Teoria i Przetwarzanie Sygnałów – Laboratorium komputerowe

2) Wybrane p arametry sygnałów i funkcja autokorelacji

1. Dla sygnałów: a) impuls prostokątny wsp. wypełnienia 50%, b) fragment sygnał sinusoidalnie zmiennego: $\sin(2\,\pi\, 0.1\, n)$, c) szum o rozkładzie gaussowskim (*randn*), wyznaczyć (korzystając z podanych poniżej definicji) następujące parametry sygnałów: wartość średnia, energia, moc średnia, wariancja, odchylenie standardowe. Uzyskane wartości porównać z wartościami zwracanymi przez funkcje Matlab'a. Przyjąć długość sygnału N=1000, indeks próbki sygnału n=0,1,...,N-1.

Uwaga: przydatne funkcje: *randn, std, var*

$$var(x) = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \bar{x})^2 , \quad \bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

$$std(x) = \sqrt{\mathrm{var}(x)}$$

$$E_x = \sum_{n=1}^{N}|x_n|^2 , \quad P_x = \frac{1}{N}\sum_{n=1}^{N}|x_n|^2$$

Kod:

```
clc
clear

N=1000;
n=0:N-1;
%funkcje
a=[ones(1,500),zeros(1,500)];%przebieg prostokatny

b= sin(2*pi*0.1*n);%przebieg sin

c=randn([1,1000]);%rozklad guassa

disp('Wartosc srednia przebiegu prostokatnego')
srednia_a=sum(a)/1000
mean(a)

disp('Wartosc energi przebiegu prostokatnego')
energia_a=sum(a.^2)

disp('Wartosc mocy sredniej  przebiegu prostokatnego')
moc_a=energia_a/N
disp('Wartosc wariancji przebiegu prostokatnego')
wariancja_a=sum((a-srednia_a).^2)/(N-1)
var(a)

disp('Wartosc odchylenia standardowego przebiegu prostokatnego')
odchylenie_a= sqrt(wariancja_a)
std(a)
%przebiegu sinusoidalny
disp('Wartosc srednia przebiegu sinusoidalnego ')
srednia_b=sum(b)/1000
mean(b)
disp('Wartosc energi przebiegu sinusoidalnego')
energia_b=sum(b.^2)

disp('Wartosc mocy sredniej  przebiegu sinusoidalnego')
moc_b=energia_b/N

disp('Wartosc wariancji przebiegu sinusoidalnego')
wariancja_b=sum((b-srednia_b).^2)/(N-1)
var(b)
```

```
disp('Wartosc odchylenia standardowego przebiegu sinusoidalnego')
odchylenie_b= sqrt(wariancja_b)
std(b)

%szum o rozkladzie gaussa
disp('Wartosc srednia szumu o rozkladzie gaussa ')
srednia_c=sum(c)/1000
mean(c)

disp('Wartosc energi szumu o rozkladzie gaussa')
energia_c=sum(c.^2)

disp('Wartosc mocy sredniej  szumu o rozkladzie gaussa')
moc_c=energia_c/N

disp('Wartosc wariancjiszumu o rozkladzie gaussa')
wariancja_c=sum((c-srednia_c).^2)/(N-1)
var(c)

disp('Wartosc odchylenia standardowego szumu o rozkladzie gaussa')
odchylenie_c= sqrt(wariancja_c)
std(c)
```

2. Dla następujących sygnałów dyskretnych:
   a) impuls (jak na rysunku) o czasie trwania N=10 próbek,
   b) x=[1,2,3,0,0]; y=[4,1,1,0,0],
   c) szumu o rozkładzie normalnym, N=100 próbek z generatora *randn*,
   d) $x_1(t) = \sin(2\pi 5t)$, $x_2(t) = \sin(2\pi 5t) + 0.5\sin(2\pi 10t) + 0.25\sin(2\pi 30t)$ dla $0 < t < 1$

[s], wyznaczyć funkcję autokorelacji (Rxx) zgodnie z poniższym wzorem

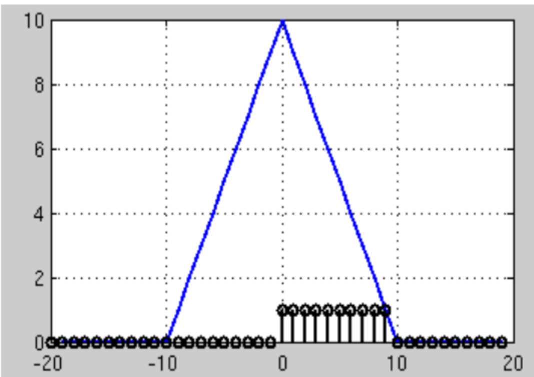
wersja „biased” $$R_{xy}(k) = \frac{1}{N}\sum_n x(n)\, y(n-k) \qquad \text{(a)}$$

wersja „unbiased” $$R_{xy}(k) = \frac{1}{N-|k|}\sum_n x(n)\, y(n-k) \qquad \text{(b)}$$

Uzyskany wynik porównać z funkcją *xcorr*.
Uwaga: do implementacji „własnej” w Matlabie bardziej dogodne są następujące wzory w zależności od znaku k (pominięto czynnik normujący przed znakiem sumy)

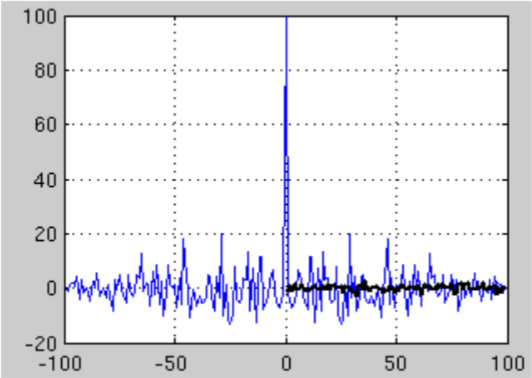
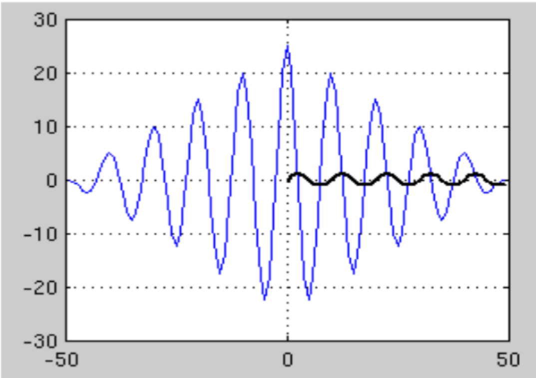
$$k \geq 0: \quad R_{xy}(k) = \sum_{n=k}^{N-1} x(n)\, y(n-k), \qquad k<0: \quad R_{xy}(k) = \sum_{n=0}^{N-1-|k|} x(n)\, y(n-k)$$

W powyższych wzorach nie uwzględniono czynników skalujących o których mowa w (a) i (b).
Uwaga: sygnał (d) jest sygnałem z czasem ciągłym, więc by wykonać obliczenia numeryczne musimy go spróbkować.
Wyniki dla przykładowych sygnałów tj. (a), (c) oraz sin(2*pi*0.1*n) pokazano poniżej.



numer próbki(n), przesunięcie (k)

Impuls prostokątny $\Pi(n)=1$, $n=0,1,\dots,9$ (czarny), oraz jego funkcja autokorelacji (kolor niebieski). Uwaga pokazana funkcja autokorelacji jest bez skalowania (tj. bez czynnika $1/N$). Oś X reprezentuje numer próbki, zaś os Y wartość sygnału lub odpowiednio jego funkcji autokorelacji.

 numer próbki(n), przesunięcie (k)	Szum (*N*=100 próbek) oraz jego funkcja autokorelacji (kolor niebieski). Uwaga pokazana funkcja autokorelacji jest bez skalowania (tj. bez czynnika 1/*N*). Oś X reprezentuje numer próbki, zaś os Y wartość sygnału lub odpowiednio jego funkcji autokorelacji.
 numer próbki(n), przesunięcie (k)	Sygnał sin(2*pi*0.1*[0:N-1]), N=50 oraz jego funkcja autokorelacji (kolor niebieski). Uwaga pokazana funkcja autokorelacji jest bez skalowania (tj. bez czynnika 1/*N*). Oś X reprezentuje numer próbki, zaś os Y wartość sygnału lub odpowiednio jego funkcji autokorelacji.

Kod:

```
clear
clc
close all

% a) impuls o czasie trwania n=10 probek
N=-20:20;
a=rectpuls(N-5,10);
[C_a,LAGS_a]=xcorr(a);
figure(1)
stem(N,a)
hold on
plot(LAGS_a,C_a)
axis([-20 20 0 10])

% b)
x=[1,2,3,0,0];
y=[4,1,1,0,0];
[C_b,LAGS_b]=xcorr(x,y);
figure(2)
stem(x)
hold on
stem(y)
hold on
plot(LAGS_b,C_b)
axis([-6 6 0 12])

% c) szum o rozkladzie normalnym N=100
N1=100;
szum = randn(1,100);
figure(3)
plot(szum)
hold on
```

```
[C_c,LAGS_c]=xcorr(szum);
plot(LAGS_c,C_c)

% d)
t=0:0.02:1;
x1 = sin(2*pi*5*t);
figure(4)
plot(x1)
hold on
[C_d1,LAGS_d1] = xcorr(x1);
plot(LAGS_d1,C_d1)
axis([-50 50 -25 27])

x2 = sin(2*pi*5*t) + 0.5*sin(2*pi*10*t) + 0.25*sin(2*pi*30*t);
figure(5)
plot(x2)
hold on
[C_d2,LAGS_d2] = xcorr(x2);
plot(LAGS_d2,C_d2)
axis([-50 50 -20 35])
```

Kod wersja druga

```
clc, close all, clear
%dane szumu
n=99;
N=0:1:n;
Y=randn(1,100);
Ra=xcorr(Y,Y);
Nra=-n:1:n;
% ---------autokorelacja "biased" DIY
z=[];
YY=fliplr(Y);
for i=1:length(Y)
    g=YY.*Y(i);
    z=[z;g];
end
[r k]=size(z);
t=r+k;
v=2;
A=[];
ii=0;
while(v<=t)
    for i=1:r
        for j=1:k
            if((i+j)==v)
               ii=ii+z(i,j);
            end
        end
    end
    v=v+1;
    A=[A ii];
    ii=0;
end
%jak sygnal jest krotkim wektorem lepiej stem()
plot(Nra,Ra,'g');hold;plot(N,Y,'r');grid
title('Autokorelacja (funkcja)')
legend('Autokorelacja','Zadany sygnał')
figure(2)
plot(Nra,A,'c');hold;plot(N,Y,'r');grid
title('Autokorelacja (własna)')
legend('Autokorelacja','Zadany sygnał')
```

```
e=sum(abs(Ra-A))
```

1) <u>Generacja sygnałów</u>

1. Napisać skrypt *imp_prost.m* generujący impuls prostokątny o czasie trwania *N*, przesunięciu *c*, szerokości *b*. Wielkości te wyrażone są liczbą próbek (przykładowo *N*=100, *c*=50, *b*=20).
Uwaga: Wykorzystać funkcje *zeros*, *ones*
Kod:

```
%zad 1
N=100;
c=50;
b=20;
x1=[zeros(1,c) ones(1,b) zeros(1,N-c-b)];
figure(1)
plot(x1,'.-r');
```

2. Napisać skrypt *sinus1.m* generujący sygnał sinusoidalnie zmienny o częstotliwości *f* [Hz], czasie trwania *Td* [s], częstotliwości próbkowania *fp* [Hz]. Na okoliczność testów przyjąć: *f*=10Hz, *fp*=100Hz, *Td*=1s.
Kod:

```
%zad 2

f2=10;
fp2=100;
Td=1;
t2=[0:1/fp2:Td];
x2=sin(2*pi*f2*t2);
figure(2)
plot(t2,x2,'.-b');
```

3. Napisać skrypt *sinus2.m* generujący sygnał sinusoidalnie zmienny o częstotliwości *f* [Hz], czasie trwania wyrażonym liczbą próbek *N*, częstotliwości próbkowania *fp* [Hz]. Na okoliczność testów przyjąć: *f*=10Hz, *fp*=100Hz, *N*=200.
Kod:

```
%zad 3
f3=10;
fp3=100;
N=200;
t3=[0:N-1]*(1/fp3);
x3=sin(2*pi*f3*t3);
figure(3)
plot(t3,x3,'.-b');
```

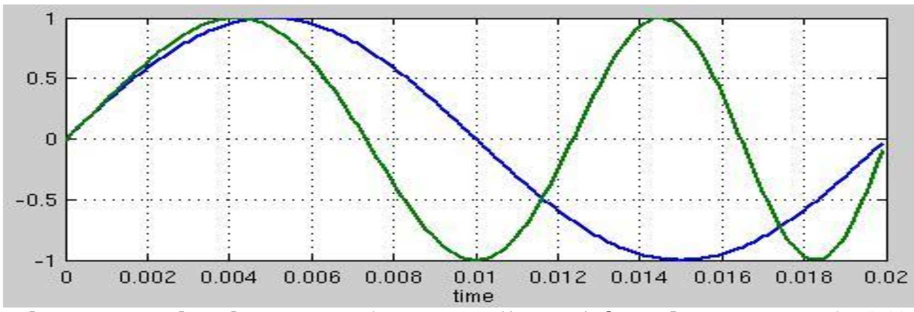
4. Częstotliwość chwilowa sygnału o liniowo narastającej częstotliwości ω_i od pewnej częstotliwości początkowej ω_0 z prędkością $k = \omega/\Delta T$ $[rad/s^2]$ określona jest:

$$\frac{d\phi}{dt} = \omega_i(t) = \omega_0 + kt$$

Wykazać, że sygnał dyskretny, którego częstotliwość chwilowa jest określona jak powyżej dany jest zależnością

$$x(n) = \sin\left(\omega_0 \frac{n}{f_p} + \frac{1}{2}k\left(\frac{n}{f_p}\right)^2 + \phi_0\right).$$

Następnie, napisać skrypt *sinus_lin_mod.m* generujący sygnał sinusoidalnie zmienny o <u>liniowo narastającej częstotliwości</u> z szybkością 50Hz/0.01s, liczba próbek *N* = 200. Częstotliwość początkowa f_0 = 50 Hz, częstotliwość próbkowania f_p = 10 kHz, faza początkowa ϕ_0=0 .
Uwaga: w podanym wzorze na *x*(*n*) współczynnik *k*, określający szybkość liniowego narastania częstotliwości, wyrażony jest w [rad/s^2], zaś w zadaniu w [1/s^2].



(krzywa: niebieska – sygnał o częstotliwości f_0, zielona – sygnał $x(n)$)

Kod:
```
%zad 4

N4=200;
f04=50
fp4=10000
fi04=0
w04=2*pi*f04;
k4=10000*pi
Td4=N4/fp4;
tw4=[0:1/fp4:Td4];
x4=sin(w04*tw4+0.5*k4*(tw4).^2+fi04);
y4=sin(w04*tw4);
figure(4)
hold on
grid on
plot(tw4,x4,'.-r',tw4,y4,'.-b');
```

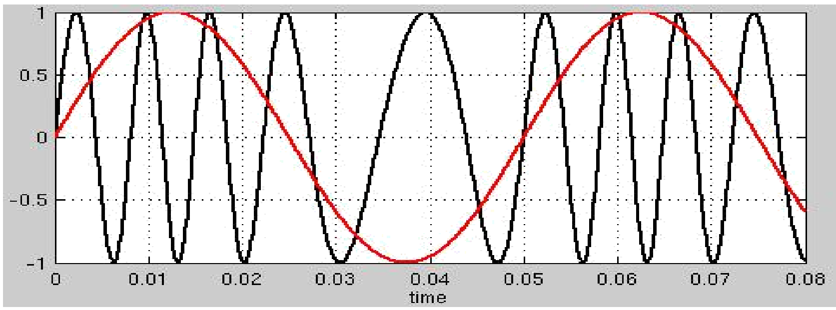
5. Częstotliwość chwilowa sygnału odstraja się od ω_0 o pewną wartość zwaną **dewiacją częstotliwości** $_\omega$ w sposób sinusoidalnie zmienny z częstotliwością ω_m:

$$\omega_i(t) = \omega_0 + \Delta_\omega \sin(\omega_m t)$$

Wykazać, że sygnał o tak zmieniającej się częstotliwości chwilowej dany jest wzorem:

$$x(t) = \sin(\phi(t)),\ \ \phi(t) = \omega_0 t - \omega^{\omega}{}_m \cos(\omega_m t) + \omega^{\omega}{}_m + \phi(0)$$

Napisać skrypt *sinus_sin_mod.m* generujący sygnał sinusoidalny z <u>sinusoidalnie modulowaną częstotliwoś cią</u> i liczbie próbek $N = 800$. Częstotliwość spoczynkowa $f_0 = 100$ Hz, dewiacja częstotliwości $\Delta_f = 50$ Hz, częstotliwość sygnału modulującego $f_m = 20$ Hz, częstotliwość próbkowania $f_p = 10000$ Hz. Uwaga: w powyższym: $\omega=2\pi f$.

