 200$ Hz. Ponieważ spełniony jest warunek Nyquista ($80 < 100$ Hz), otrzymano bezbłędną rekonstrukcję.

```
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)];

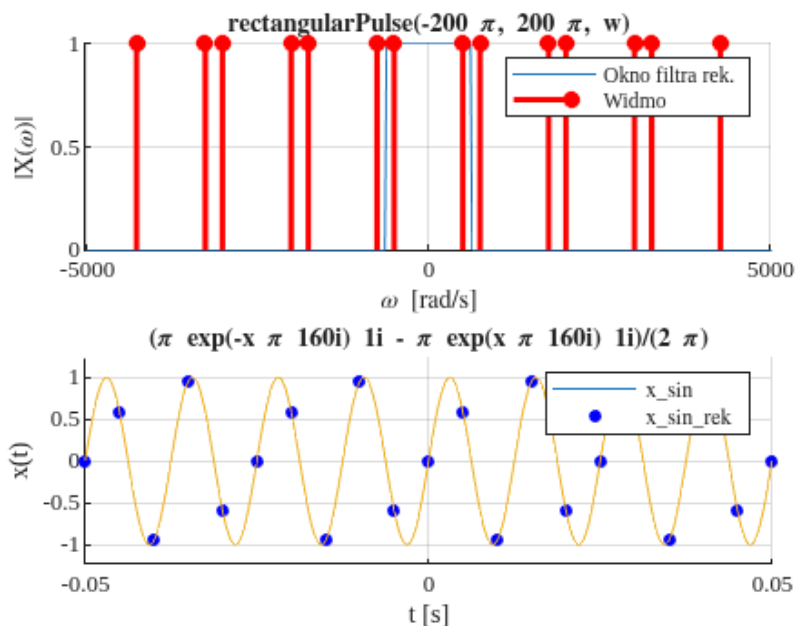
figure; subplot(2,1,1); hold on; grid on;
```

```

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
Tp = 1/fp;
t_p = BND_t(1):Tp:BND_t(2);
y_p = double(subs(x_sin, t, t_p));
plot(t_p, y_p, 'bo', 'MarkerFaceColor', 'b', 'MarkerSize', 4);
ezplot(x_sin_rek, BND_t) % syg. odtworzony
xlabel('t [s]'); ylabel('x(t)')
legend('x\_sin', 'x\_sin\_rek');

```



Naniesienie dyskretnych punktów pomiarowych w chwilach wielokrotności okresu próbkowania pozwala zaobserwować, że sygnał zrekonstruowany interpoluje, czyli łączy punkty próbek w sposób idealny.

Zadanie 3)

Zadanie polegało na przetestowaniu odporności systemu na łamanie twierdzenia o próbkowaniu. Zbadano szereg częstotliwości wejściowych, w tym przekraczających częstotliwość Nyquista ($f_g = 100$ Hz).

```

s_cases = [1/5, 6/5, 11/5, 16/5, 4/5, 9/5, 14/5];
for i = 1:length(s_cases)
    s = s_cases(i);
    ws = s*wg;

    % 1 - generacja sygnału

