

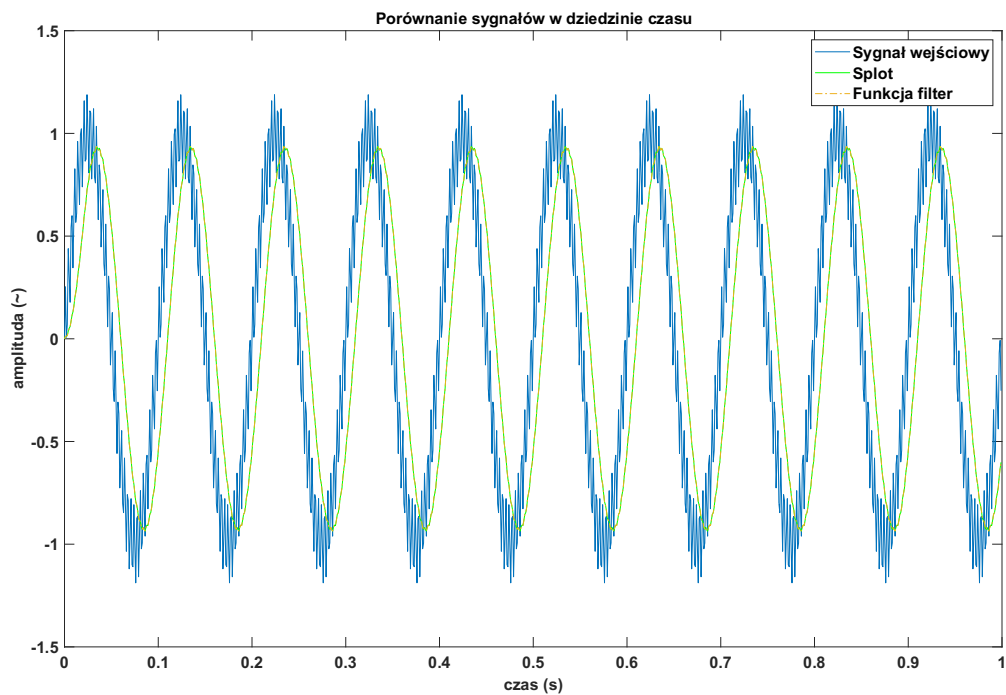
	<p style="text-align: center;">Wydział Inżynierii Mechanicznej i Robotyki</p> <p style="text-align: center;">Katedra Robotyki i Mechatroniki</p>	
<p style="text-align: center;">Identyfikacja Układów Mechatronicznych</p> <p style="text-align: center;">Sprawozdanie</p>		
<p>Laboratorium 2: Filtracja cyfrowa – Filtry FIR i IIR</p>		
<p>Imię i Nazwisko: Jakub Nowak, Maciej Mazurkiewicz</p>		
<p>Nr grupy: 2</p>		