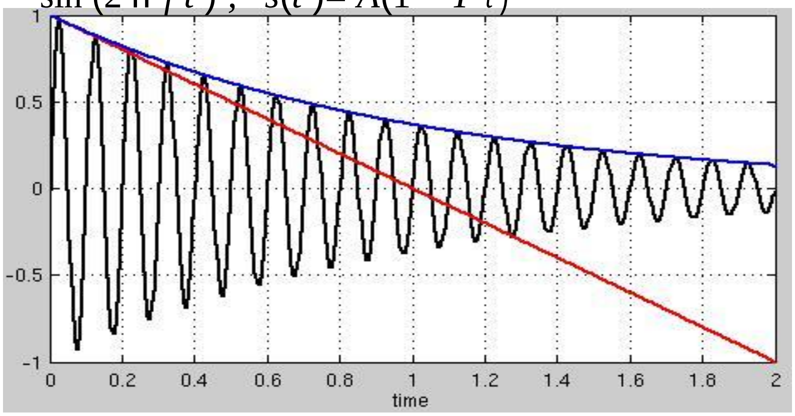


(krzywa: czarna – sygnał $x(n)$, czerwona – sygnał modulujący o częstotliwości f_m)

Kod:
```
N5=800;
f05=100;
delf5=50;
fm5=20;
fp5=10000;
fi05=0;
w05=2*pi*f05;
delw5=2*pi*delf5 ;
wm5=2*pi*fm5;
Td5=N5/fp5;
tw5=[0:1/fp5:Td5];
x5=sin(w05*tw5-(delw5/wm5)*cos(wm5*tw5)+delw5/wm5+fi05);
y5=sin(wm5*tw5);
figure(5)
hold on
grid on
plot(tw5,x5,'.-k',tw5,y5,'.-r');
```

6. Napisać skrypt *sinus_exp.m* generujący sygnał sinusoidalnie zmienny o częstotliwości f = 10Hz i wykładniczo malejącej amplitudzie, ze stałą czasową T = 1s, czas trwania wygenerowanego sygnału Td=2 sekundy. Narysować styczną do obwiedni sygnału w punkcie t=0. Wykorzystać poniższe równania, dobrać częstotliwość próbkowania.

$$x(t) = A e^{-t/T} \sin(2\pi f t), \quad s(t) = A\left(1 - \frac{1}{T} t\right)$$



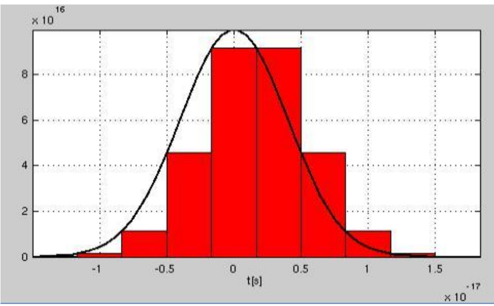
(krzywa: czarna – sygnał $x(n)$, niebieska – obwiednia, czerwona – styczna)

Kod:
```
f=10;
fp=500;
T=1;
Td=2;

t=(0:1/fp:Td-1/fp);

y=exp(-t/T) .*sin(2*pi*f*t);
s=1-t/T;
figure(6)
plot(t,y)
hold on
```

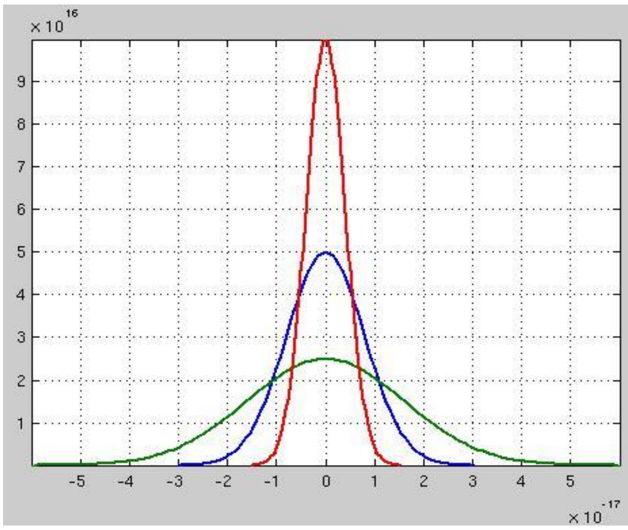
```
plot(t,s)
hold on
plot(t,exp(-t))
```

7. Napisać skrypt *ddirac.m* rysujący przybliżenie impulsu Diraca (poprzez funkcję wykładniczą), wyznaczyć pole powierzchni pod wykresem impulsu. W zadaniu należy wykorzystać funkcję $\delta(t,\tau)$, pokazaną poniżej. Przy dyskretyzacji funkcji $\delta(t,\tau)$ przyjąć N=1000 jako liczbę próbek oraz $\tau=10^{-17}$. Dobrać krok dyskretyzacji osi czasu, tak by na wykresie uwidocznić fragment funkcji odpowiadający odcinkowi czasu t o długości ok $5\times\tau$ (patrz rysunek poniżej).

$$\delta(t,\tau)=\frac{1}{\tau}e^{-\pi t^2/\tau^2}$$



Ilustracja zasady całkowania numerycznego metodą prostokątów.



Funkcja aproksymująca $\delta(t,\tau)$ dla $\tau=4\times10^{-17}$ (zielona), $\tau=2\times10^{-17}$ (niebieska) oraz $\tau=10^{-17}$ (czerwona)

Kod:

```
%Zad. 1.7
N = 1000;
tau = 10^-17;
tau2 = 2*10^-17;
tau3 = 4*10^-17;

t = -5/2*tau:5*tau/(N-1):5/2*tau;
di = exp(-pi*t.^2/tau^2)/tau;
di2 = exp(-pi*t.^2/tau2^2)/tau2;
di3 = exp(-pi*t.^2/tau3^2)/tau3;
plot(t,di,'.-k',t,di2,'-b',t,di3,'-r')

P_di = sum(di)*(t(2)-t(1))
P_di2 = sum(di2)*(t(2)-t(1))
P_di3 = sum(di3)*(t(2)-t(1))
```

Teoria i Przetwarzanie Sygnałów – Laboratorium komputerowe

3) Szereg Fouriera

1. Narysować kilka funkcji z podanych zbiorów i wyznaczyć ich iloczyny skalarne

$$\text{Zbiór 1.}\quad \left\{\frac{1}{\sqrt{T}},\ \sqrt{\frac{2}{T}}\cos\frac{2\pi}{T}nt,\ \sqrt{\frac{2}{T}}\sin\frac{2\pi}{T}nt\ :n=1,2,\ldots\right\}$$

$$\text{Zbiór 2.}\quad \left\{\frac{1}{\sqrt{T}}e^{j\frac{2\pi}{T}nt}\ :n=0,\pm1,\pm2,\ldots\right\}$$

$$\text{Zbiór 1.}\quad \left\{\frac{1}{\sqrt{T}},\ \sqrt{\frac{2}{T}}\cos\frac{2\pi}{T}nt,\ \sqrt{\frac{2}{T}}\sin\frac{2\pi}{T}nt\ :n=1,2,\ldots\right\}$$

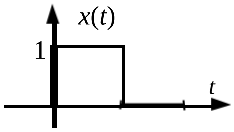
$$\text{Zbiór 2.}\quad \left\{\frac{1}{\sqrt{T}}e^{j\frac{2\pi}{T}nt}\ :n=0,\pm1,\pm2,\ldots\right\}$$

Kod:

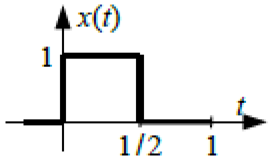
```
clear; clc;

%rysunek kilku funkcji i ich iloczyn skalarny
T = 1; n = 1; t = 0:0.001:T;
f1 = 1/sqrt(T);

figure(1);
subplot(5,1,1);
f2a = sqrt(2/T)*cos(2*pi*t/T); plot(f2a); grid on; axis([0 1000 -2 2]);
subplot(5,1,2);
f2b = sqrt(2/T)*cos(2*pi*2*t/T); plot(f2b); grid on; axis([0 1000 -2 2]);
subplot(5,1,3);
f2c = sqrt(2/T)*cos(2*pi*3*t/T); plot(f2c); grid on; axis([0 1000 -2 2]);
subplot(5,1,4);
f2d = sqrt(2/T)*cos(2*pi*4*t/T); plot(f2d); grid on; axis([0 1000 -2 2]);
subplot(5,1,5);
f2e = sqrt(2/T)*cos(2*pi*5*t/T); plot(f2e); grid on; axis([0 1000 -2 2]);

figure(2);
subplot(5,1,1);
f3a = sqrt(2/T)*sin(2*pi*t/T); plot(f3a); grid on; axis([0 1000 -2 2]);
subplot(5,1,2);
f3b = sqrt(2/T)*sin(2*pi*2*t/T); plot(f3b); grid on; axis([0 1000 -2 2]);
subplot(5,1,3);
f3c = sqrt(2/T)*sin(2*pi*3*t/T); plot(f3c); grid on; axis([0 1000 -2 2]);
subplot(5,1,4);
f3d = sqrt(2/T)*sin(2*pi*4*t/T); plot(f3d); grid on; axis([0 1000 -2 2]);
subplot(5,1,5);
f3e = sqrt(2/T)*sin(2*pi*5*t/T); plot(f3e); grid on; axis([0 1000 -2 2]);

C1 = dot(f2a,f3a)
C2 = dot(f2b,f2e)
```

2. Dany jest szereg Fouriera reprezentujący sygnał ciągu impulsów prostokątnych (patrz rys):

$$x(t) \approx \sum_{n=-N}^{N} X_n e^{j\frac{2\pi}{T}nt}, \qquad X_n = \begin{cases} 1/2, & n=0 \\ \dfrac{(-1)^n - 1}{-j2\pi n}, & n \neq 0 \end{cases},$$



$$x(t) \approx \sum_{n=-N}^{N} X_n e^{j\frac{2\pi}{T}nt}, \qquad X_n = \begin{cases} 1/2, & n=0 \\ \dfrac{(-1)^n - 1}{-j2\pi n}, & n \neq 0 \end{cases},$$



Napisać program, który wyznaczy sygnał $x(t)$ dla kilku wybranych wartości N (patrz przykładowe rysunki). Sygnał wyznaczyć dla t z następujących przedziałów: t=[0; T], t=[-T; T], t=[-T/2; T]. Zaobserwować własność okresowości rekonstruowanego sygnału, a także efekt Gibbsa. Sprawdzić jak zachowuje się błąd aproksymacji w funkcji N określony poniższym wzorem:

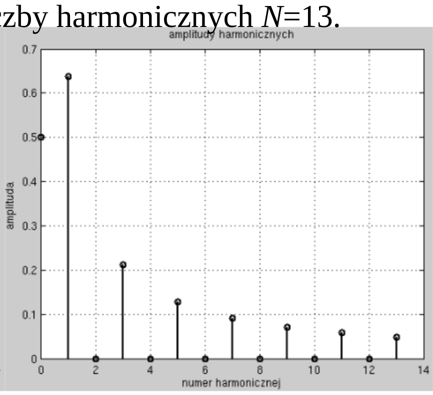
$$e_N^2 = \int_0^T \left| x(t) - \sum_{n=-N}^{N} X_n e^{j\omega_0 nt} \right|^2 dt = \int_0^T |x(t)|^2 dt - T \sum_{n=-N}^{N} |X_n|^2$$

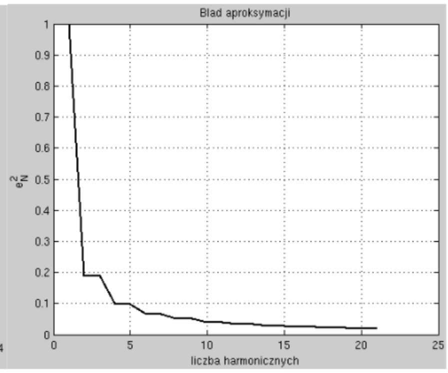
Uwaga. Ponieważ $X_n = -X_{-n}$ powyższy szereg można również zapisać w postaci

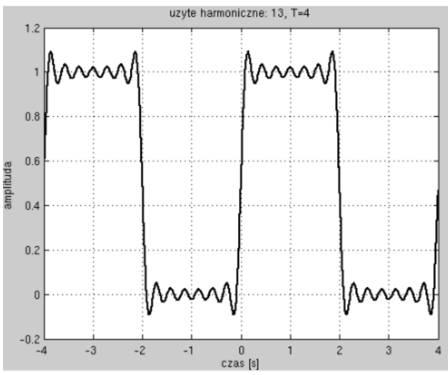
$$x(t) \approx \sum_{n=-N}^{N} X_n e^{j\frac{2\pi}{T}nt} = X_0 + \sum_{n=1}^{N} X_n \left(e^{j\frac{2\pi}{T}nt} - e^{-j\frac{2\pi}{T}nt}\right) = X_0 + \sum_{n=1}^{N} 2jX_n \sin\left(\frac{2\pi}{T}nt\right), \text{ gdzie}$$

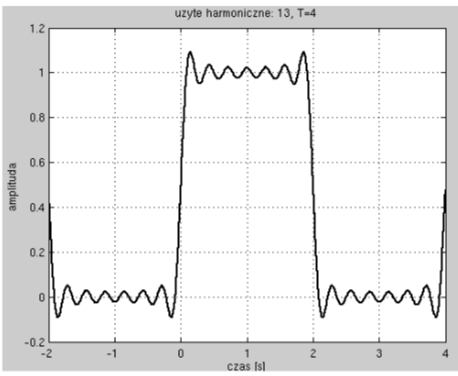
X_0 jest składową stałą sygnału, zaś $2jX_n$ jest amplitudą n-tej harmonicznej.

Przykładowe rysunki dla T=4 i liczby harmonicznych N=13.

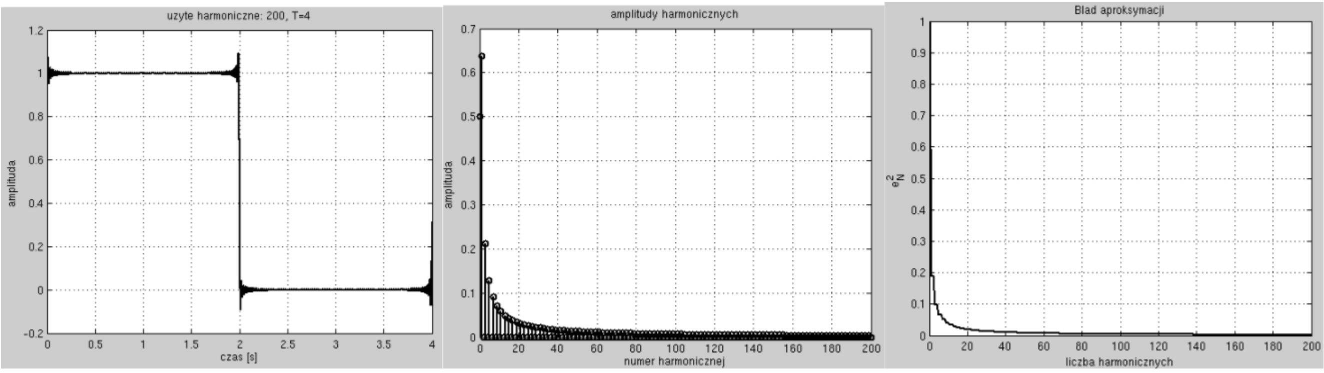








Przykładowe rysunki dla T=4 i liczby harmonicznych N=200.

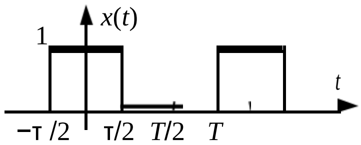


Kod:

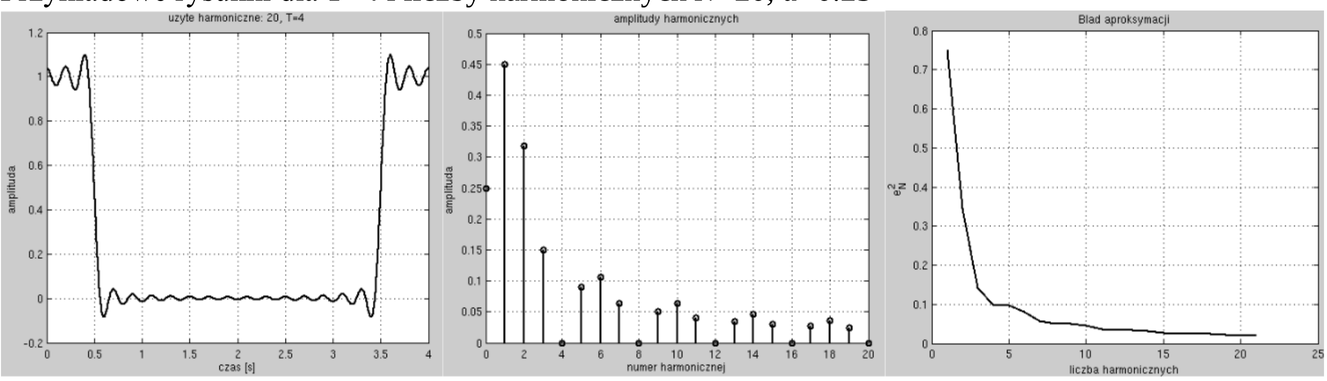
```
T=4;
N=13;
n=[1:N];
dt=0.001;
t=[0:dt:T];
X0=0.5;
Xn=((-1).^n'-1)./(-j*2*pi*n')
xt=X0+sum(2*j*Xn.*sin(2*pi.*n'.*t/T));
figure(1); hold on; grid on; %uĹĽyte harmoniczne z wzoru x(t)
plot(t,xt)
figure(2); hold on; grid on; title('Amplitudy harmonicznych'); %amplitudy
harmonicznych
stem(n,2*abs(Xn),'b')
stem(0,X0,'b')

for m=1:N
    E(m)=dt*sum(abs(xt).^2) - T*sum(T*abs(Xn(1:m)).^2);
end
figure(3); hold on; grid on; title('Blad aproksymacji') %bĹ‚Ä…d aproksymacji
plot(n,abs(E))
```

3. Korzystając ze wzoru (a) wyznaczyć współczynniki zespolonego szeregu Fouriera (b) reprezentujący niżej przedstawiony sygnał prostokątny o współczynniku wypełnienia $d = \tau/T$. Napisać program wyznaczający (rekonstruujący) przebieg czasowy sygnału $x(t)$ dla t=[0; T], t=[-T; T], t=[-T/2; T]. Przyjąć współczynnik wypełnienia (d) równy 25%. Zaobserwować zachowanie się modułów współczynników Fouriera (a), zwłaszcza tych związanych z wyższymi harmonicznymi, w funkcji zmiany współczynnika wypełnienia (d).