```

```

x_sin = sin(ws*t);

% 2 - transformata i modelowanie próbkowania (aliasy)
X_FT_sin_org = fourier(x_sin);
X_FT_sin = X_FT_sin_org + symsum((subs(X_FT_sin_org, w, w - K*wp) + ...
    subs(X_FT_sin_org, w, w + K*wp)), K, 1, 3);

% 3 - filtracja i rekonstrukcja
FILT_FT = rectangularPulse(-wg, wg, w);
x_sin_rek = ifourier(X_FT_sin * FILT_FT)

% 4 - ustawienie osi do wizualizacji
BND_t = [-10/fp, 10/fp]; % Zakres czasu
BND_w = [-4*wp, 4*wp]; % Zakres częstotliwości

w_SMP = [BND_w(1):wp/10:BND_w(2)];

% 5 - wizualizacja
figure();
% Dziedzina częstotliwości
subplot(2,1,1); hold on; grid on;
fplot(FILT_FT, BND_w, 'k--');
xlabel('\omega [rad/s]'); ylabel('|X(\omega)|');
legend('Filtr Rekonstrukcyjny', 'Prążki Widma (Aliaszy)');

% Dziedzina czasu
subplot(2,1,2); hold on; grid on;
fplot(x_sin, BND_t, 'b');
fplot(x_sin_rek, BND_t, 'r--'); % rekonstrukcja

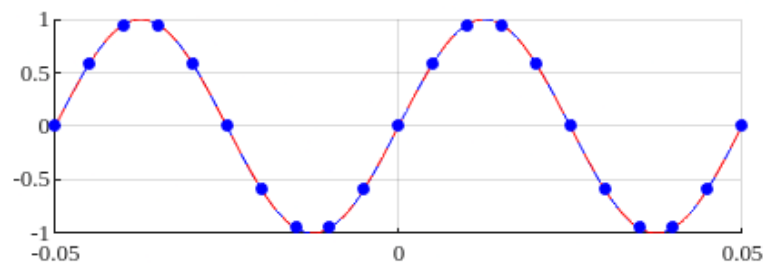
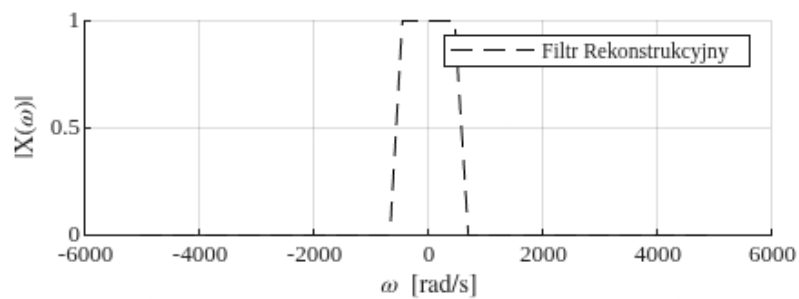
% Dodanie wizualizacji próbkowania
Tp = 1/fp;
t_p = BND_t(1):Tp:BND_t(2);
y_p = double(subs(x_sin, t, t_p));
plot(t_p, y_p, 'bo', 'MarkerFaceColor', 'b', 'MarkerSize', 4);

drawnow;
end

```

$$x_{sin_rek} = \frac{\pi e^{-40\pi xi} i - \pi e^{40\pi xi} i}{2\pi}$$

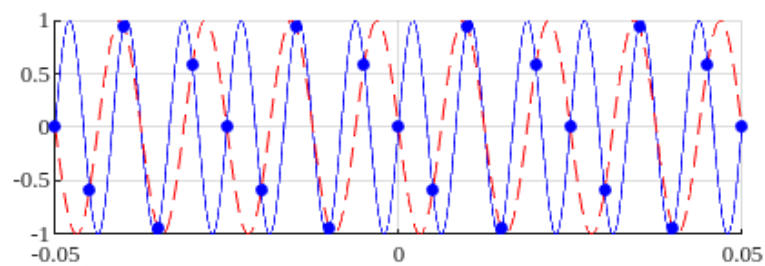
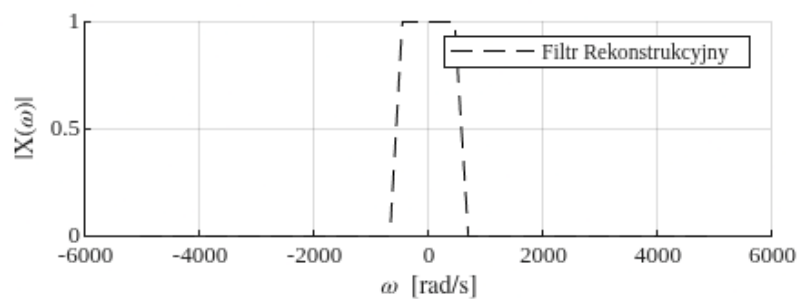
Warning: Ignoring extra legend entries.