## Zadanie 1 – Metody filtracji w Matlabie

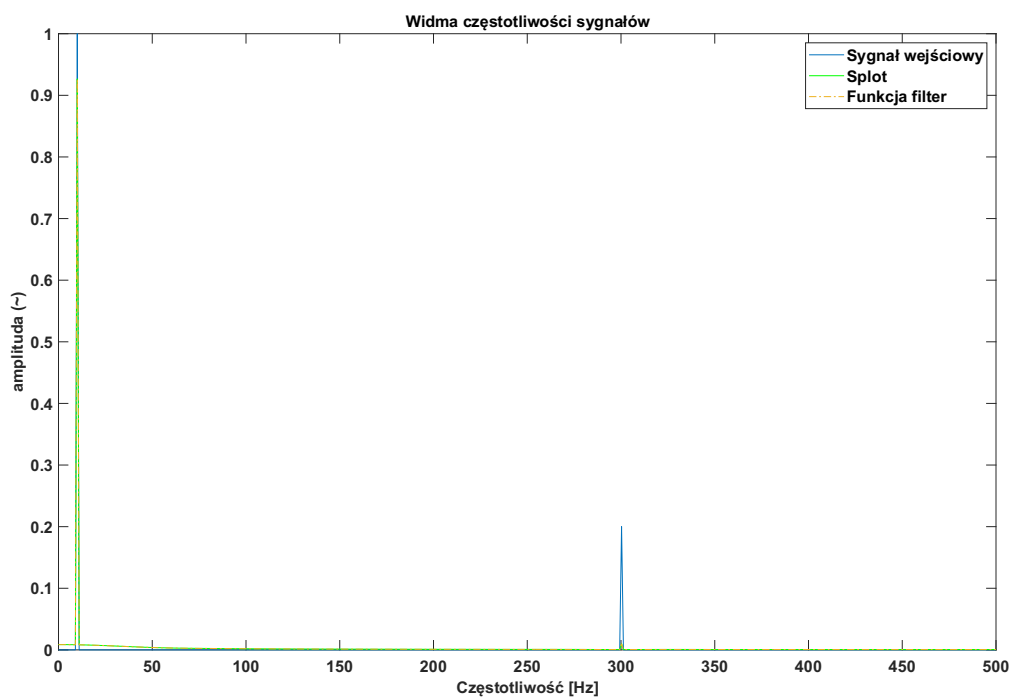
### Zadanie 1.1

Zadanie 1.1 polegało na filtracji sygnału o dwóch częstotliwościach składowych: 10 Hz oraz 300 Hz. Do tego celu zostało wykorzystane okno prostokątne o długości 21 próbek, które zastosowano poprzez 2 sposoby: splot oraz funkcję *filter*. Widoczna na **Rysunku 1.1** charakterystyka ukazuje, że sygnały pochodzące z powyższych sposobów są dokładnie takie same. Została odfiltrowana jedna z częstotliwości składowych, a faza sygnału została przesunięta.

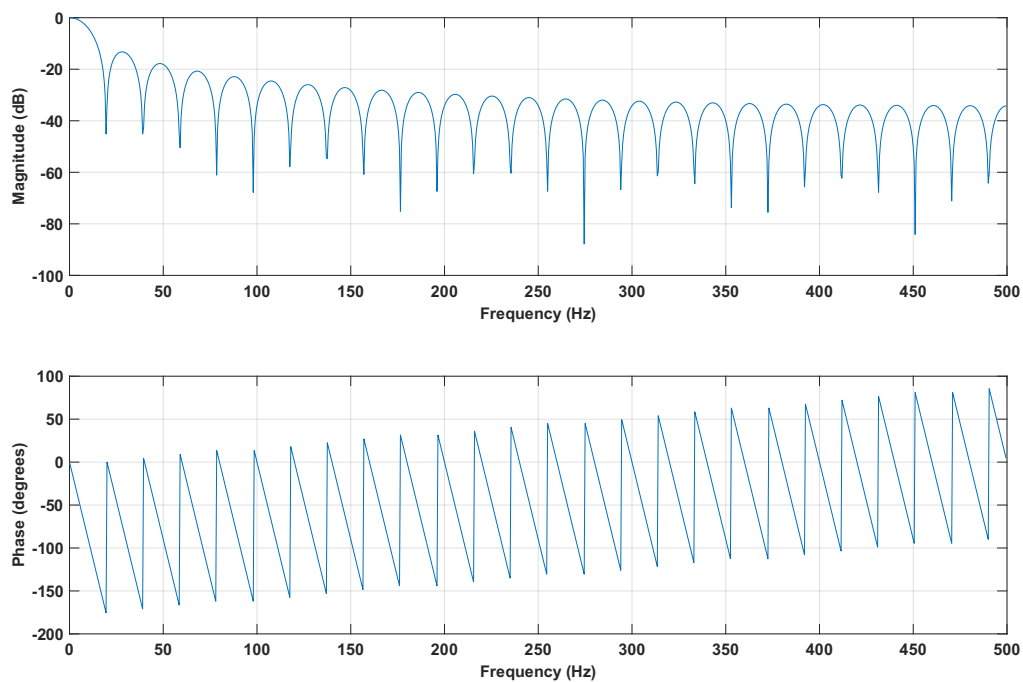
Na widmie częstotliwościowym – **Rysunek 1.2** widoczne jest, iż składowa o częstotliwości 300 Hz została odfiltrowana, a amplituda składowej 10 Hz pozostała niemal bez zmian. Jest to zachowanie zgodne z charakterystyką filtra widoczną na **Rysunku 1.3**. Dla 10 Hz wzmocnienie wynosi -0,64 dB (wartość liniowa równa 0,93), zaś dla 300 Hz -26.65 dB co odpowiada wartości liniowej 0,046.