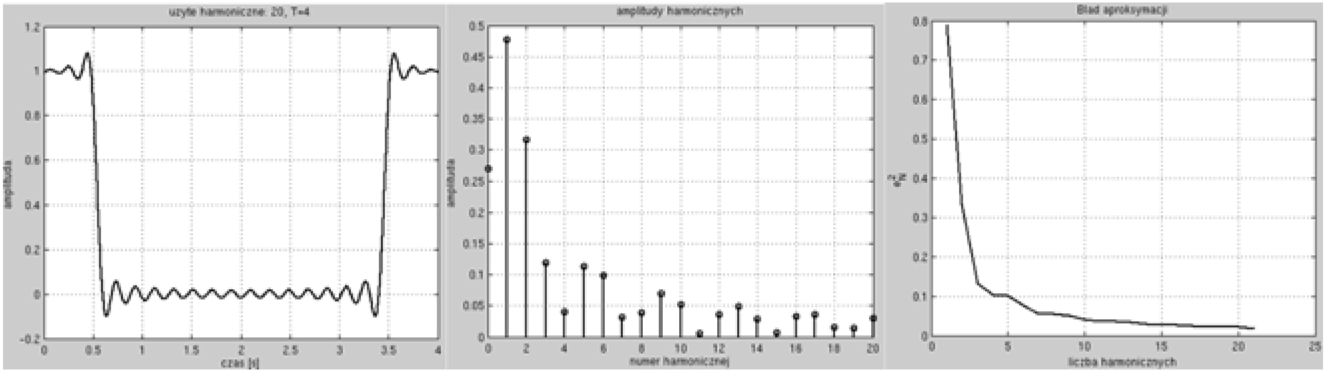
$$X_n = \frac{1}{T}\int_{-\tau/2}^{\tau/2} 1\, e^{-j\omega_0 nt}\,dt \quad \text{(a)} \qquad x(t) \approx \sum_{n=-N}^{N} X_n e^{\frac{j2\pi nt}{T}}, \quad \text{(b)}$$



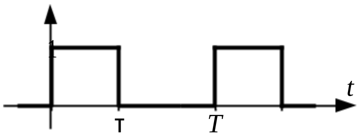
Przykładowe rysunki dla T=4 i liczby harmonicznych N=20, d=0.25



Przykładowe rysunki dla T=4 i liczby harmonicznych N=20, d=0.27



4. Sprawdzić (analitycznie), że niżej podane szeregi Fouriera (a) i (b) odpowiadają przebiegowi prostokątnemu pokazanemu na poniższym rysunku:



$$\tau / T = d \,,$$

Napisać program, który wyznaczy sygnał $x(t)$ dla $d = 0.25$ oraz kilku wybranych wartości N. Sprawdzić nierówność Bessela dla obu szeregów (a) i (b).

$$x(t) = d + \frac{1}{\pi}\sum_{n=1}^{N} \frac{2}{n}\sin(\omega_0 n \tau/2)\cos(\omega_0 n t - \omega_0 n \tau/2) \qquad \text{(a)}$$

$$x(t) = d + \frac{1}{\pi}\sum_{n=1}^{N} \frac{1}{n}[\sin \omega_0 n \tau \cos \omega_0 n t - [\cos \omega_0 n \tau - 1]\sin \omega_0 n t]\,. \qquad \text{(b)}$$

Dla przypomnienia niżej podano nierówność Bessela wraz z odpowiadającą jej szeregiem Fouriera.

$$\int_a^b |x(t)|^2\, dx \ge \sum_{k=0}^{\infty} |a_k|^2 \, \| \varphi_k(t) \|^2 \,, \quad x(t) = \sum_{k=0}^{\infty} a_k \varphi_k(t) \qquad \text{(c)}$$

5. Wyznaczyć współczynniki zespolonego szeregu Fouriera reprezentującego sygnał trójkątny o zerowej składowej stałej. W zadaniu należy skorzystać z wyników zadania 2 (dla przebiegu prostokątnego z zerową składową DC, tj. $X_0 = 0$) oraz faktu, że sygnał trójkątny jest całką sygnału prostokątnego.

Rozwinięcie w szereg Fouriera sygnału prostokątnego bez składowej stałej, o wypełnieniu 50 %, którego jeden okres:

$$x^{\sqcap}(t) = \begin{cases} 1/2, & 0 < t < T/2 \\ -1/2, & T/2 < t < T \end{cases}$$

dane jest wzorem:

$$x^{\sqcap}(t) = \sum_n X_n^{\sqcap} e^{j\omega_0 n t} \,, \quad \omega_0 = \frac{2\pi}{T} \,, \quad X_n^{\sqcap} = \begin{cases} 0, & n = 0 \\ \dfrac{(-1)^n - 1}{-j 2\pi n}, & n \neq 0 \end{cases}$$

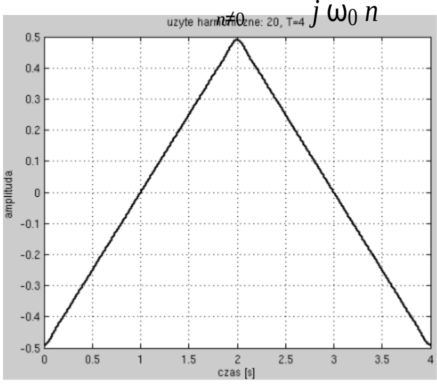
Całkując $x^{\sqcap}(t)$ w przedziale od 0 do $t < T$ otrzymujemy:

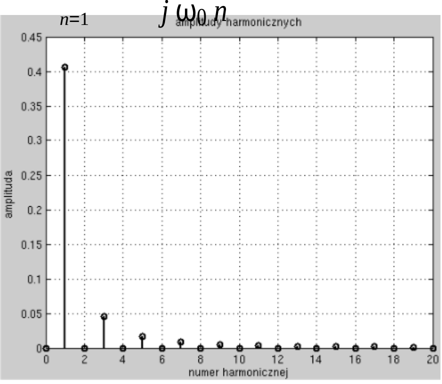
$$x(t) = \int_0^t x^{\sqcap}(\tau) d\tau = \int_0^t \sum_{n\neq 0} X_n^{\sqcap} e^{j\omega_0 n \tau} d\tau = \sum_{n \neq 0} X_n^{\sqcap} \frac{1}{j\omega_0 n}(e^{j\omega_0 n t} - 1)$$

$$x(t) = -\sum_{n\neq 0} X_n^{\sqcap} \frac{1}{j\omega_0 n} + \sum_{n \neq 0} X_n^{\sqcap} \frac{1}{j\omega_0 n} e^{j\omega_0 n t}$$

Pierwszy czynnik w powyższym równaniu stanowi składową stałą. Pomijając ją otrzymujemy sygnał trójkątny o zerowej składowej stałej:

$$x_0(t) = \sum_{n\neq 0} X_n^{\sqcap} \frac{1}{j\omega_0 n} e^{j\omega_0 n t} = \sum_{n=1}^{\infty} X_n^{\sqcap} \frac{1}{j\omega_0 n} 2\cos(\omega_0 n t)\,.$$





Wyjaśnić dlaczego w powyższym szeregu pojawiły się funkcje kosinus, a dlaczego w szeregu reprezentującym przebieg prostokątny $x^{\Pi}(t)$ są funkcje sinus.

Kod:

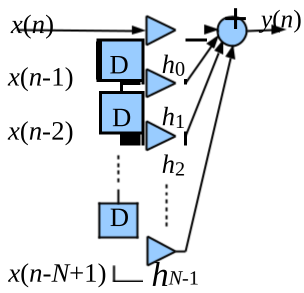
```
clc;
clear all;
close all;
%zadanie 5
T=4;
N=13;
n=[1:N];
dt=0.001;
t=[0:dt:T];
X0=0.5;
Xn=((-1).^n'-1)./(-j*2*pi*n')
w0=2*pi/T;
xtrojkat=sum(Xn.*2.*cos(w0*n'.*t)./(j*w0*n')); %trĂłjkÄ…t o zerowej skĹ‚adowej
staĹ‚ej
figure(4); grid on; hold on;
plot(t,xtrojkat)
figure(5);
A5=abs(Xn.*2.*cos(w0*n'.*t)./(j*w0*n'));  %amplitudy harmonicznych z x(t) na
trojkat
stem(n,A5(:,1)')
grid on; title('Amplitudy harmonicznych');
```

Laboratorium

<u>Układy dyskretne LTI – projektowanie filtrów typu FIR</u>

Z1. Napisać funkcję *y* = filtruj(*x*, *h*), która wyznacza sygnał *y* będący wynikiem filtracji sygnału *x* przez filtr FIR o odpowiedzi impulsowej *h*. Implementację należy wykonać w oparciu o schemat blokowy pokazany poniżej.

$$y(n)=\sum_{k=0}^{N-1} h(k)\,x(n-k),$$



Uzyskany wynik porównać z funkcją *filter* z Matlaba. Do testowania testowania poprawności działania użyć sygnałów *x* = [1 2 3], *h* = [1 1 1 1]: *filter*(*h*, 1, *x*).

Kod:

```
clc;clear all;close all;
x=[1 2 3];
h=[1 1 1 1];

N=length(x);

y2=[];
 for k=0:N-1;
     n=k:N-1;

     y(n+1)=sum(h(1+k).*x(n-k+1));
     y=circshift(y,1);
    %PT: jest OK, ale prosze rozwazyc inna konstrukcje do przesuwnania
bufora w dol
 end

y
fil=filter(h,1,x)
```

Z2. Napisać funkcję (ewentualnie skrypt) wyznaczającą próbki odpowiedzi impulsowej filtru dolnoprzepustowego o częstotliwości granicznej g = /4 i górnoprzepustowego o g = /8. Odpowiedź impulsowa idealnego (nie obciętego) filtru o charakterystyce częstotliwościowej $H(e^{j\omega})$ dana jest wzorem:

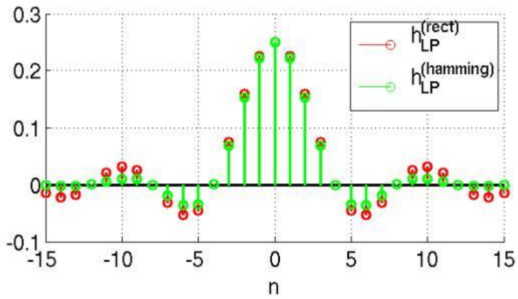
$$h(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\omega})e^{j\omega n}\,d\omega\,, \quad -\infty<n<\infty\,.$$

Odpowiedź impulsowa dolnoprzepustowego filtru typu FIR jest równa:

$$h_{LP}(n)=\begin{cases}\frac{\omega_g}{\pi}\,, & n=0\\ \frac{\sin(\omega_g n)}{\pi n}\,, & n\neq 0\end{cases}\,,$$

Po obcięciu oknem $w(n)$ mamy:

$$h^{W}_{LP}(n)=w(n)h_{LP}(n),\quad -N\leq n\leq N\,.$$



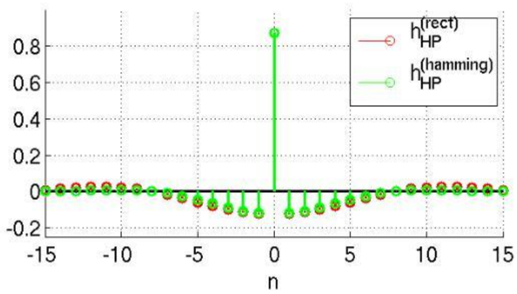
Rys. Odpowiedzi impulsowe filtrów z oknem prostokątnym i oknem Hamminga, N=15, g = /4.

Odpowiedź impulsowa górnoprzepustowego filtru typu FIR z oknem $w(n)$ jest równa:

$$h_{HP}(n)=\begin{cases}1-\frac{\omega_g}{\pi}\,, & n=0\\ -\sin(\omega_g n)\,, & n\neq 0\end{cases}\,,$$

Po obcięciu oknem $w(n)$ mamy:

$$h^{W}_{HP}(n)=w(n)h_{HP}(n),\quad -N\leq n\leq N\,.$$



Rys. Odpowiedzi impulsowe filtrów z oknem prostokątnym i oknem Hamminga, N=15, g = /8.

Uwaga: funkcja okna czasowego $w(n)$ może być uzyskana w wyniku wywołania funkcji: *hamming* lub *blackman* pakietu Matlaba. Odpowiedzi impulsowe narysowano funkcją *stem*.

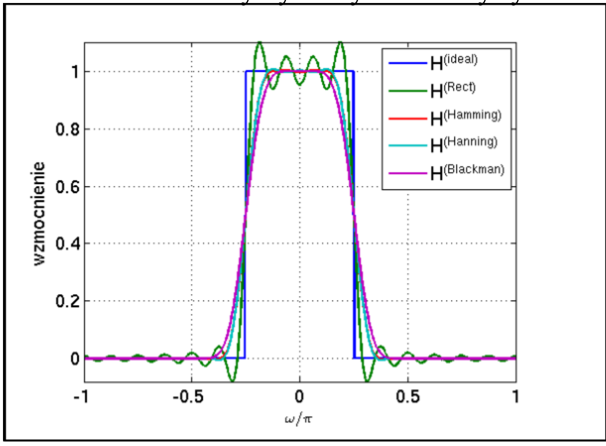
Kod:

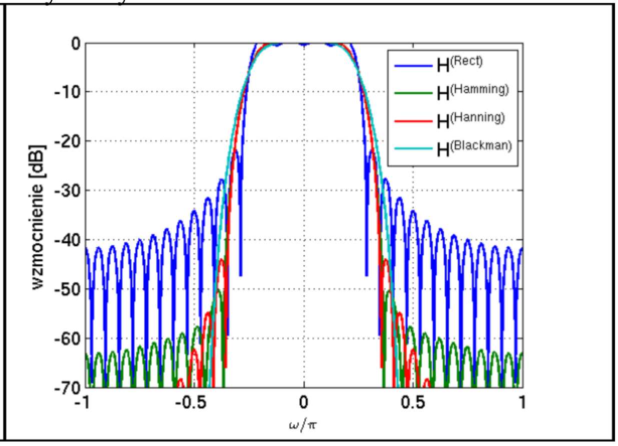
```
%% filtr dolnoprzepustowy
N=15;
n1=-N:N;

wgr=pi()/4;

%ujemna czeĹ›Ä‡
n=-N:(-1);
h_lp_u=sin(wgr*n)./(pi()*n);
%zero
h_lp_0=wgr/pi();
%czÄ™Ĺ›c+
n=1:N;
h_lp_d=sin(wgr*n)./(pi()*n);

hlp=[h_lp_u,h_lp_0,h_lp_d];
stem(n1,hlp,'r');grid on;hold on


win=hamming(size(n1,2));
hlp_ham=hlp'.*win;;
```

```
stem(n1,hlp_ham,'g');
grid on;
hold on;
legend('h_{LP}^{(rect)}','h_{LP}^{(hamming)}')



%% filtr gĂłrnoprzepustowy
N=15;
n2=-N:N;
wgr_g=pi()/8;

%ujemna czeĹ›Ä‡
n=-N:(-1);
h_hp_u=(-sin(wgr_g*n))./(pi*n);
%zero
h_hp_0=1-(wgr_g/pi);
%czÄ™Ĺ›c+
n=1:N;
h_hp_d=(-sin(wgr_g*n))./(pi*n);

hhp=[h_hp_u,h_hp_0,h_hp_d];
figure(2)
stem(n2,hhp,'r');
grid on;
hold on;
win2=hamming(size(n2,2));
hhp_ham=hhp'.*win2;
stem(n2,hhp_ham,'g');
grid on;
hold on;
legend('h_{HP}^{(rect)}','h_{HP}^{(hamming)}')
```

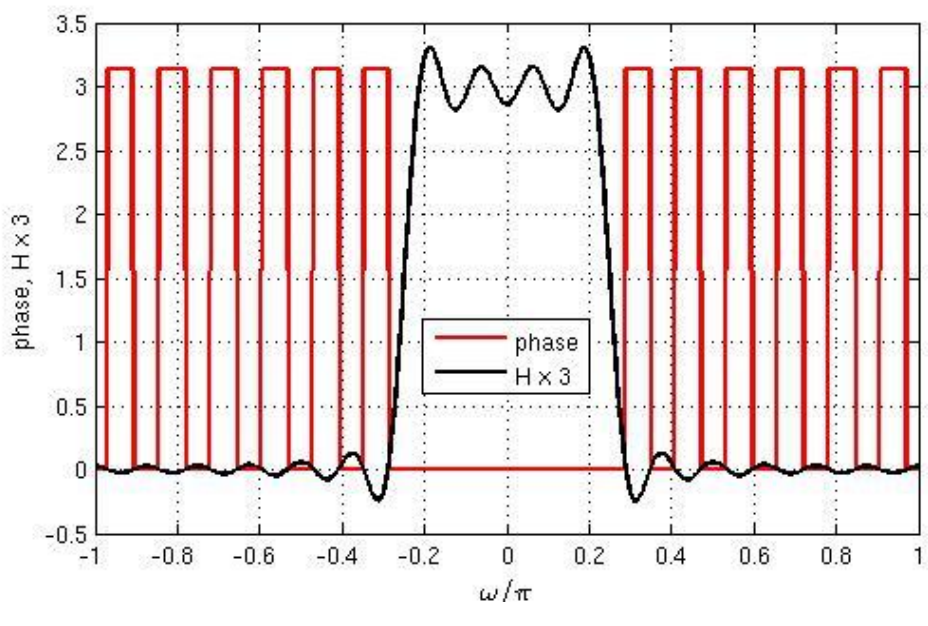
Z3a. Korzystając ze wzoru $H(e^{j\omega}) = \sum_{n=-N}^{N} h(n) e^{-j\omega n}$,

wyznaczyć ch-kę częstotliwościową filtrów z poprzedniego zadania dla różnych funkcji okien, przy N=15. Uwaga: zmienna ω jest ciągła, więc w przypadku realizacji komputerowej wartości charakterystyki wyznaczamy tylko dla wybranych wartości ω.