x_sin_rek =

$$-\frac{\pi e^{-160\pi xi}i - \pi e^{160\pi xi}i}{2\pi}$$

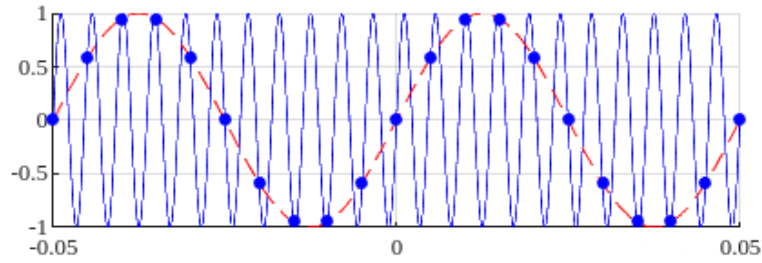
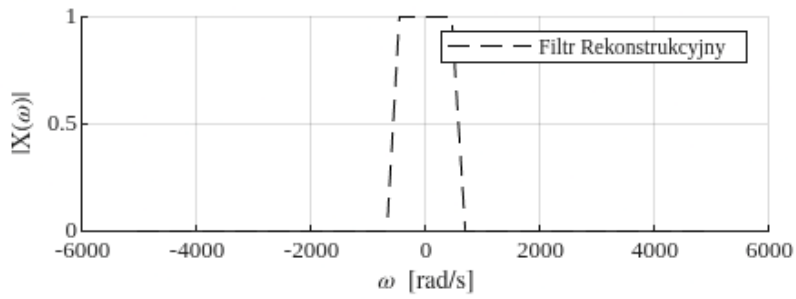
Warning: Ignoring extra legend entries.



x_sin_rek =

$$\frac{\pi e^{-40\pi xi}i - \pi e^{40\pi xi}i}{2\pi}$$

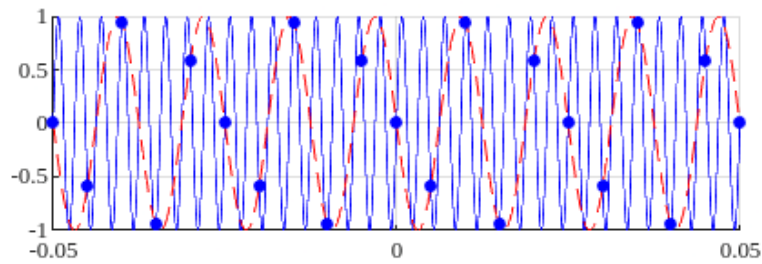
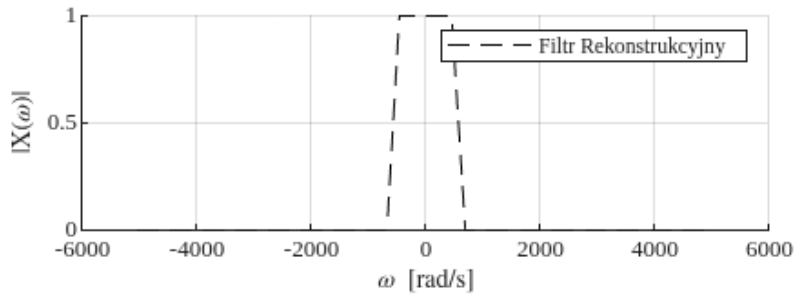
Warning: Ignoring extra legend entries.



x_sin_rek =

$$-\frac{\pi e^{-160\pi xi}i - \pi e^{160\pi xi}i}{2\pi}$$

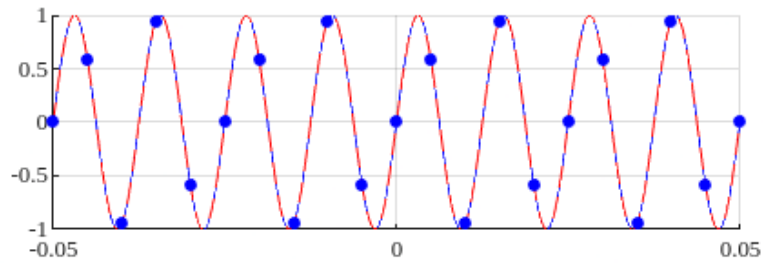
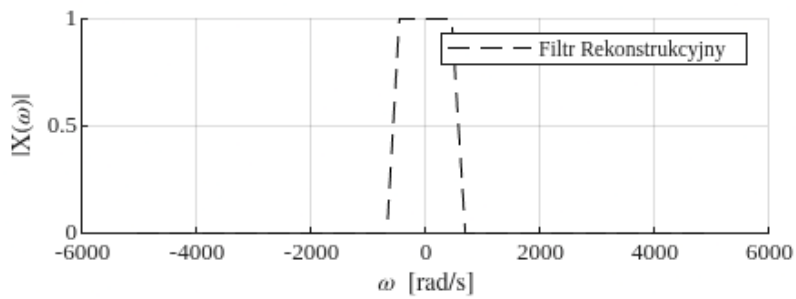
Warning: Ignoring extra legend entries.