Rysunek 1.1.1 Porównanie sygnałów czasowych



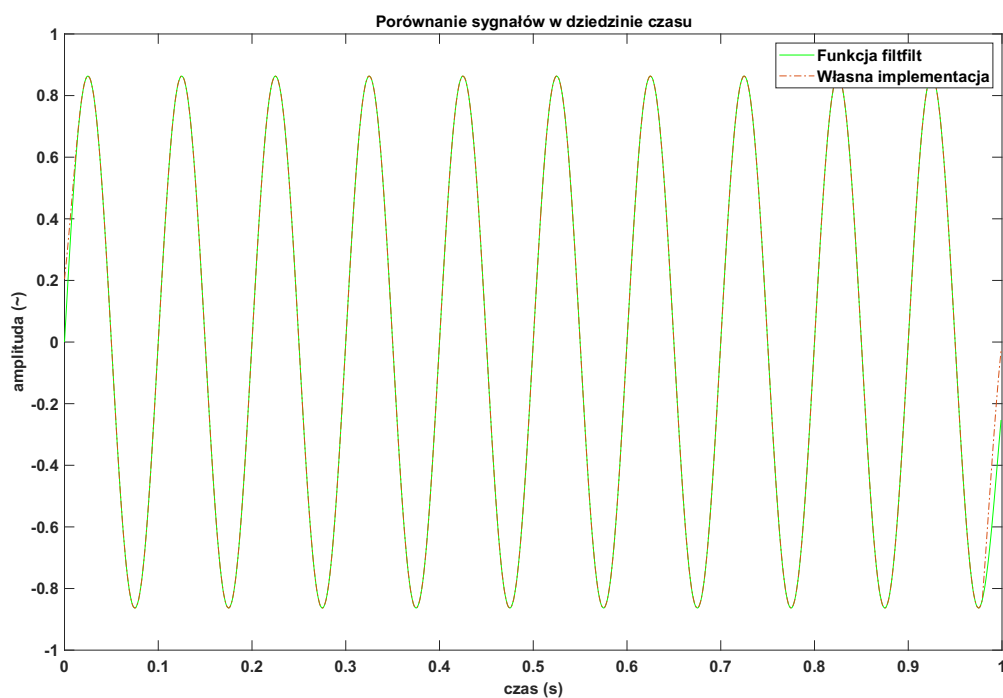
Rysunek 1.1.2 Porównanie widm częstotliwościowych