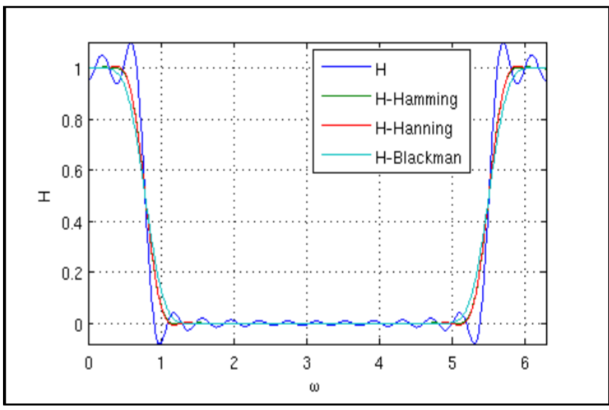


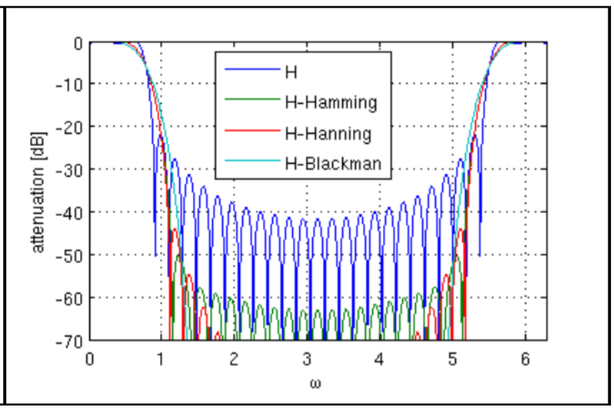


Ponieważ wyznaczana funkcja $H(e^{j\omega})$ jest zespolona, to zwykle przedstawia się jej przebieg w postaci dwóch wykresów: modułu *abs*, zwanego ch-ką amplitudową, i argumentu *angle*, zwanego ch-ka fazową. Ch-kę amplitudową zwykle przedstawia się stosując skalę logarytmiczną dla osi Y. Poniżej pokazano charakterystykę amplitudową (bez modułu, ale przeskalowaną w wartości 3 x) oraz charakterystykę fazową.



Uwaga: zwróć uwagę i przekonaj się doświadczalnie, że funkcja $H(e^{j\omega})$ jest okresowa.





Kod:

```
clc;clear all;close all;
N=15;
n=-N:N;
PI=pi;
w=linspace(-PI,PI,1000);

leng_n=size(n,2);%length(n);%
leng_w=size(w,2);%length(w);%

win_rect=rectwin(length(n));
win_hamm=hamming(length(n));
win_hann=hanning(length(n));
win_black=blackman(length(n));
wgr=pi/4;
%ujemna czeĹ›Ä‡
n1=-N:(-1);
h_lp_u=sin(wgr.*n1)./(pi().*n1);
%zero
h_lp_0=wgr/pi();
%czÄ™Ĺ›c+
n1=1:N;
h_lp_d=sin(wgr.*n1)./(pi().*n1);

hlp=[h_lp_u,h_lp_0,h_lp_d];
%----------------------------------------------
xw1=linspace(-PI/pi,-0.25,333);xw2=linspace(-
0.25,0.25,333);xw3=linspace(0.25,PI/pi,334);xw=[xw1,xw2,xw3];
sqr1=zeros(1,333);sqr2=ones(1,333);sqr3=zeros(1,334); sqr=[sqr1 sqr2 sqr3];
plot(xw, sqr, 'b'); hold on

%H
for zmienna=1:leng_w
    H(zmienna) = sum(hlp.*exp(-1i*w(zmienna)*n));
end
plot(w/pi,real(H),'g');grid on;

%%rect
hlp_rect=hlp'.*win_rect;
%H_rect=[];
for zmienna1=1:leng_w
    H_rect(zmienna1) = sum(hlp_rect'.*exp(-1i*w(zmienna1)*n));
end
plot(w/pi,real(H_rect),'g');grid on;hold on

%%hamming
hlp_hamm=hlp'.*win_hamm;
for zmienna2=1:leng_w
    H_hamm(zmienna2) = sum(hlp_hamm'.*exp(-1j*w(zmienna2)*n));
end
plot(w/pi,real(H_hamm),'r');grid on;hold on

%%hanning
hlp_hann=hlp'.*win_hann;
for zmienna3=1:leng_w
    H_hann(zmienna3) = sum(hlp_hann'.*exp(-1j*w(zmienna3)*n));
end
plot(w/pi,real(H_hann),'c');grid on;hold on

%%blackmann
hlp_black=hlp'.*win_black;
for zmienna4=1:leng_w
    H_black(zmienna4) = sum(hlp_black'.*exp(-1j*w(zmienna4)*n));
end
plot(w/pi,real(H_black),'m');grid on;hold on
```

```
xlabel('\omega/\pi')
ylabel('wzmocnienie')
legend('H^{(ideal)}','H^{(Rect)}','H^{(Hamming)}','H^{(Hanning)}','H^{(Blac
kman)}')

%%pozostaĹ‚e wykresy
figure(2)%skara logarytmiczna osi y
plot(w/pi,20*log10(abs(H_rect)),'b');grid on;hold on;%okno prostokÄ…tne
plot(w/pi,20*log10(abs(H_hamm)),'g');grid on;hold on;%okno Hamminga
plot(w/pi,20*log10(abs(H_hann)),'r');grid on;hold on;%okno Hanninga
plot(w/pi,20*log10(abs(H_black)),'c');grid on;hold on;%okno Blackmana
axis([-1 1 -70 0] )
xlabel('\omega/\pi')
ylabel('wzmocnienie [dB]')
legend('H^{(Rect)}','H^{(Hamming)}','H^{(Hanning)}','H^{(Blackman)}')

%phase
figure(3)
plot(w/pi,3*real(H),'k','LineWidth',1.2);grid on;hold on%H*3
plot(w/pi,angle(real(H)),'r');
xlabel('\omega/\pi')
ylabel('phase, H x 3')
legend('H x 3','phase')

%okresowoĹ›Ä‡ funkcji - sprawdzenie doĹ›wiadczalne
w1=linspace(0,2*pi,1000);
for zmo=1:leng_w
    H_w1(zmo)= sum(hlp.*exp(-1i*w1(zmo)*n));
    H_hamm1(zmo) = sum(hlp_hamm'.*exp(-1j*w1(zmo)*n));
    H_hann1(zmo) = sum(hlp_hann'.*exp(-1j*w1(zmo)*n));
    H_black1(zmo) = sum(hlp_black'.*exp(-1j*w1(zmo)*n));
end
figure(4)
plot(w1,real(H_w1),'b');grid on;hold on;
plot(w1,real(H_hamm1),'g')%okno hamminga
plot(w1,real(H_hann1),'r')%okno Hanninga
plot(w1,real(H_black1),'c')%okno blackmanna
axis([0 2*pi -0.1 1.1])
xlabel('\omega')
ylabel('H')
legend('H','H^{(Hamming)}','H^{(Hanning)}','H^{(Blackman)}')

figure(5) %skla logarytmiczna osi y
plot(w1,20*log10(abs(H_w1)),'b');grid on;hold on;%okno prostokÄ…tne
plot(w1,20*log10(abs(H_hamm1)),'g');grid on;hold on;%okno Hamminga
plot(w1,20*log10(abs(H_hann1)),'r');grid on;hold on;%okno Hanninga
plot(w1,20*log10(abs(H_black1)),'c');grid on;hold on;%okno Blackmana
axis([0 2*pi -70 0] )
xlabel('\omega')
ylabel('wzmocnienie [dB]')
legend('H','H^{(Hamming)}','H^{(Hanning)}','H^{(Blackman)}')

%-------------------------------------------------------------------
%% Filtr gĂłrnoprzepustowy

N=15;
n=-N:N;
wgr_g=pi()/8;
w=linspace(-pi,pi,1000);

leng_n=size(n,2);%length(n);%
leng_w=size(w,2);%length(w);%
%ujemna czeĹ›Ä‡
n2=-N:(-1);
h_hp_u=(-sin(wgr_g*n2))./(pi*n2);
%zero
```

```
h_hp_0=1-(wgr_g/pi);
%czÄ™Ĺ›c+
n2=1:N;
h_hp_d=(-sin(wgr_g*n2))./(pi*n2);

hhp=[h_hp_u,h_hp_0,h_hp_d];

hhp_rect=hhp'.*win_rect;
hhp_hamm=hhp'.*win_hamm;
hhp_hann=hhp'.*win_hann;
hhp_black=hhp'.*win_black;

for i=1:leng_w
    H_ideal_hp(i)=sum(hhp.*exp(-1i*w(i)*n));
    H_rect_hp(i)=sum(hhp_rect'.*exp(-1i*w(i)*n));
    H_hamm_hp(i)=sum(hhp_hamm'.*exp(-1i*w(i)*n));
    H_hann_hp(i)=sum(hhp_hann'.*exp(-1i*w(i)*n));
    H_black_hp(i)=sum(hhp_black'.*exp(-1i*w(i)*n));
end

figure(6)
plot(w/pi,real(H_w1),'b');grid on;hold on;
plot(w/pi,real(H_rect_hp),'g')%okno prostokÄ…tne
plot(w/pi,real(H_hamm_hp),'r')%okno hamminga
plot(w/pi,real(H_hann_hp),'c')%okno Hanninga
plot(w/pi,real(H_black_hp),'m')%okno blackmanna
%axis([0 2*pi -0.1 1.1])
xlabel('\omega')
ylabel('H')
legend('H^{(Ideal)}','H^{(Rect)}','H^{(Hamming)}','H^{(Hanning)}','H^{(Blac
kman)}')

figure(7) %skla logarytmiczna osi y
plot(w/pi,20*log10(abs(H_rect_hp)),'b');grid on;hold on;%okno prostokÄ…tne
plot(w/pi,20*log10(abs(H_hamm_hp)),'g');grid on;hold on;%okno Hamminga
plot(w/pi,20*log10(abs(H_hann_hp)),'r');grid on;hold on;%okno Hanninga
plot(w/pi,20*log10(abs(H_black_hp)),'c');grid on;hold on;%okno Blackmana
%axis([0 2*pi -70 0] )
xlabel('\omega')
ylabel('wzmocnienie [dB]')
legend('H','H^{(Hamming)}','H^{(Hanning)}','H^{(Blackman)}')

figure(8)%faza
plot(w/pi,3*real(H_ideal_hp),'k','LineWidth',1.2);grid on;hold on%H*3
plot(w/pi,angle(real(H_ideal_hp)),'r');
xlabel('\omega/\pi')
ylabel('phase, H x 3')
legend('H x 3','phase')
```

Z3b. Wyznaczyć ch-kę częstotliwościową filtrów z poprzedniego zadania dla różnych funkcji okien używając funkcji *freqz*. Uwaga: funkcja *freqz* wyznacza ch-kę filtru przyczynowego tzn. korzysta ze wzoru

$$H_P(e^{j\omega})=\sum_{n=0}^{L-1} h_p(n)e^{-j\omega n},$$

gdzie $L = 2N+1$ jest długością odpowiedzi impulsowej $h_p(n)=h(n-N)$. Wywołanie [*Hp*, *W*] = freqz(*h*) zwraca wektor próbek charakterystyki częstotliwościowej *Hp* oraz wektor wartości częstotliwości *W*, w których charakterystyka ta została wyliczona. Wywołanie *freqz*(*h*), z pominięciem argumentów wynikowych *Hp* i *W*, rysuje charakterystykę amplitudową i fazową. Charakterystyki obliczone przez funkcję *freqz* należy porównać z wynikami z punktu 3a. Następnie należy je skorygować (wektor *Hp*), tak by uzyskać charakterystykę filtru nieprzyczynowego tj. *H* tak jak w zadaniu 3a.

$$H_P(e^{j\omega}) = \sum_{n=0}^{2N} h_p(n) e^{-j\omega n} = \sum_{n=0}^{2N} h(n-N) e^{-j\omega n} = \sum_{n=-N}^{N} h(n) e^{-j\omega(n+N)} = e^{-j\omega N} H(e^{j\omega}).$$

Kod:

```
%% %Zadanie 3b
N=15;
n=-N:N;
w=linspace(-pi,pi,1000);
leng_w=length(w);
%okna
win_rect=rectwin(length(n));
win_hamm=hamming(length(n));
win_hann=hanning(length(n));
win_black=blackman(length(n));

%Filtr dolnoprzepustowy
wgr=pi/4;
%ujemna czeĹ›Ä‡
n1=-N:(-1);
h_lp_u=sin(wgr.*n1)./(pi().*n1);
%zero
h_lp_0=wgr/pi();
%czÄ™Ĺ›c+
n1=1:N;
h_lp_d=sin(wgr.*n1)./(pi().*n1);

hlp=[h_lp_u,h_lp_0,h_lp_d];

%okna
hlp_rect=hlp'.*win_rect;
hlp_hamm=hlp'.*win_hamm;
hlp_hann=hlp'.*win_hann;
hlp_black=hlp'.*win_black;

for i=1:leng_w
    H_lp(i)=sum(hlp.*exp(-1i*w(i)*n));
    H_rect_lp(i)=sum(hlp_rect'.*exp(-1i*w(i)*n));
    H_hamm_lp(i)=sum(hlp_hamm'.*exp(-1i*w(i)*n));
    H_hann_lp(i)=sum(hlp_hann'.*exp(-1i*w(i)*n));
    H_black_lp(i)=sum(hlp_black'.*exp(-1i*w(i)*n));
end

[hhlp,W_lp]=freqz(hlp);
figure(9)
subplot(3,1,1),plot(w/pi,20*log10(abs(H_lp)),'r');grid on; hold on;
xlabel('Charakterystyka z punktu 3a')
subplot(3,1,2),plot(W_lp/pi,20*log10(abs(hhlp)),'k');grid on
h_lp2=H_lp.*exp(-1j*w*N);
xlabel('Freqz')
subplot(3,1,3),plot(w/pi,20*log10(abs(h_lp2)),'b');grid on;
xlabel('Charakterystyka filtru po korekcie')

%okno hamminga
figure(10)
[h_hamm_lpp,W_hamm_lp]=freqz(H_hamm_lp);
subplot(3,1,1),plot(w/pi,20*log10(abs(H_hamm_lp)),'r');grid on; hold on;
xlabel('Charakterystyka z punktu 3a')
subplot(3,1,2),plot(W_hamm_lp/pi,20*log10(abs(h_hamm_lpp)),'k');grid on
h_lp22=H_hamm_lp.*exp(-1j*w*N);
xlabel('Freqz')
subplot(3,1,3),plot(w/pi,20*log10(abs(h_lp22)),'b');grid on;
xlabel('Charakterystyka filtru po korekcie')
title('Okno hamminga')
```

```
%filtr gĂłrnoprzepustowy
N=15;
n=-N:N;
wgr_g=pi()/8;
w=linspace(-pi,pi,1000);

leng_n=size(n,2);%length(n);%
leng_w=size(w,2);%length(w);%
%ujemna czeĹ›Ä‡
n2=-N:(-1);
h_hp_u=(-sin(wgr_g*n2))./(pi*n2);
%zero
h_hp_0=1-(wgr_g/pi);
%czÄ™Ĺ›c+
n2=1:N;
h_hp_d=(-sin(wgr_g*n2))./(pi*n2);

hhp=[h_hp_u,h_hp_0,h_hp_d];

hhp_rect=hhp'.*win_rect;
hhp_hamm=hhp'.*win_hamm;
hhp_hann=hhp'.*win_hann;
hhp_black=hhp'.*win_black;

for i=1:leng_w
    H_ideal_hp(i)=sum(hhp.*exp(-1i*w(i)*n));
    H_rect_hp(i)=sum(hhp_rect'.*exp(-1i*w(i)*n));
    H_hamm_hp(i)=sum(hhp_hamm'.*exp(-1i*w(i)*n));
    H_hann_hp(i)=sum(hhp_hann'.*exp(-1i*w(i)*n));
    H_black_hp(i)=sum(hhp_black'.*exp(-1i*w(i)*n));
end
[hhhp,W_hp]=freqz(hhp);
figure(11)
subplot(3,1,1),plot(w/pi,20*log10(abs(H_ideal_hp)),'r');grid on; hold on;
xlabel('Charakterystyka z punktu 3a')
subplot(3,1,2),plot(W_hp/pi,20*log10(abs(hhhp)),'k');grid on
h_hp2=H_ideal_hp.*exp(-1j*w*N);
xlabel('Freqz')
subplot(3,1,3),plot(w/pi,20*log10(abs(h_hp2)),'b');grid on;
xlabel('Charakterystyka filtru po korekcie')

figure(12)
%okno hamminga

[h_hamm_hpp,W_hamm_hp]=freqz(H_hamm_hp);

subplot(3,1,1),plot(w/pi,20*log10(abs(H_hamm_hp)),'r');grid on; hold on;
xlabel('Charakterystyka z punktu 3a')
subplot(3,1,2),plot(W_hamm_hp/pi,20*log10(abs(h_hamm_hpp)),'k');grid on
h_hp22=H_hamm_hp.*exp(-1j*w*N);
xlabel('Freqz')
subplot(3,1,3),plot(w/pi,20*log10(abs(h_hp22)),'b');grid on;
xlabel('Charakterystyka filtru po korekcie')
title('Okno hamminga')
```

Z4. Zaprojektować cyfrowy filtr dolnoprzepustowy o częstotliwości granicznej f0 = 150 Hz, przy częstotliwości próbkowania *fp* = 1000Hz. Narysować ch-kę amplitudową. Sprawdzić działanie zaprojektowanego filtru poprzez filtrację sygnału złożonego z dwóch sinusoid o częstotliwościach *f*1 = 100Hz i *f*2 = 250Hz.

```
%clc;clear all;close all;
f0=150;
```