x_sin_rek =

$$\frac{\pi e^{-160\pi xi}i - \pi e^{160\pi xi}i}{2\pi}$$

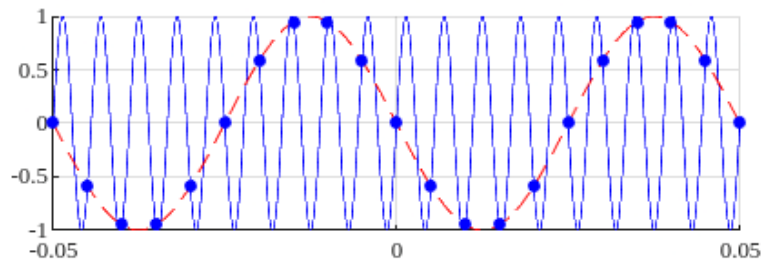
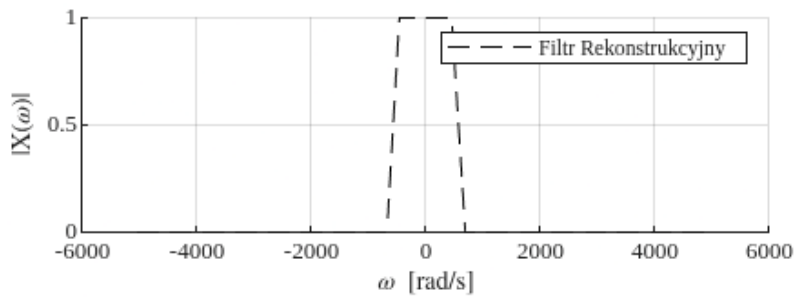
Warning: Ignoring extra legend entries.



x_sin_rek =

$$-\frac{\pi e^{-40\pi x i} i - \pi e^{40\pi x i} i}{2\pi}$$

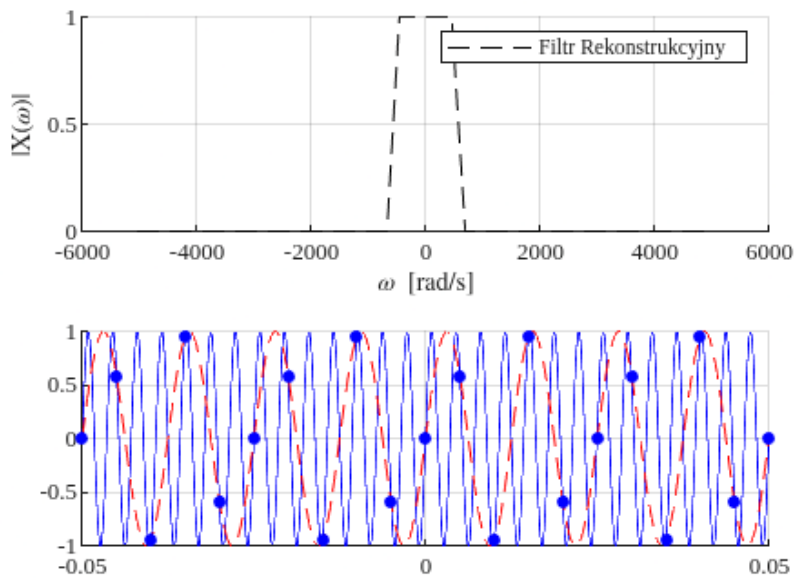
Warning: Ignoring extra legend entries.



x_sin_rek =

$$\frac{\pi e^{-160\pi x i} i - \pi e^{160\pi x i} i}{2\pi}$$

Warning: Ignoring extra legend entries.