Rysunek 1.1.3 Charakterystyka filtra

## Zadanie 1.2

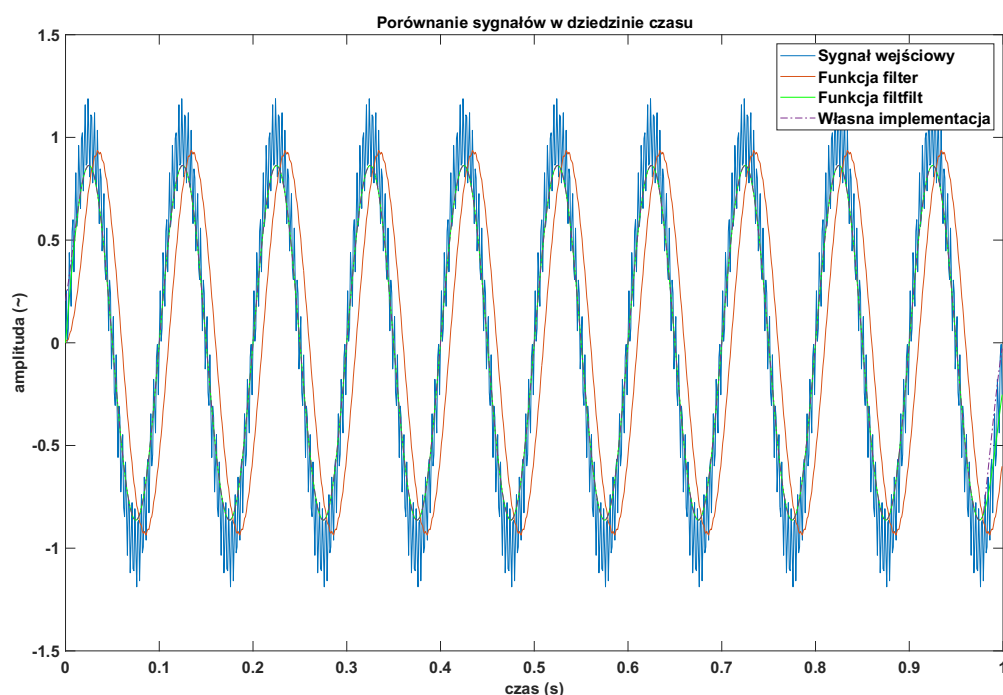
W poprzednim podpunkcie zostało zauważone zjawisko przesunięcia fazy po filtracji sygnału. Aby zapobiec temu zjawisku stosuje się filtrację zero-fazową. Polega ona na filtracji sygnału, odwróceniu w czasie, ponownej filtracji i ponownym odwróceniu w czasie. W ten sposób wprowadzone przesunięcia fazy się równoważą – **Rysunek 1.4**.



Rysunek 1.1.4 Porównanie sygnałów czasowych

Na **Rysunku 1.4** widoczne jest porównanie sygnałów czasowych po filtracji za pomocą funkcji *filtfilt()* oraz własnej implementacji. Można zaobserwować, iż przez większość czasu wykresu pokrywają się. Odstępstwa widoczne są przy początku oraz końcu sygnałów. Wynika to z tego, że funkcja *filtfilt()* minimalizuje tego typu błędy poprzez dostosowanie wartości początkowej i końcowej z sygnałem przed filtracją. Zgadza się to z wykresem na **Rysunku 1.5**. Wartości na początku i końcu sygnału dla funkcji *filtfilt()* pokrywają się z oryginalnym sygnałem, zaś w przypadku własnej implementacji widoczne są rozbieżności.

Zarówno sygnał po filtracji funkcją *filtfilt()* oraz za pomocą własnej implementacji jest w tej samej fazie co sygnał oryginalny. Sygnał po filtracji za pomocą funkcji *filter()* w stosunku do nich jest przesunięty w fazie.