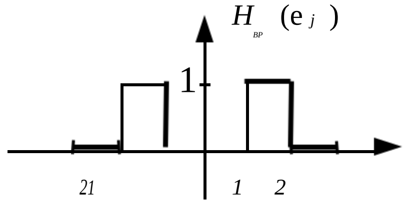
```
fp=1000;wp=2*pi*fp;
f1=100;w1=2*pi*f1;
f2=250;w2=2*pi*f2;
w=linspace(-pi,pi,fp);


leng_w=size(w,2)
w0=(f0/fp)*2*pi;
N=15;
n=-N:N;
%ujemna czeĹ›Ä‡
n1=-N:(-1);
h_lp_u=sin(w0*n1)./(pi()*n1);
%zero
h_lp_0=w0/pi();
%czÄ™Ĺ›c+
n1=1:N;
h_lp_d=sin(w0*n1)./(pi()*n1);

hlp=[h_lp_u,h_lp_0,h_lp_d];

%stem(n,hlp,'r');grid on
Np=100;
t=(0:1/fp:Np-1)/fp;
sin1=sin(w1*t);
sin2=sin(w2*t);
sin_sum=sin1+sin2;

figure (13)
subplot(3,1,1),plot(t,sin1);grid on
xlabel('sinus f1')
subplot(3,1,2),plot(t,sin2);grid on
xlabel('sinus f2')
subplot(3,1,3),plot(t,sin_sum);grid on
xlabel('suma sinusĂłw')

figure(14)
for zmienna=1:leng_w
    H4(zmienna) = sum(hlp.*exp(-1i*w(zmienna)*n));
end
plot(w/pi,real(H4),'g');grid on;hold on
figure(15)
[HZ4,W]=freqz(hlp);
plot(W/pi,20*log10(abs(HZ4)));grid on

fil=filter(H4,1,sin_sum);
figure(16)
plot(t,fil);grid on;
```

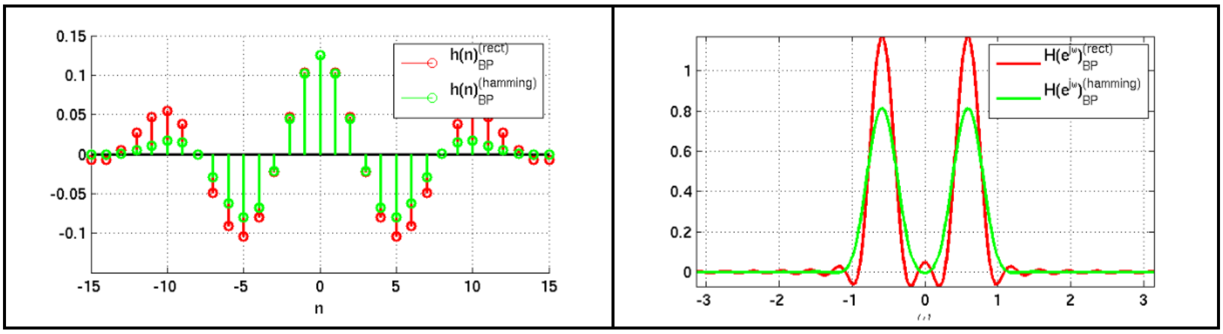
Z5a. Korzystając z metody okien wyznaczyć (analitycznie) odpowiedź impulsową idealnego filtru pasmowo-przepustowego $h_{BP}(n)$, a następnie obciąć ją wybranym oknem uzyskując $h^W_{BP}(n)$. Charakterystyka częstotliwościowa $H_{BP}(e^{j\omega})$ filtra idealnego pokazana jest poniżej.
Napisać skrypt w Matlabie rysujący odpowiedź impulsową (funkcja *stem*) oraz charakterystykę częstotliwościową zaprojektowanego filtra. Przykładowa charakterystyka częstotliwościowa filtru z oknem prostokątnym i oknem Hamminga dla N=15, $\omega_1=\pi/8$, $\omega_2=\pi/4$, poniżej.
Uwaga: do realizacji tego zadania można wykorzystać wynik zadania 2.

$$h_{BP}(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi} H_{BP}(e^{j\omega})e^{j\omega n}\,d\omega\ ,$$

$$h^W_{BP}(n)=w(n)h_{BP}(n)$$

$$-N\le n\le N$$





Kod:

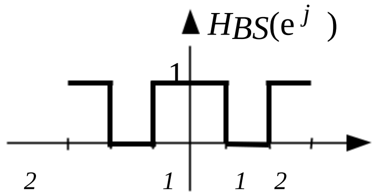
```
N=15;
w1=pi/8;
w2=pi/4;
w=linspace(-pi,pi,1000);
n=-N:N;

n1=-N:-1;
h_bsU=((sin(w2.*n1)-sin(w1*n1))./(pi*n1));

h_bs0=(w2-w1)/pi;
n1=1:N;
h_bsD=(sin(w2.*n1)-sin(w1*n1))./(pi*n1);

h_bs=[h_bsU,h_bs0,h_bsD];

win_hamm=hamming(length(n));

%h_bp_rect=h_bp'.*rectwin(length(n));
h_bs_hamm=h_bs'.*win_hamm;


for i=1:length(w)

    H(i)=sum(h_bs.*exp(-1j*w(i)*n));
   H_bs_hamm(i)=sum(h_bs_hamm'.*exp(-1j*w(i)*n));
end

figure(1)
stem(n,h_bs,'r');grid on;hold on;
stem(n,h_bs_hamm,'g');
legend('h(n)_{BP}^{(rect)}','h(n)_{BP}^{(hamming)}')
```

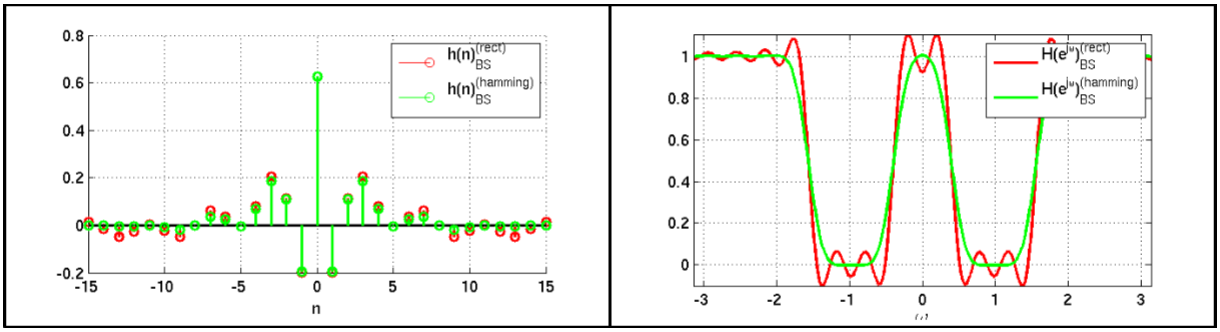
```
xlabel('n')

figure(2)
plot(w/pi,real(H),'r');grid on;hold on;
plot(w/pi,real(H_bs_hamm),'g');
legend('H(e^{j\omega})_{BP}^{(rect)}','H(e^{j\omega})_{BP}^{(hamming)}')
xlabel('\omega/\pi')
```

Z5b. Korzystając z metody okien wyznaczyć (analitycznie) odpowiedź impulsową idealnego filtru pasmowo-zaporowego $h_{BS}(n)$, a następnie obciąć ją wybranym oknem uzyskując $h^{W}_{BS}(n)$. Charakterystyka częstotliwościowa $H_{BS}(e^{j\omega})$ filtra idealnego pokazana jest poniżej. Napisać skrypt w Matlabie rysujący odpowiedź impulsową oraz charakterystykę częstotliwościową zaprojektowanego filtra. Przykładowa charakterystyka częstotliwościowa filtru z oknem prostokątnym i oknem Hamminga dla N=15, $\omega_1=\pi/8$, $\omega_2=\pi/2$, poniżej.
Uwaga: do realizacji tego zadania można wykorzystać wynik zadania 2.

$$h_{BS}(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_{BS}(e^{j\omega})e^{j\omega n}\, d\omega$$

$$h^{W}_{BS}(n) = w(n)h_{BS}(n)$$





Kod:

```
N=15;
w1=pi/8;
w2=pi/2;
w=linspace(-pi,pi,1000);
n=-N:N;

n1=-N:-1;
h_bsU=((sin(w1.*n1)-sin(w2*n1))./(pi*n1));

h_bs0=1+(w1-w2)/pi;
n1=1:N;
h_bsD=(sin(w1.*n1)-sin(w2*n1))./(pi*n1);

h_bs=[h_bsU,h_bs0,h_bsD];

win_hamm=hamming(length(n));

%h_bp_rect=h_bp'.*rectwin(length(n));
h_bs_hamm=h_bs'.*win_hamm;


for i=1:length(w)

    H(i)=sum(h_bs.*exp(-1j*w(i)*n));
   H_bs_hamm(i)=sum(h_bs_hamm'.*exp(-1j*w(i)*n));
end

figure(1)
stem(n,h_bs,'r');grid on;hold on;
stem(n,h_bs_hamm,'g');
legend('h(n)_{BP}^{(rect)}','h(n)_{BP}^{(hamming)}')
xlabel('n')

figure(2)
plot(w,real(H),'r');grid on;hold on;
plot(w,real(H_bs_hamm),'g');
legend('H(e^{j\omega})_{BS}^{(rect)}','H(e^{j\omega})_{BS}^{(hamming)}')
xlabel('\omega')
```

Z6. Korzystając z funkcji: *fir1* oraz *kaiserord* i *kaiser* pakietu Matlab zaprojektować następujące filtry z oknem Kaisera:

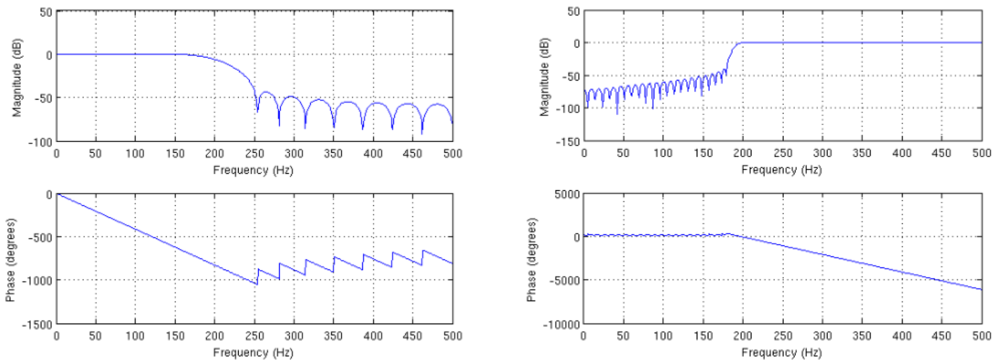
a) dolnoprzepustowy: pasmo przenoszenia do 150 Hz, nierównomierność w paśmie przenoszenia poniżej 0.05, minimalne tłumienie w paśmie zaporowym 40dB dla 250 Hz, częstotliwości próbkowania fp = 1000Hz;

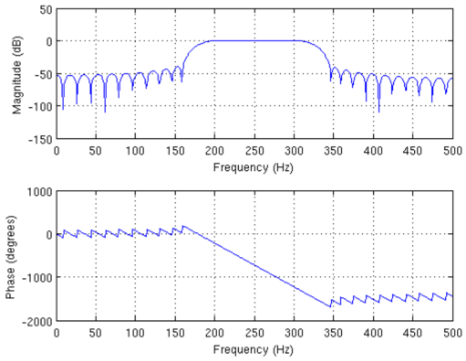
b) górnoprzepustowy: częstotliwość graniczna pasma przenoszenia fg = 180 Hz, nierównomierność w paśmie przenoszenia poniżej 0.05, minimalne tłumienie w paśmie zaporowym 40dB dla 200 Hz, częstotliwości próbkowania fp = 1000Hz;

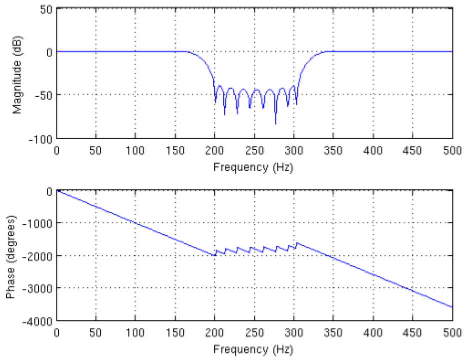
c) pasmowo-przepustowy: zakres częstotliwości pasma przepustowego 200 Hz do 300 Hz, nierównomierność 0.05, częstotliwości pasma zaporowego to: 160 Hz i 350 Hz. Minimalne tłumienie dla podanych częstotliwości to 40dB;

d) pasmowo-zaporowy: zakres częstotliwości pasma zaporowego 200 Hz do 300 Hz. Minimalne tłumienie dla podanych częstotliwości to 40dB. Granice pasma przepustowego to: 160 Hz i 350 Hz. Nierównomierność ch-ki częstotliwościowej w paśmie przepustowym to 0.05;

Narysować charakterystyki częstotliwościowe uzyskanych filtrów. Sprawdzić, czy zaprojektowane filtry spełniają wymagania projektowe.







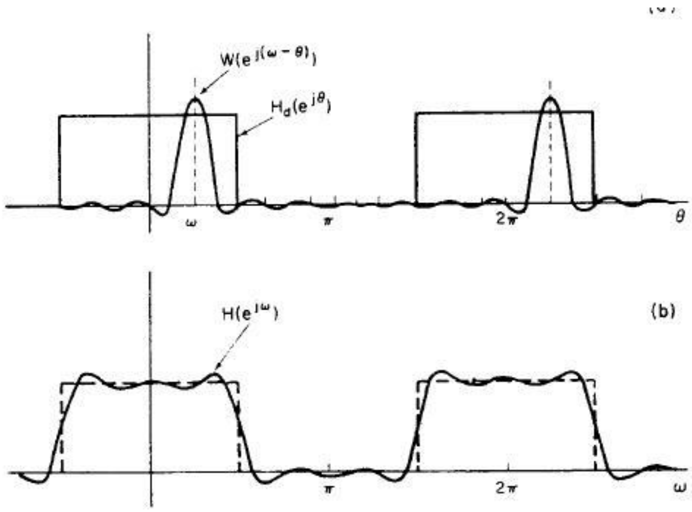

```
fp=1000;

f_mag = [150 250];
a = [1 0];
odch = [0.05 0.01];%0.01-> 20log10(0.01) =-40dB

[n,Wn,beta,ftype] = kaiserord(f_mag,a,odch,fp);
win_kai_LP=kaiser(n+1,beta);
h_LP = fir1(n,Wn,ftype,win_kai_LP,'noscale');

figure(1)
freqz(h_LP,1,1024,fp);grid on;

%-----------------------------------------------------
%gĂłrnoprzepustowy
fp=1000;

f_mag = [180 200];
a = [0 1];
odch = [0.05 0.01];%0.01-> 20log10(0.01) =-40dB

[n,Wn,beta,ftype] = kaiserord(f_mag,a,odch,fp);
win_kai_HP=kaiser(n+1,beta);
h_HP = fir1(n,Wn,ftype,win_kai_HP,'noscale');

figure(2)
freqz(h_HP,1,1024,fp);grid on
%-------------------------------------------------------
%pasmowo-przepustowy

fp=1000;

f_mag = [160 200 300 350];
a = [0 1 0];
odch = [0.01 0.05 0.01];%0.01-> 20log10(0.01) =-40dB

[n,Wn,beta,ftype] = kaiserord(f_mag,a,odch,fp);
win_kai_BP=kaiser(n+1,beta);
h_BP = fir1(n,Wn,ftype,win_kai_BP,'noscale');

figure(3)
freqz(h_BP,1,1024,fp);grid on

%--------------------------------------------------------
%Pasmowow-zaporowy
fp=1000;

f_mag = [160 200 300 350];
a = [1 0 1];
```

```
odch = [0.05 0.01 0.05];%0.01-> 20log10(0.01) =-40dB

[n,Wn,beta,ftype] = kaiserord(f_mag,a,odch,fp);
win_kai_BS=kaiser(n+1,beta);
h_BS = fir1(n,Wn,ftype,win_kai_BS,'noscale');

figure(4)
freqz(h_BS,1,1024,fp);grid on
```

Z7. Sprawdzić numerycznie podaną zależność

$$H_w(e^{j\omega})=\sum_{n=0}^{N-1} h(n)w(n)e^{-j\omega n}=\frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\Omega})W(e^{j(\omega-\Omega)})d\Omega$$

Zależność ta wiąże widmo filtra o odpowiedzi impulsowej *h*(*n*) obciętej oknem *w*(*n*) z widmem filtra idealnego (nie obciętego) i widmem okna.



Teoria i Przetwarzanie Sygnałów – Laboratorium

Układy dyskretne LTI – projektowanie filtrów typu IIR

Strasznie niepewne

Z1. Zaprojektować filtr silnie wzmacniający sygnał o częstotliwość $\omega_0 = \pi/4$. Zrobić rysunek pokazujący rozmieszczenie zer i biegunów transmitancji filtra. Zapisać transmitancję filtra korzystając z ogólnej jej postaci, gdzie d_k są biegunami, zaś c_k zerami transmitancji:

$$H(z) = K\frac{\prod_{k=1}^{N}(1-c_k z^{-1})}{\prod_{k=1}^{M}(1-d_k z^{-1})} = z^{-N+M}K\frac{\prod_{k=1}^{N}(z-c_k)}{\prod_{k=1}^{M}(z-d_k)},$$

Przydatne funkcje Matlaba to: *poly* oraz *conj*. Korzystając z funkcji *freqz* wyznaczyć ch-kę częstotliwościową zaprojektowanego filtra. Następnie korzystając z funkcji *filter* wyznaczyć odpowiedź impulsową zaprojektowanego filtra (odpowiedź na pobudzenie sygnałem o postaci d = [1, 0, 0, ….]). W rozwiązaniu wykorzystać funkcję *poly*.

Uwaga: funkcja *poly* jako argument otrzymuje wektor miejsc zerowych wielomianu i zwraca jego współczynniki, np. *poly*([1, 2]) zwraca [1, -3, 2], tj. $(z-1)(z-2) = 1\cdot z^2 - 3\cdot z + 2$.