Wniosek:

Filtr rekonstrukcyjny przepuszcza tylko składowe z zakresu $(-f_g, f_g)$.

Zadanie 4)

W tym zadaniu zamiast sygnału sinusoidalnego, zbadano sygnał o widmie ciągłym w kształcie trójkąta, rozciągającym się do granicy pasma Nyquista, by sprawdzić, jak zachowują się szerokie widma w procesie próbkowania.

```
figure();
X_Triangle_Org = triangularPulse(-wg, wg, w);

% Dodanie Aliasów
X_Triangle_Aliased = X_Triangle_Org + symsum(subs(X_Triangle_Org, w, w-K*wp)
+...
subs(X_Triangle_Org, w, w+K*wp),
K, 1, 3); % Podstawienie za K wartości od 1 do 3

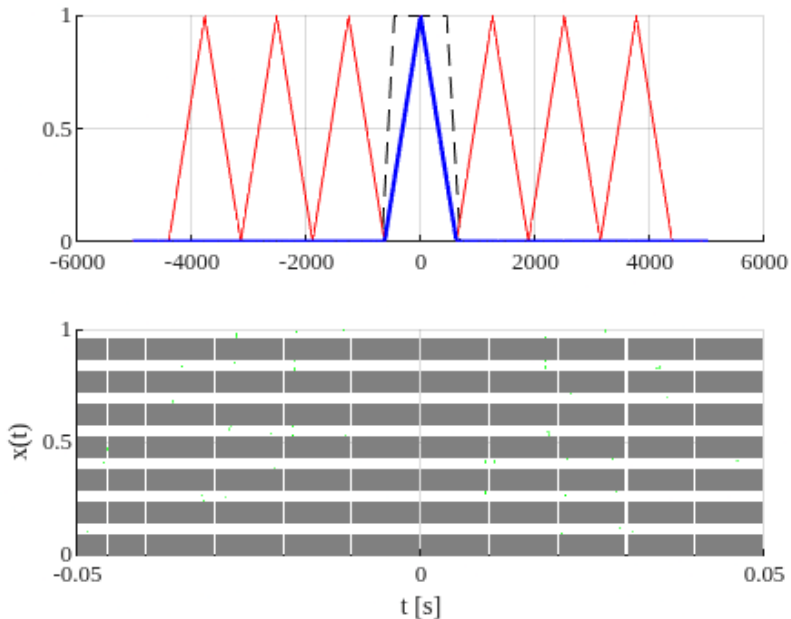
% Filtr rekonstrukcji
FILT_REC = rectangularPulse(-wg, wg, w);

% Rekonstrukcja w dziedzinie czasu

x_triangle_rek = ifourier(X_Triangle_Aliased*FILT_REC);

subplot(2,1,1); hold on; grid on;
fplot(FILT_REC, BND_w, 'k--');
fplot(X_Triangle_Aliased, BND_w, 'r');
fplot(X_Triangle_Org, BND_w, 'b', 'LineWidth', 1.5); % Rysowanie oryginału
jako punkt odniesienia
```

```
subplot(2,1,2); hold on; grid on;
x_triangle_theory = ifourier(X_Triangle_Org); % Teoretyczny sygnał po
rekonstrukcji, przejście z dziedziny częstotliwości do czasu
fplot(x_triangle_theory, BND_t, 'b');
fplot(x_triangle_rek, BND_t, 'g--'); % rekonstrukcja sygnału trójkątnego
xlabel('t [s]'); ylabel('x(t)');
```



Zadanie 5)