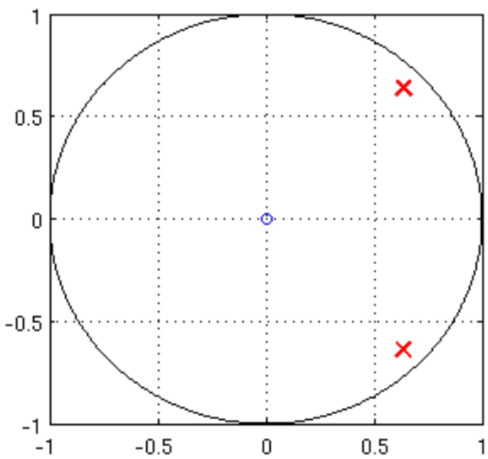
W algorytmach przetwarzania sygnałów stosowane są wielomiany zmiennej z^{-1}. Zauważmy, że elementy wektora [1, 2] są również miejscami zerowymi wielomianu zmiennej z^{-1}:

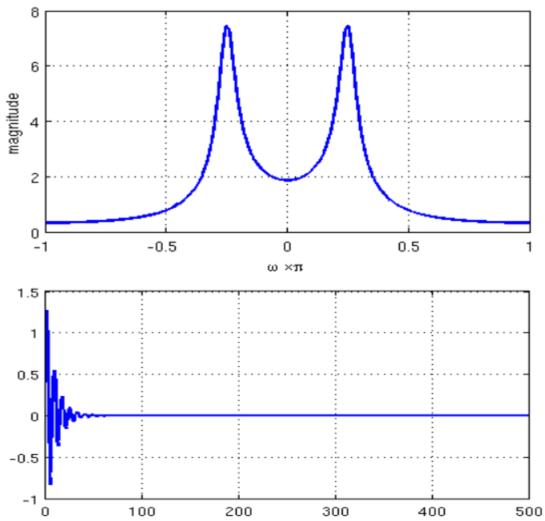
$$(1-1\cdot z^{-1}1)(1-2\cdot z^{-1}) = 1-3\cdot z^{-1}+2\cdot z^{-2}$$

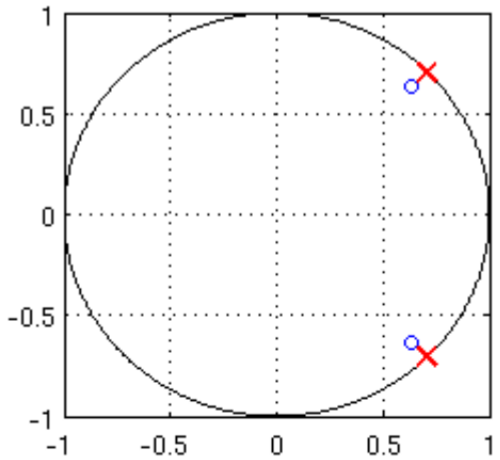
Dlatego funkcję *poly* można bezpośrednio wykorzystać do wyznaczania współczynników transmitancji układu, które mogą być użyte przez np. funkcję *filter(B, A, x)*, która stosuje następującą notację dla współczynników b i a:

$a(1)*y(n)$ = $b(1)*x(n) + b(2)*x(n-1) + ... + b(Nb+1)*x(n-Nb)$ $- a(2)*y(n-1) - ... - a(Na+1)*y(n-Na)$,

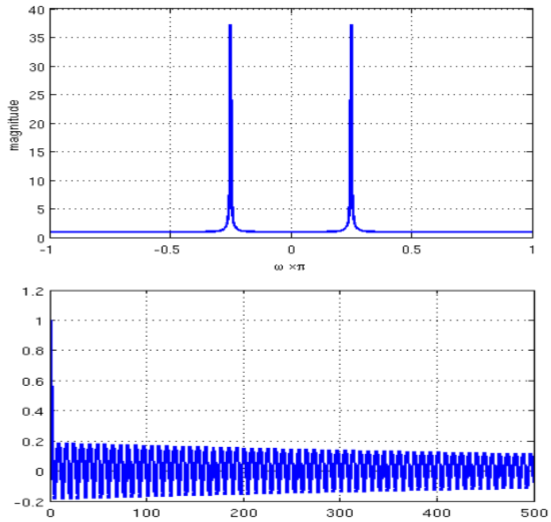
tj. współczynnik wielomianu $A(z^{-1})$ lub $B(z^{-1})$ przy z^{-n} jest na $(n+1)$ miejscu wektora A lub B.







o – zero, x – biegun



Zauważ, że: $(1-d_1 z^{-1})(1-d_2 z^{-1}) = z^{-2}(z-d_1)(z-d_2) = \frac{(z-d_1)(z-d_2)}{z^2}$.

W oparciu o powyższe spostrzeżenie, wytłumacz obecność podwójnego zera w transmitancji filtru pierwszego.

Kod: (sam nw)
```
clear; clc; close all;

%--Zadanie 1
d=[1,zeros([1 499])]; %sygnaĹ, impulsowy
w0=pi/4;
w=linspace(-pi,pi,1000);
rp=0.99;
p1=rp*exp(j*w0);
rz=0.9;
z1=rz*exp(j*w0);
P=poly([p1 conj(p1)]); %mianownik- bieguny
Z=poly([z1 conj(z1)]); %licznik- zera

figure(1)
plot(real(z1), imag(z1),'go'); hold on; grid on;
plot(real(conj(z1)), imag(conj(z1)),'go');
plot(real(p1), imag(p1),'rx');
plot(real(conj(p1)), imag(conj(p1)),'rx');
plot(real(exp(j*w)),imag(exp(j*w)),'b');

figure(2);
[h,w]=freqz(Z,P);
plot(w/pi,h)
grid on; xlabel('\omega x \pi'); ylabel('Wzmocnienie');

y=filter(Z,P,d);
figure(3);
plot(y)
grid on;
```

Z2. Korzystając z funkcji *polyval* oraz poniższego wzoru wyznaczyć ch-ki częstotliwościowe filtrów z poprzedniego zadania. Porównać uzyskane wyniki z wynikiem funkcji *freqz*.

$$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_{M-1} z^{-(M-1)}}{a_0 + a_1 z^{-1} + \ldots + a_{N-1} z^{-(N-1)}}, \quad H(e^{j\omega}) = \frac{b_0 + b_1 e^{-j\omega} + \ldots + b_{M-1} e^{-j\omega(M-1)}}{a_0 + a_1 e^{-j\omega} + \ldots + a_{N-1} e^{-j\omega(N-1)}},$$

przy czym $a_0 = 1$.

$K-1$ $M-1$

$$y(n)=\sum_{k=0} b(k)x(n-k)-\sum_{k=1} a(k)y(n-k)$$

Uwaga: funkcja *freqz(B, A)* wyznacza wartość $H(z) = B(z)/A(z)$ dla z leżących na okręgu jednostkowym ($z=e^{j\omega}$). Współczynniki wielomianów $B(z)$ i $A(z)$ uporządkowane są w następujący sposób: $B=[b_0, b_1, b_2, ...]$, $A=[a_0, a_1, a_2, ...]$.

Funkcja *Y=polyval(P, X)* wyznacza wartość wielomianu o współczynnikach zadanych wektorem $P=[p_0, p_1, p_2, ...]$, dla wartości podanych w wektorze X, tj.

$$Y = p_0 x^N + p_1 x^{N-1} + ... + p_{N-1}$$

Wykorzystanie funkcji *polyval* do wyznaczania ch-ki częstotliwościowej układu wymaga: (1) odwrócenia kolejności elementów wektorów zawierających współczynniki transmitancji układu (funkcja *fliplr*), (2) wyznaczenia wartości wielomianu dla $x=e^{-j\omega}$, przy ω zmieniającym się w zakresie od $-\pi$ do π .

```
clear; clc; close all;

%--Zadanie 1
d=[1,zeros([1 499])]; %sygnaĹ, impulsowy
w0=pi/4;
w=linspace(-pi,pi,1000);
rp=0.99;
p1=rp*exp(j*w0);
rz=0.9;
z1=rz*exp(j*w0);
P=poly([p1 conj(p1)]); %mianownik- bieguny
Z=poly([z1 conj(z1)]); %licznik- zera

figure(1)
plot(real(z1), imag(z1),'go'); hold on; grid on;
plot(real(conj(z1)), imag(conj(z1)),'go');
plot(real(p1), imag(p1),'rx');
plot(real(conj(p1)), imag(conj(p1)),'rx');
plot(real(exp(j*w)),imag(exp(j*w)),'b');

figure(2);
[h,w]=freqz(Z,P);
plot(w/pi,h)
grid on; xlabel('\omega x \pi'); ylabel('Wzmocnienie');

y=filter(Z,P,d);
figure(3);
plot(y)
grid on;

%--Zadanie 2
x=exp(-j*w);
B=polyval(fliplr(Z),x);
A=polyval(fliplr(P),x);
H=B./A;
figure(4);
plot(w/pi,H); grid on;
```

Z3. Zaprojektować filtr silnie tłumiący sygnał (tzw. notch) o częstotliwość f0 = 990Hz, częstotliwość próbkowania 8000Hz. Zrobić rysunek pokazujący rozmieszczenie zer i ewentualnych biegunów transmitancji filtra. Wyznaczyć odpowiedź impulsową zaprojektowanego filtra.

Kod:

```
clc; close all ; clear all ;
```

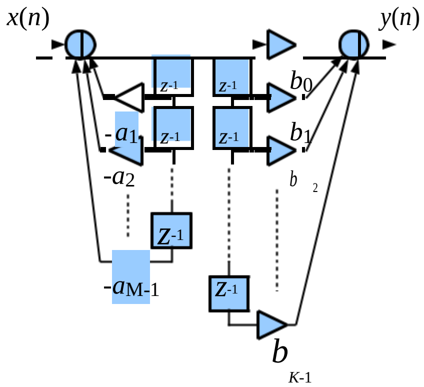
```
f0 = 990;
fp = 8000;
sb = [f0-1 f0+1]*2*(1/fp); % stopband
pb = [f0-5 f0+5]*2*(1/fp); % passband
pr = 0.1; % passripples
sr = 60; % stopripples

[N, Wn] = ellipord(pb, sb, pr, sr);
[b, a] = ellip(N, pr, sr, Wn, 'stop');

figure(1)
freqz(b,a)
title('Freqz plot')

t = linspace(0,2*pi,200);
r = 1;
x = r*cos(t);
y = r*sin(t);
%PT: w = linspace(0,2*pi,200); plot(exp(j*w))

figure(2)
plot(x, y, '.b')
grid on;
hold on;

[zeros, poles, gain] = ellip(N, pr, sr, Wn, 'stop');
plot(poles, 'xr')
hold on;
plot(zeros, 'ob')
title('Zero-pole plot')
```

Z4.Wyznaczyć zera i bieguny układu dyskretnego o poniższej transmitancji. Czy układ jest stabilny ?

$$H(z)=\frac{(z^{-1}-j)(z^{-1}+j)}{(z^{-1}+0.5-0.5j)(z^{-1}+0.5+0.5j)}$$

```
%Wyznaczenie zer i biegunow z transmitancji
L1 = [-1i 1]; L2 = [1i 1]; L = L1.*L2; %nie wiem czy współczynniki
%we właściwej kolejności
M1 = [(-0.5*1i+0.5) 1]; M2 = [(0.5*1i+0.5) 1]; M = M1.*M2;
[H, W] = freqz(L,M);
figure(10); zplane(L,M);
```

Z5. Napisać procedurę implementującą (wg poniższego schematu blokowego) filtry IIR z zadania **Z1**. Porównać jej działanie z funkcją *filter*, a następnie wykorzystać do wyznaczenia odpowiedzi impulsowych filtrów zaprojektowanych w z zadaniu **Z1**.



Kod:

```
w0=pi/4;
w=linspace(-pi,pi,1000);
rp=0.99;
p1=rp*exp(j*w0);
rz=0.9;
z1=rz*exp(j*w0);
P=poly([p1 conj(p1)]); %mianownik- bieguny
Z=poly([z1 conj(z1)]); %licznik- zera
x=[1, zeros([1 499])];

Nx=length(x);
M=length(P);
F=zeros(1,M-1);
K=length(Z);
D=zeros(1,K-1);
y=0;
    for k=1:Nx
    y(k)=x(k)*Z(1)+sum(D.*Z(2:end))-sum(F.*P(2:end));
    D(2:end)=D(1:end-1);
    D(1)=x(k);
    F(2:end)=F(1:end-1);
    F(1)=y(k);
    end
  figure(8)
  plot(y)
  grid on; ylabel('y(n)'); xlabel('n');
y_filter=filter(Z,B,x);
e=y_filter-y;
figure(9)
stem(e)
grid on; ylabel('e(n)'); xlabel('n');
```

Z6. Narysować: charakterystyki częstotliwościowe i odpowiedzi impulsowe filtrów o niżej podanych transmitancjach. Następnie określić położenie miejsc zerowych licznika i mianownika transmitancji. Stwierdzić czy filtr jest stabilny, czy nie.

$$H_1(z)=\frac{1}{(1+0.5z^{-1})}\ ,\qquad H_2(z)=\frac{1}{(0.5+z^{-1})}\ .$$

Kod:

```
%--Zadanie 6
num_coef=[1]; %wspĂłĹ‚czynniki transmitancji
den_coef=[0.5 1]; %(1+0.5z^-1)
figure(10);
zplane(num_coef,den_coef) %poĹ‚oĹĽenia biegunĂłw
[H,w]=freqz(num_coef,den_coef); %ch-ka czestotliwosciowa
figure(11);
plot(w,H); grid on; xlabel('\omega'); ylabel('H');
d=[1,zeros([1 499])]; %sygnaĹ‚ impulsowy
y=filter(num_coef,den_coef,d); %odpowiedĹş impulsowa
figure(12);
plot(y); grid on; title('Odp impulsowa');
zera=roots(num_coef);
bieguny=roots(den_coef);
if(abs(bieguny)<1)
    stabilnosc=1;
else stabilnosc=0;
end
```

Z7. Używając funkcji Matlaba (buttord, butter) do projektowania filtrów IIR zaprojektować filtr pasmowo przepustowy na częstotliwości Wp = [700 800], Ws = [650 850] Hz przy częstotliwości próbkowania 2000 Hz, zafalowaniach ch-ki amplitudowej w paśmie przenoszenia Rp = 0.1 dB oraz tłumienie Rs = 40 dB.

Kod:

```
Wp=[700 800]/1000;
Ws=[650 850]/1000;
Rp=0.1;
Rs=40;
[n,Wn]=buttord(Wp,Ws,Rp,Rs);
[z,p,k]=butter(n,Wn);
sos=zp2sos(z,p,k);
figure(13)
freqz(sos,512,2000)
```

Dyskretna transformacja Fouriera (DFT)

Z1. Stosując algorytm dyskretnej transformacji Fouriera (DFT)

$$X(k)=\sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}kn}, \quad k=0,1,\ldots,N-1,$$

wyznacz widma $X(k)$ następujących sygnałów zespolonych:

$x(n)=e^{j\frac{2\pi}{N}mn}$ oraz $x(n)=e^{-j\frac{2\pi}{N}mn}$,

dla n=0,1,...,N-1 i różnych *całkowitoliczbowych* wartości $0\le m\le N-1$.

Zauważ, że $x(n)=e^{-j\frac{2\pi}{N}mn}=e^{j\frac{2\pi}{N}(N-m)n}$.

Wyznacz $X(k)$ w sposób analityczny (na papierze) i porównaj otrzymany wynik z obliczeniami numerycznymi (matlab).

```
%Wyznaczanie widma sygnalow zespolonych uzywajac DFT
N = 50;
n = 0:N-1;
m =[0;1;8;16;N-1];
x1 = exp(1j*2*pi/N*m*n);
x2 = exp(-1j*2*pi/N*m*n);

figure(1)
for w = 1:length(m)
    for k = 0:N-1
        X1(w,k+1) = sum(x1(w,:).*exp(1j*2*pi/N*k.*n));
        X2(w,k+1) = sum(x2(w,:).*exp(-1j*2*pi/N*k.*n));
    end
    subplot(1,2,1), stem(0:k, abs(X2(w,:))), hold on
    subplot(1,2,2), stem(0:k, abs(X1(w,:))), hold on
end

subplot(1,2,1), xlim([0, 50]), grid on, title('x2 = exp(j2\pi/mn)')
legend('m=0','m=1','m=8','m=16','m=N-1'), xlabel('k')
subplot(1,2,2), xlim([0, 50]), grid on, title('x1 = exp(-j2\pi/mn)')
legend('m=0','m=1','m=8','m=16','m=N-1'), xlabel('k')
```

Z2. Stosując algorytm dyskretnej transformacji Fouriera (DFT) wyznaczyć widma poniższych sygnałów mających po N próbek (n=0,1,...,N-1).

1) $x(n)=\mathrm{DC}+\sin(\frac{2\pi}{N}mn)$, $x(n)=\mathrm{DC}+\cos(\frac{2\pi}{N}mn)$,

   dla m będącego liczbą całkowitą $0\le m\le N-1$ i dla różnych wartości DC,

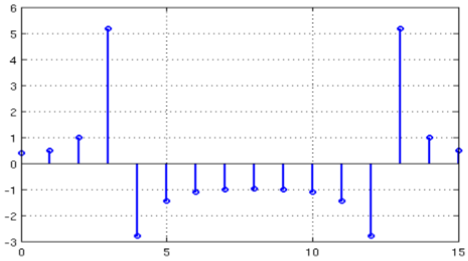
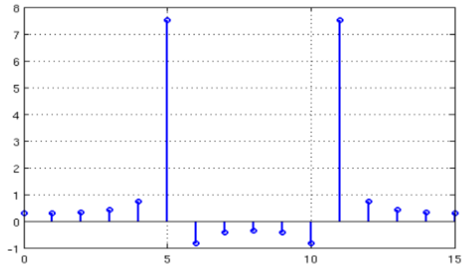
2) $x(n)=\sin(\frac{2\pi}{N}mn+\phi)$,

   dla dla m będącego liczbą całkowitą i różnych wartości przesunięcia fazy ϕ,

3) $x(n)=\sin(\frac{2\pi}{N}mn)$, dla różnych wartości m nie będących liczbami całkowitymi.

Narysuj część rzeczywistą i urojoną widma (na jednym rysunki) oraz moduł (na drugim rysunku). Zaobserwuj symetrię modułu widma, oraz części rzeczywistej i urojonej widma.

Dla przypadku, kiedy *m* jest liczbą całkowitą wyznacz $X(k)$ w sposób analityczny i porównaj otrzymany wynik z wynikiem z matlaba.

x(n) = sin(2*pi/16*3.3*[0:15])	x(n) = cos(2*pi/16*5.1*[0:15])
	
real(X)	real(X)

```
N=15;
n=[0:N-1];
m=2;

DC=5;
x=DC+sin(2*pi*m*n/N);
for k=0:N-1
    X(k+1)=sum(x(n+1).*exp(-j*2*pi*k*n/N));
end
figure(2)
subplot(2,1,1);
stem(n,real(X)); grid on; title('Re(X)') %widma sin
subplot(2,1,2);
stem(n,imag(X)); grid on; title('Im(X)')

x=DC+cos(2*pi*m*n/N);
for k=0:N-1
    X(k+1)=sum(x(n+1).*exp(-j*2*pi*k*n/N));
end
figure(3)
subplot(2,1,1);
stem(n,real(X)); grid on; title('Re(X)') %widma cos
subplot(2,1,2);
stem(n,imag(X)); grid on; title('Im(X)')

%2)
fi=pi/4;
x=sin(2*pi*m*n/N+fi);
for k=0:N-1
    X(k+1)=sum(x(n+1).*exp(-j*2*pi*k*n/N));
end
figure(4)
subplot(2,1,1);
stem(n,real(X)); grid on; title('Re(X)') %widma dnaej 2 funkcji
subplot(2,1,2);
stem(n,imag(X)); grid on; title('Im(X)')

%3)
m=3.5;
x=sin(2*pi*m*n/N+fi);
for k=0:N-1
    X(k+1)=sum(x(n+1).*exp(-j*2*pi*k*n/N));
end
figure(5)
subplot(2,1,1);
stem(n,real(X)); grid on; title('Re(X)') %widma danej 3 funkcji
subplot(2,1,2);
stem(n,imag(X)); grid on; title('Im(X)')
```

Z3. Zaprojektować algorytm do analizy częstotliwościowej sygnałów pozwalającej na rozróżnienie składowych różniących się o 1 Hz, i rozpiętości amplitudy 1000x, częstotliwość próbkowania fp=1000Hz.

Uwaga: wykorzystać DFT z oknem Kaisera. Szerokość listka głównego okna Kaisera i poziom 1-szego listka bocznego warunkowana jest parametrem i długością okna *N*.

$$\beta = \begin{cases} 0, & A_{sl} < 13.26dB \\ 0.76609(A_{sl} - 13.26)^{0.4} + 0.09834(A_{sl} - 13.26), & 13.26 < A_{sl} < 60dB \\ 0.12438(A_{sl} + 6.3), & A_{sl} > 60dB \end{cases}$$

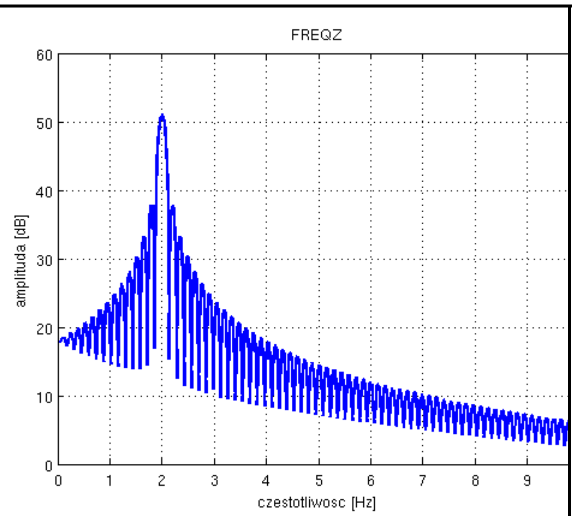
$$N = \left\lceil \frac{24\pi(A_{sl} + 12)}{155\Delta_{ml}} \right\rceil + 1$$

W powyższym wzorze $_{ml}$ jest szerokością listka głównego wyrażoną w [rad/s], tj $_{ml} = 2\pi(\Delta f / f_p)$, a A_{sl} = 60 dB poziomem tłumienia listków bocznych.

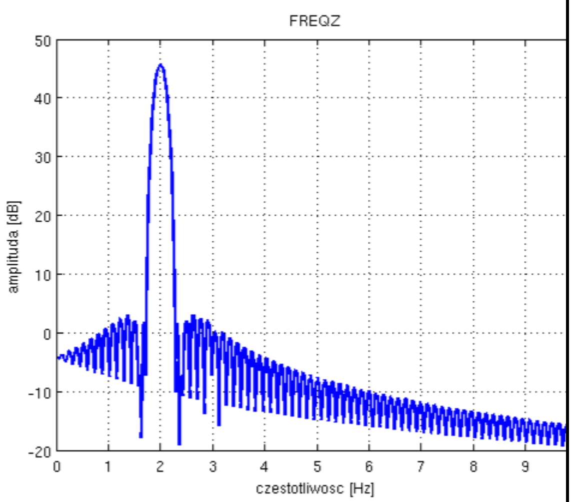
Przykładowo, weźmy sygnał

```
f1 = 2;
f2 = f1+1;
n = 0:N-1;
x = sin(2*pi*f1/fp*n) + 0.001*cos(2*pi*f2/fp*n);
```

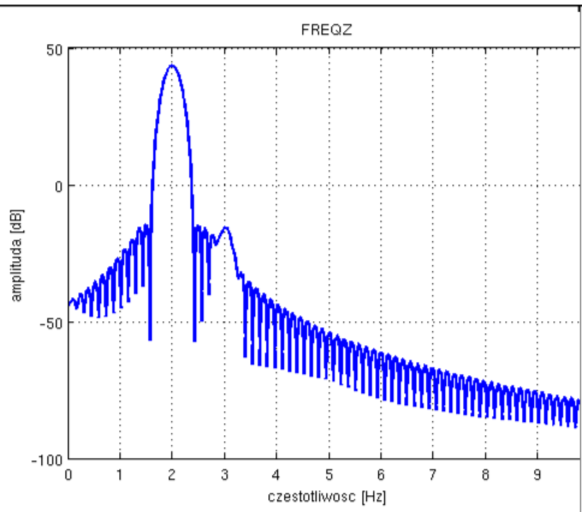
Stosując znane okna otrzymujemy następujące widma:



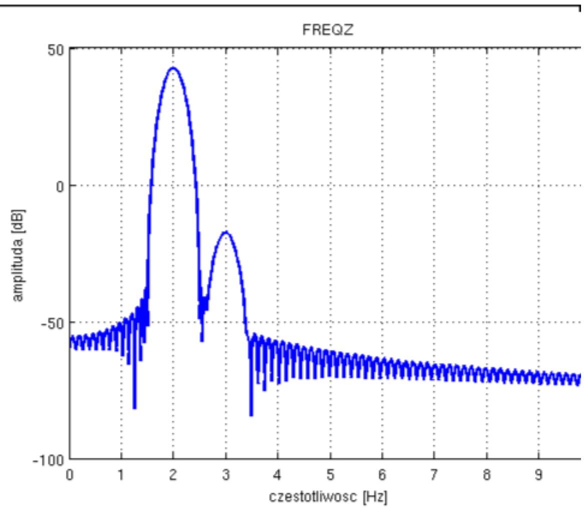
Okno prostokątne



Okno Hamminga



Okno Blackmana



Okno Kaisera

```
%analiza czestotliwosciowa rozrozniajaca skladowe
df = 1;
fp = 1000;
Asl = 80;
dml = 2*pi*df/fp;
beta = 0.12438*(Asl+6.3);
N = round((24*pi*(Asl+12))/(155*dml)+1);
n = 0:N-1;

f1 = 2;
f2 = f1+df;
x = sin(2*pi*f1/fp*n)+0.001*cos(2*pi*f2/fp*n);

w = kaiser(N,beta)';
xw = x.*w;
figure(5)
f = 0:0.001:10;
freqz(xw,1,f,fp)
```

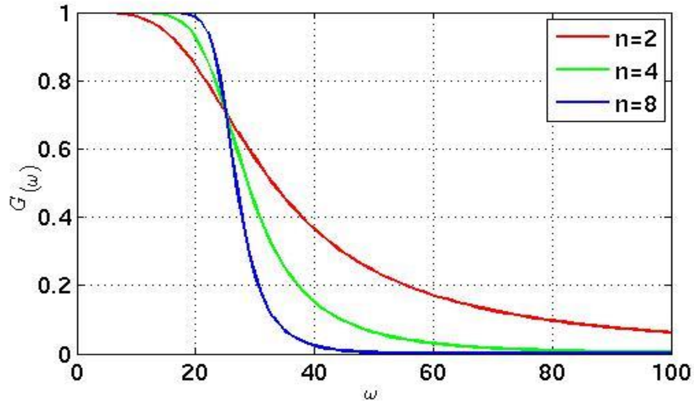
Z4. Napisać dwie wersje funkcji wyznaczającej splot kołowy dwóch sygnałów. Wersja pierwsza powinna korzystać z definicji splotu, a wersja druga z twierdzenia o splocie dla DFT.

```
%splot sygnałow
x = [1 2 3 2 1];
h = [0 1 2 3 4];
N = length(x);

%wersja 1 - z definicji splotu
y=zeros(1,N);
for n = 0:N-1
    y(n+1) = sum(x.*circshift(fliplr(h),n+1));
end

%wersja 2 - z DFT
n = 0:N-1;
for k = 0:N-1
    X(k+1) = sum(x.*exp(-1j*2*pi/N*k.*n));
    H(k+1) = sum(h.*exp(-1j*2*pi/N*k.*n));
end
Y = X.*H;
for k = 0:N-1 %odwrotna transformacja
    yp(k+1) = 1/N*sum(Y.*exp(1j*2*pi/N*k.*n));
end
y
```

Projektowanie cyfrowych filtrów IIR typu Butherworth'a

Moduł charakterystyki częstotliwościowej filtru analogowego *n*-tego rzędu

$$G(\omega)=|(H(j\omega))|=\frac{1}{\sqrt{1+\left(\frac{j\omega}{j\omega_c}\right)^{2n}}}$$



$$G(\omega)=|(H(j\omega))|=\frac{1}{\sqrt{1+\left(\frac{j\omega}{j\omega_c}\right)^{2n}}}$$

Z1. Napisz skrypt, który narysuje moduł charakterystyki częstotliwościowej filtru analogowego o rzędzie *n* = 2, 4, i 8, tak jak pokazano to na rysunku powyżej.

Kod:

```
clc;clear all;close all;
n=2;
w=0:0.01:100;
wc=25;
G_n2=1./(sqrt(1+(1j*w/(1j*wc)).^(2*n)));

plot(w,G_n2,'r','LineWidth',1.1);
grid on;
hold on

n=4;
G_n4=1./(sqrt(1+(1j*w/(1j*wc)).^(2*n)));
plot(w,G_n4,'g','LineWidth',1.1);
grid on;
hold on

n=8;
G_n8=1./(sqrt(1+(1j*w/(1j*wc)).^(2*n)));

plot(w,G_n8,'b','LineWidth',1.1);
grid on;
hold on
xlabel('\omega')
ylabel('G(\omega)')
legend('n=2','n=4','n=8')
```

Z2. Korzystając z metody transformacji biliniowej zaprojektuj cyfrowy dolnoprzepustowy filtr Buttherworth'a o częstotliwości granicznej $\omega_{3dB} = 0.8 \cdot \pi = 2\pi f_{3dB} / f_s$ i rzędzie n=3.

Rozwiązanie powyższego zadania wymaga wykonania kilku kroków

K1. wyznaczamy częstotliwość graniczna (3dB) filtra analogowego:

$$\omega_c = 2/T\, tg(\omega_{3dB}/2)$$

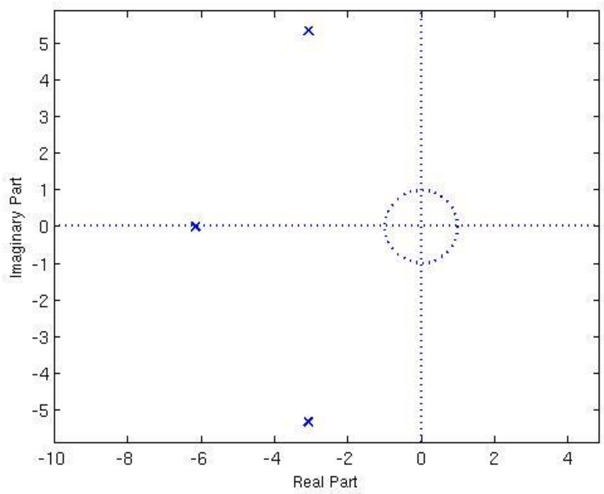
przyjmując $f_s = 1$ mamy $T = 1$

$$\omega_c = 2\, tg(\omega_{3dB}/2)$$

K2. wyznaczamy bieguny analogowego filtra Buttherworth'a:

$$s_k = \omega_c\, e^{j\pi(1+2k)/2n} e^{j\pi/2}, \quad k = 0,1,\dots, n-1$$

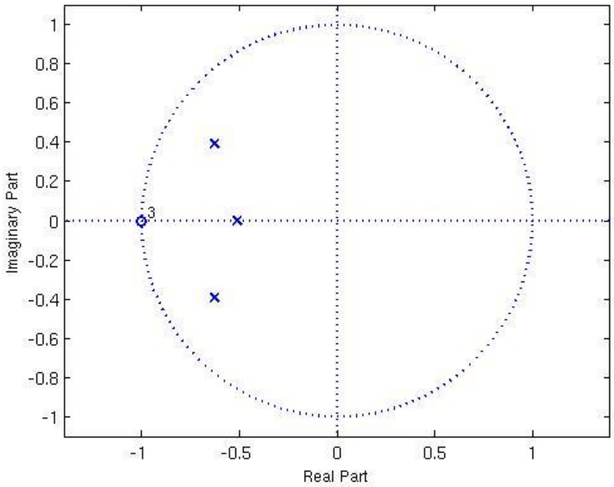
korzystając z funkcji *zplane* rysujemy położenie wyznaczonych biegunów filtru analogowego



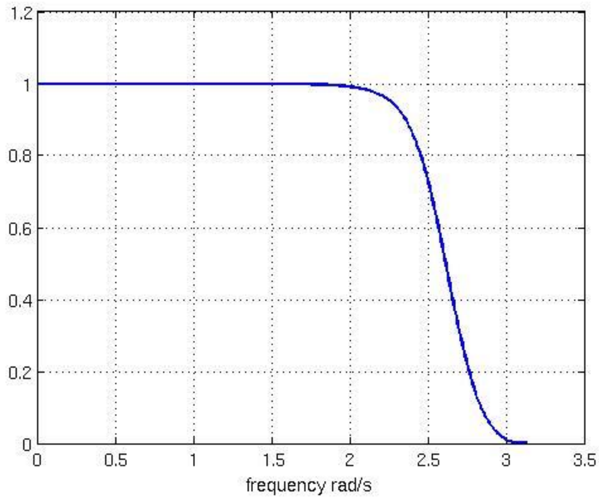
K3. Korzystając z transformacji biliniowej budujemy transmitancję filtru cyfrowego

$$H(z)=G_0 \frac{\prod_{k=1}^{n}(z-(-1))}{\prod_{k=1}^{n}(z-(2+s_k)/(2-s_k))}$$, gdzie czynnik $G_0=\prod_{k=1}^{n}(-s_k)/\prod_{k=1}^{n}(2-s_k)$

jest wzmocnieniem DC (dla ω=0) filtra. Zera transmitancji $H(z)$ są w miejscach z= -1, a bieguny w $(2+s_k)/(2-s_k)$. Korzystając z funkcji *zplane* rysujemy położenie wyznaczonych zer i biegunów filtru cyfrowego.



K4. Korzystając z funkcji *prod* i *poly* budujemy licznik i mianownik transmitancji $H(z)$ filtru cyfrowego i wyznaczamy jego charakterystykę częstotliwościową. Sprawdzamy, czy dla częstotliwości granicznej $\omega_{3dB}=0.8\cdot\pi$ wzmocnienie wynosi $1/\sqrt{2}$.