Ostatnie ćwiczenie symulowało próbkowanie fali prostokątnej, która teoretycznie posiada nieskończone widmo prążkowe (nieparzyste harmoniczne). Sygnał zdefiniowano jako sumę szeregu Fouriera (N=20 wyrazów).

```
cases_multipliers = [1/5, 4/5, 1, 6/5]; % Mnożniki dla a, b, c, d

% Zakres harmoniczných
N_harm = 20;
n_range = -N_harm:N_harm;

for i = 1:length(cases_multipliers)
    s = cases_multipliers(i);
    fs = s*fg;
    ws = 2*pi*fs;

    % Budowa widma fali prostokątnej

    X_Spectrum_Cont = 0;

    for n = n_range
        if n == 0
            Xn = 0.5; % Składowa stała (czyli wartość średnia)
```

```

elseif mod(n,2)==0
    Xn = 0; % Dla parzystych
else
    Xn = 1/(1j*n*pi);
end
X_Spectrum_Cont = X_Spectrum_Cont + 2*pi * Xn * dirac(w-n*ws);
end

% Symulacja próbkowania

X_Spectrum_Aliased = X_Spectrum_Cont;

for k = 1:3 % dodawanie aliasów w pętli dla stabilności
    X_Spectrum_Aliased = X_Spectrum_Aliased + ...
        subs(X_Spectrum_Cont, w, w - k*wp) + ...
        subs(X_Spectrum_Cont, w, w + k*wp);
end

% Filtracja i rekonstrukcja

FILT_REC = rectangularPulse(-wg, wg, w);

X_Final = X_Spectrum_Aliased*FILT_REC;