Kod:

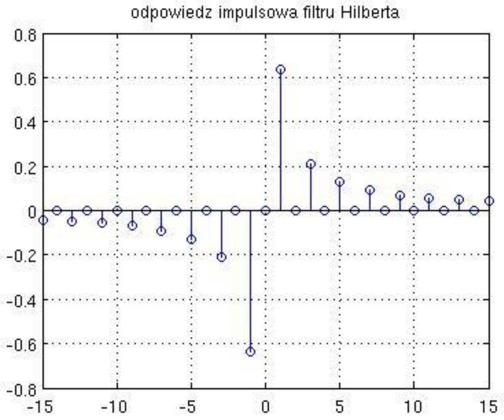
```
clc;clear all; close all;
n=3;
w3dB=0.8*pi;

%K1
T=1; fs=1;
wc=(2/T)*tan(w3dB/2);
%K2
k=0:1:n-1;
sk=wc*exp(j*pi()*(1+2*k)/(2*n))*exp(j*pi/2);

sk=sk';

zplane([],sk)
title('Bieguny filtru analogowego')

%K3
G0=prod(-sk)/prod(2-sk);
figure(2)
zera=[-1 -1 -1]';
bieguny=(2+(sk))./(2-(sk));
zplane(zera,bieguny)
title('Bieguny filtru cyfrowego')

%K4

mian = poly(bieguny);
licz =  poly(zera);

figure(3)

[H,w1]=freqz(G0*licz,mian);
plot(w1,(abs(H)));grid on;
```

Transformata Hilberta, sygnał analityczny, demodulacja amplitudy i fazy

Z1. Napisać skrypt w języku Matlab (lub funkcję) wyznaczającą odpowiedz impulsową filtra Hilberta, wg równania:

$$h_w(n)=\begin{cases} w(n)\dfrac{1}{\pi n}(1-(-1)^n), & -N<n<N \ \text{ i } n\neq 0 \\ 0, & n=0 \end{cases},$$

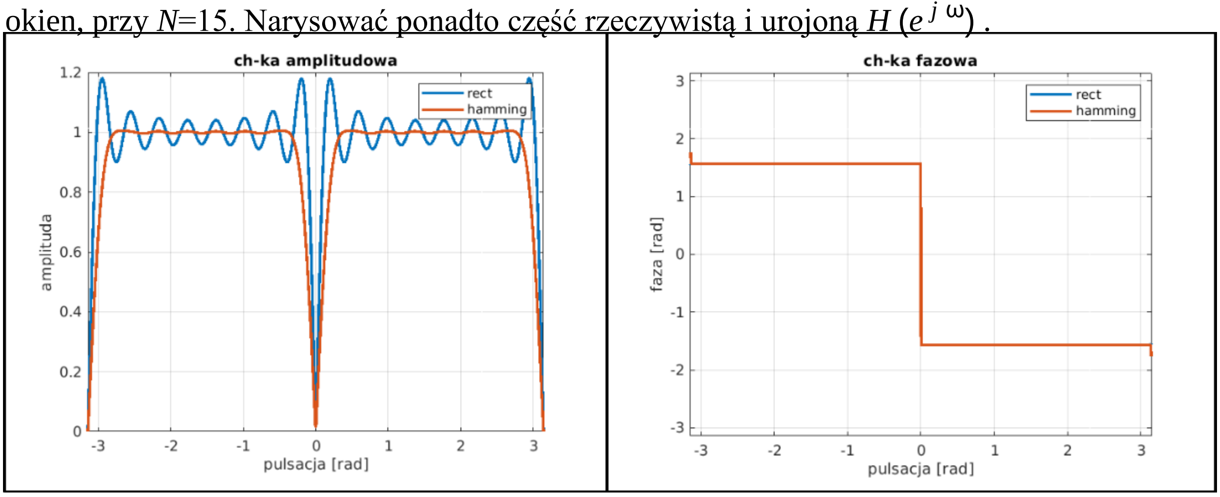
gdzie $w(n)$ jest funkcją okna czasowego (hamming, blackman, itp.). N – określa długość odpowiedzi impulsowej filtru $L=2N+1$.



Kod:
```
N=15;
L=2*N+1;
wblack=(blackman(2*N+1))';
n=-N:1:N;
n_minus=-N:-1;
    hw1=((1-(-1).^n_minus)./(pi.*n_minus));

n_plus=1:N;
     hw3=((1-(-1).^n_plus)./(pi.*n_plus));
hw=[hw1 0 hw3];

hw_black=wblack.*[hw1 0 hw3];
stem(n,hw_black);
grid on;
title('odpowiedĹş impulsowa filtru Hilberta')
```

Z2. Korzystając ze wzoru $H(e^{j\omega})=\sum_{n=-N}^{N} h(n)e^{-j\omega n}$,

wyznaczyć ch-kę amplitudową i fazową filtru Hilberta. Cha-ki wyznaczyć dla różnych funkcji okien, przy N=15. Narysować ponadto część rzeczywistą i urojoną $H(e^{j\omega})$.



Zwrócić uwagę, że zmienna ω jest ciągła, więc w przypadku realizacji komputerowej wartości tej funkcji wyznaczamy tylko dla wybranych dyskretnych wartości.

Ponieważ wyznaczana funkcja $H(e^{j\omega})$ jest zespolona, to zwykle przedstawia się jej przebieg w postaci dwóch wykresów: modułu *abs*, zwanego ch-ką amplitudową, i argumentu *angle*, zwanego ch-ka fazową.

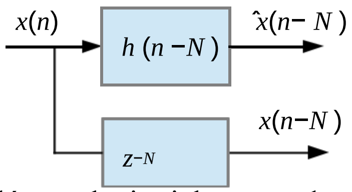
Uwaga: zwróć uwagę i przekonaj się doświadczalnie, że funkcja $H(e^{j\omega})$ jest okresowa.

```
N=15;
L=2*N+1;
wblack=(blackman(2*N+1))';
n=-N:1:N;
n_minus=-N:-1;
    hw1=((1-(-1).^n_minus)./(pi.*n_minus));

n_plus=1:N;
     hw3=((1-(-1).^n_plus)./(pi.*n_plus));
hw=[hw1 0 hw3];

hw_black=wblack.*[hw1 0 hw3];

w_omega=linspace(-pi,pi,1000);
n=-N:1:N;
for zmienna2=1:length(w_omega)
    H_black(zmienna2) = sum(hw_black.*exp(-1j*w_omega(zmienna2)*n));
end


for zmienna2=1:length(w_omega)
    H(zmienna2) = sum(hw.*exp(-1j*w_omega(zmienna2)*n));
end
figure(2)
plot(w_omega,abs(H),'b');
grid on;
hold on;
plot(w_omega,abs(H_black),'r');
grid on;
hold on
legend('rect','hamming')
title('ch-ka amplitudowa')
xlim([-pi,pi])

figure(3)
plot(w_omega,angle(H),'b');
grid on;
hold on;
plot(w_omega,angle(H_black),'r');
grid on;
hold on
legend('rect','hamming')
title('ch-ka fazowa')
xlim([-pi,pi])
```

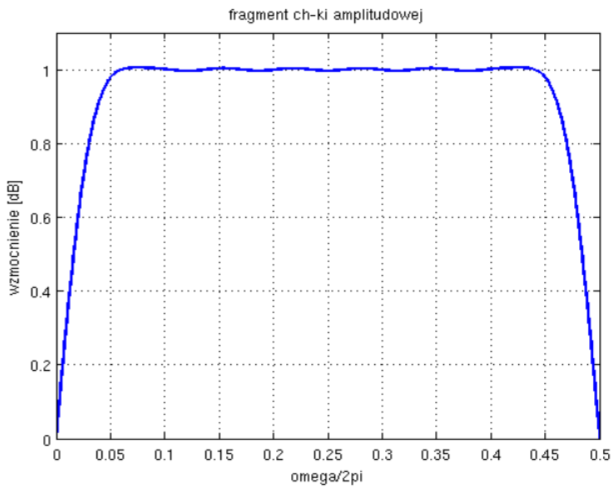
Z3. Korzystając z wyników poprzednich zadań, napisać program wyznaczający sygnał analityczny odpowiadający danemu sygnałowi rzeczywistemu. Sygnał analityczny definiowany jest poniższym wzorem

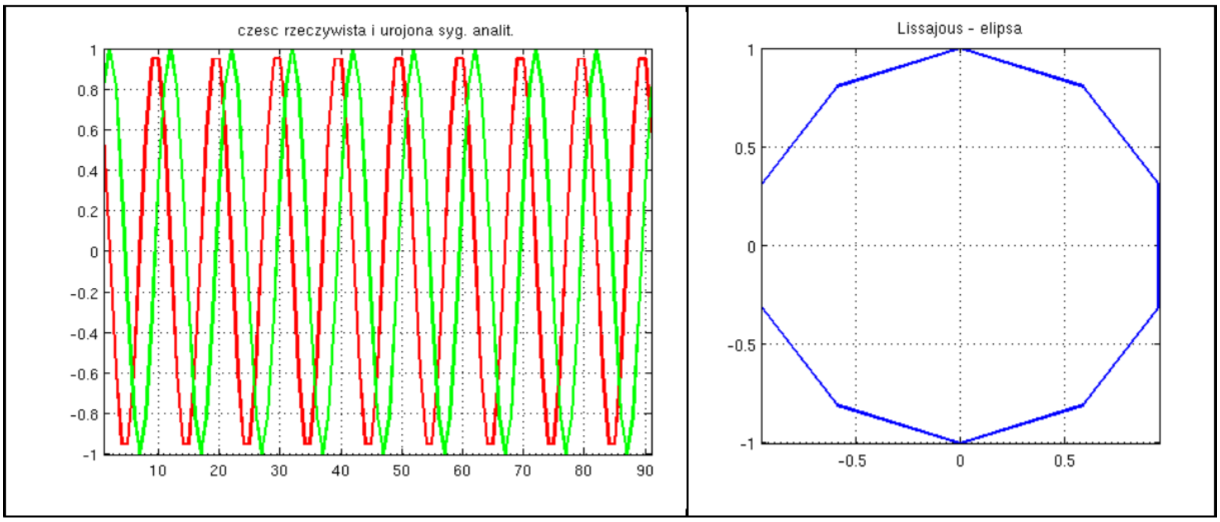
$$x_+(n)=x(n)+j\hat{x}(n)$$

i może być uzyskany z wykorzystaniem filtru Hilberta h(n) zgodnie z poniższym schematem.



Dla celów zadania, jako sygnał rzeczywisty $x(n)$ przyjąć przebieg kosinusoidalnie zmienny o pulsacji unormowanej równej 2*pi*0.1 rad, a następnie 2*pi*0.12 rad.






```
N=15;
L=2*N+1;
w=blackman(L);

for n=-N:N
    if n ~= 0
        hb(n+N+1)=w(n+N+1)/(pi*n)*(1-(-1)^n);
    end
end
Nh=N;
N = 4*2*Nh;
n = 0:N-1;


xx=cos(2*pi*0.1.*n);
%xx=cos(2*pi*0.1.*n);
;

x_hi=filter(hb, 1, xx);

x=[zeros(1, Nh), xx(1:N-Nh)];
xa=x+j*x_hi;
```

```
xa = xa(2*Nh:N);


figure(4)
n1=1:length(xa);
plot(n1, imag(xa),'g', n1, real(xa),'r')
axis('tight');
```

Z4. Demodulacja amplitudy. Załóżmy, że dany jest sygnał

$$x(n)=A(n)\cos(\omega_0 n)\,,\ A(n)\geq 0 \tag{1}$$

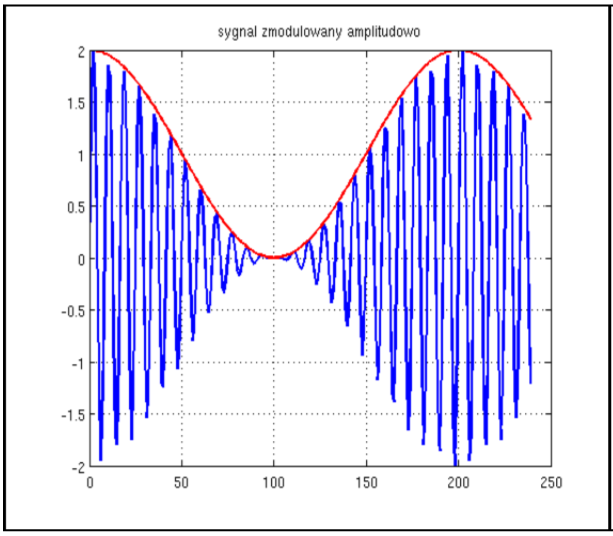
i poszukujemy wartości $A(n)$, która jest funkcją wolnozmienną i stanowi obwiednię sygnału $x(n)$. Zadanie to można łatwo rozwiązać mając do dyspozycji tzw. sygnał analityczny odpowiadający sygnałowi (1):

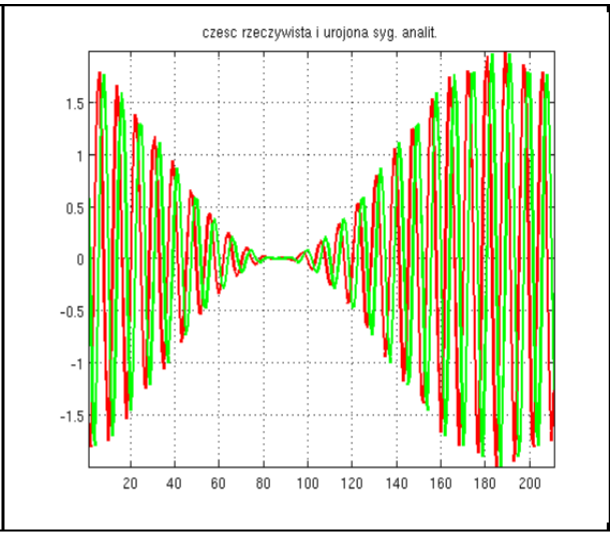
$$x_+(n)=x(n)+j\,\hat{x}(n)=A(n)\cos(\omega_0 n)+j\,A(n)\sin(\omega_0 n)$$

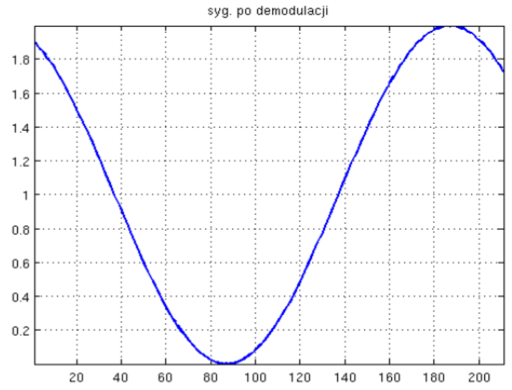
Wtedy

$$|x(n)| = \sqrt{(\Re\{x_+(n)\})^2+(\Im\{x_+(n)\})^2} = A(n)\sqrt{\cos^2(\omega_0 n)+\sin^2(\omega_0 n)} = A(n)$$

W zadaniu przyjąć, że częstotliwość unormowana przebiegu $A(n)$ jest równa `2*pi*0.005 rad`. Jako częstotliwość kątową nośnej przyjąć: 1) `2*pi*0.05 rad`, oraz 2) `2*pi*0.1 rad`. Wyjaśnić czym spowodowana jest różnica w jakości pracy demodulatora przy dwóch różnych częstotliwościach nośnej.







Kod:

```
N=15;
L=2*N+1;
w=blackman(L);

for n=-N:N
    if n ~= 0
        hb(n+N+1)=w(n+N+1)/(pi*n)*(1-(-1)^n);
    end
end
Nh=N;
N = 8*2*Nh;
n = 0:N-1;
n2=n;

w0=2*pi*0.005;
wkn1=2*pi*0.1;  %PT: 0.05 to troche za malo
%wkn2=2*pi*0.01;
a=1+cos(w0*n);

x=a.*sin(wkn1*n);
a1=a;
x1=x;
N=length(x);

x_hi=filter(hb, 1, x);

x=[zeros(1, Nh), x(1:N-Nh)];
x_an=x+j*x_hi;

x_an = x_an(2*Nh:N);

n1=1:length(x_an);
```

```
figure(5)


hold on
plot(n2, x1,'b')
plot( n2, a1,'r')
title("SygnaĹ, zmodulowany aplitudowo");
%
figure(6)
plot(n1, imag(x_an),'g')
hold on
plot( n1, real(x_an),'r')
title("Czesc rzeczywista i urojona syg.anl");
figure(7)
a_hat = sqrt(real(x_an).^2 + imag(x_an).^2);

plot(a_hat, 'LineWidth', 2);
grid;
title('syg. po demodulacji');
axis('tight')
```

Z5. Demodulacja fazy. Załóżmy, że dane są dwa sygnały x(n) i y(n) jak poniżej

$$x(n)=A(n)\cos(\omega_0 n), \quad y(n)=B(n)\cos(\omega_0 n+\varphi(n)) \tag{28}$$

$$x_+(n)=x(n)+j\,\hat{x}(n)=A(n)\cos(\omega_0 n)+j\,A(n)\sin(\omega_0 n) \tag{29}$$

$$y_+(n)=y(n)+j\,\hat{y}(n)=B(n)\cos(\omega_0 n+\varphi(n))+j\,B(n)\sin(\omega_0 n+\varphi(n)) \tag{30}$$

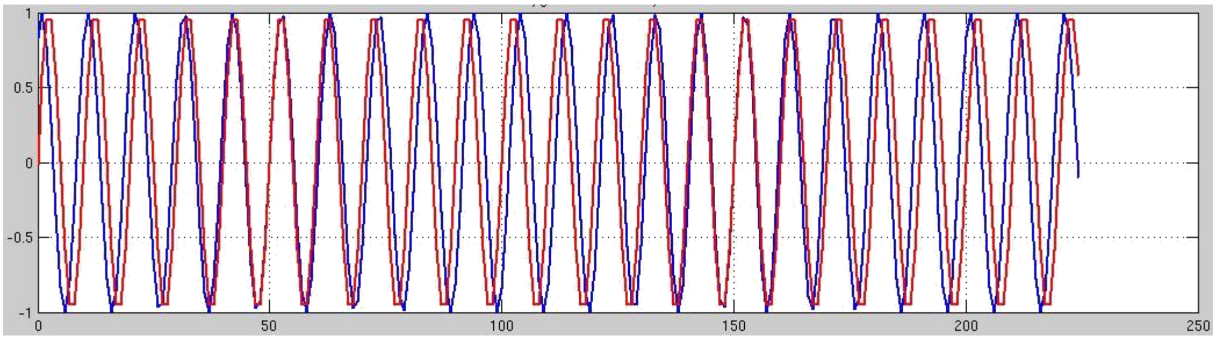
$$x_+(n)=A(n)e^{j\omega_0 n} \quad , \quad y_+(n)=B(n)e^{j(\omega_0 n+\varphi(n))} \tag{31}$$

$$x_+^*(n)\,y_+(n)=A(n)e^{-j\omega_0 n}B(n)e^{j(\omega_0 n+\varphi(n))} \tag{32}$$

$$x_+^*(n)\,y_+(n)=A(n)B(n)e^{j\varphi(n)} \tag{33}$$

$$tg\,\varphi(n)=\frac{\Im(x_+^*(n)\,y_+(n))}{\Re(x_+^*(n)\,y_+(n))} \tag{34}$$

Należy utworzyć dwa sygnały analityczne (29) i (30), a następnie ich iloczyn jak pokazuje (33). Wzór (34) można zaimplementować w Matlabie korzystając z funkcji *angle*, której argumentem będzie sygnał (33).
Jako sygnał x(n) można wykorzystać sygnał nośny z zadania Z4, tj przyjmując ża A(n)=1. Jako sygnał $\phi(n)$ można wziąć sygnał A(n) z zadania Z4. Sygnały te pokazano na poniższym rysunku.



Rys. Sygnał x(n) i y(n)