x_rek = ifourier(X_Final, w, t);

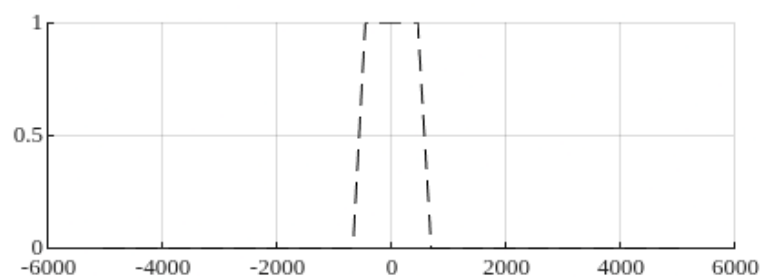
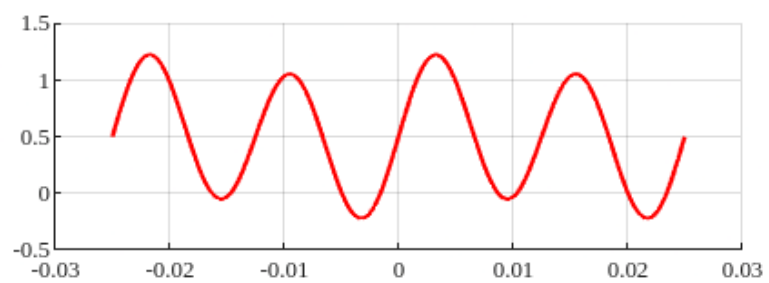
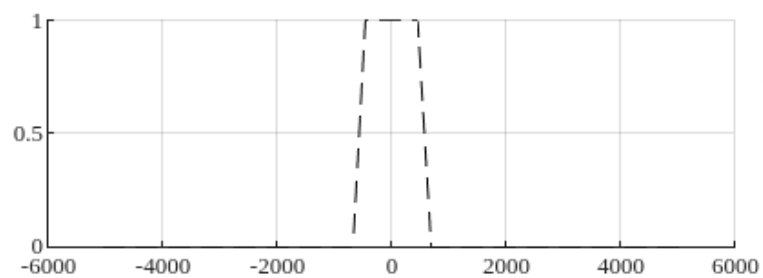
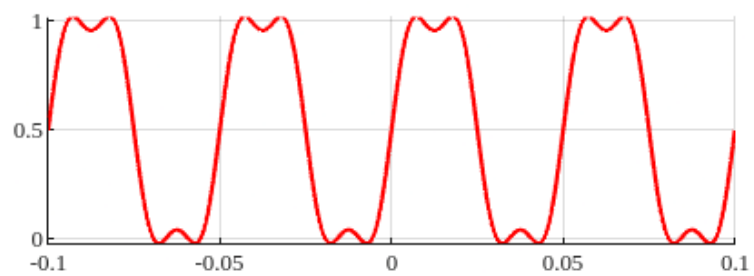
figure();
subplot(2,1,1); hold on; grid on;
T_sig = 1/fs; % okres sygnału i 4 okresy na wykresie
BND_t = [-2*T_sig, 2*T_sig];

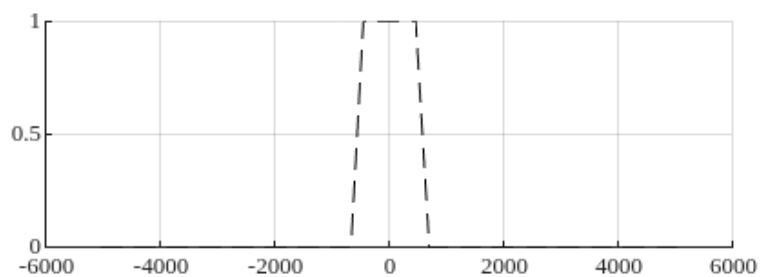
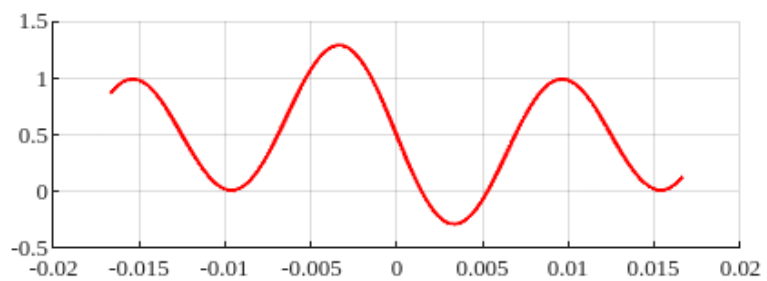
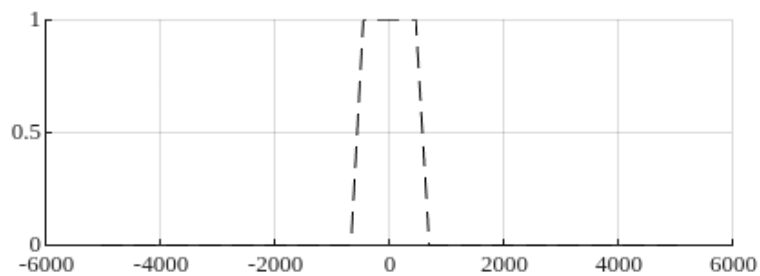
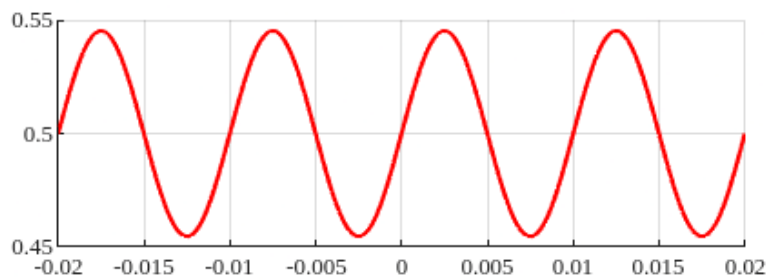
fplot(x_rek, BND_t, 'r', 'LineWidth', 1.5);
% Idealny prostokąt dla odniesienia
%t_num = linspace(BND_t(1), BND_t(2), 1000);
%x_ideal = 0.5 + 0.5*square(ws*t_num);
%plot(x_ideal, t_num, 'b--');

subplot(2,1,2); hold on; grid on;
BND_w = [-4*wp, 4*wp];
fplot(FILT_REC, BND_w, 'k--'); % Filtr

drawnow;
end

```





Filtr ucina wyższe harmoniczne, które wyjdą poza $f_{\text{graniczne}}$ (f_g), co spowoduje, że pozostanie jedynie "goła" sinusoida, a prostokąt zamieni się w sinus.

Wniosek:

Idealna rekonstrukcja sygnałów o ostrych zboczach (szerokopasmowych) wymaga nieskończenie wysokiej częstotliwości próbkowania lub zaakceptowania utraty szczegółów (wygładzenia sygnału)