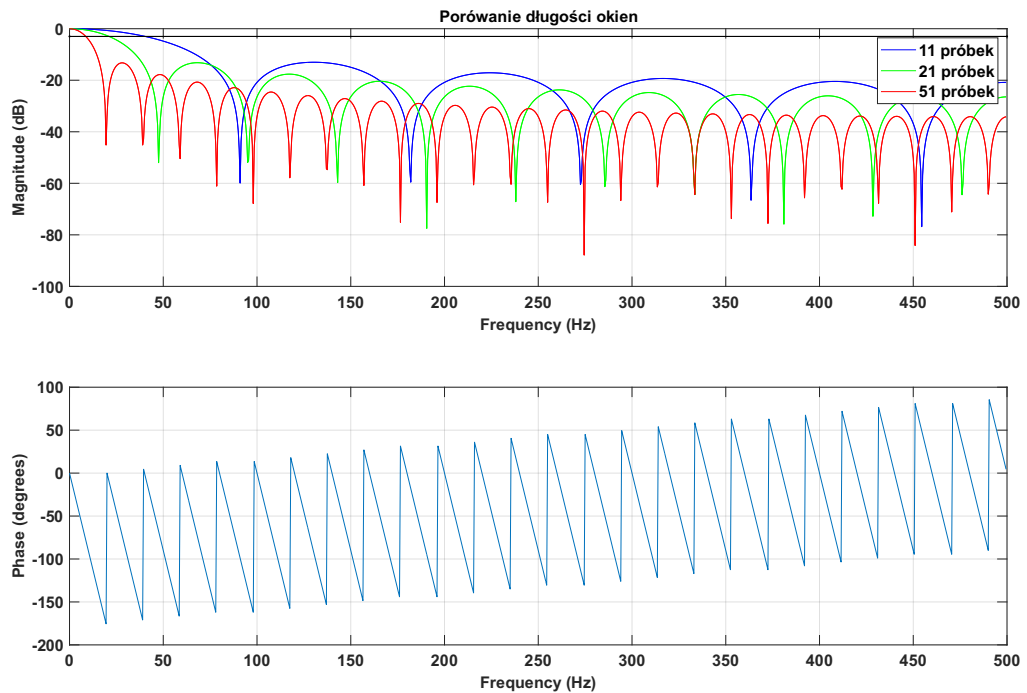
Rysunek 1.1.5 Porównanie sygnałów czasowych

### Zadanie 1.3

Do wykonania powyższego podpunktu wykorzystano okna o różnych amplitudach oraz długościach. Za pomocą funkcji *freqz()* zostały wykreślone ich charakterystyki, które zostały przedstawione na **Rysunku 1.6**. Metodą połowy mocy zostało wyznaczone ich pasmo przenoszenia przedstawione w tabeli:

Długość okna	Częstotliwość graniczna
11 próbek	40,35 Hz
21 próbek	21,08 Hz
51 próbek	8,67 Hz

Widoczne jest zatem, iż wraz z długością okna pasmo przenoszenia filtra zmniejsza się.



Rysunek 1.1.6 Porównanie charakterystyk filtrów

Kod wykorzystany do wykonania zadania 1:

```
clear all
close all

%% Zadanie 1.1
fs = 1000;
N = 1000;
t = 0:1/fs:(N-1)/fs;

Y = sin(2*pi*10*t) + 0.2*sin(2*pi*300*t);

h = zeros(21,1);
h(:) = 1/21; % Okno prostokątne
Yfft = abs(fft(Y))/N*2;
yc = conv(Y,h); % Splot
yc = yc(1:1000);
yf = filter(h,1,Y); % Funkcja filter

figure(1)
plot(t,Y)
hold on
plot(t,yc,'g')
hold on
plot(t,yf,'-.-')
% hold off
xlabel('czas (s)')
ylabel('amplituda (~)')
legend('Sygnał wejściowy','Splot','Funkcja filter')
title('Porównanie sygnałów w dziedzinie czasu')

YCfft = abs(fft(yc))/N*2;
YFfft = abs(fft(yf))/N*2;
```

```

figure(2)
x = linspace(0,fs,N);
plot(x,Yfft)
hold on
plot(x,YCfft,'g')
plot(x,YFfft,'-.')
xlim([0 fs/2])
xlabel('Częstotliwość [Hz]')
ylabel('amplituda (~)')
legend('Sygnał wejściowy','Splot','Funkcja filter')
title('Widma częstotliwości sygnałów')

figure(3)
freqz(h,1,length(yc),fs)
title('Charakterystyka filtra')

%% Zadanie 1.2

yff = filtfilt(h,1,Y);

YF2 = conv(Y,h);
YF2 = YF2(1:1000);
YF2 = YF2(end:-1:1);
YF2 = conv(YF2,h);
YF2 = YF2(1:1000);
YF2 = YF2(end:-1:1);
% YF2 = YF2(1:1000);

figure(4)
plot(t,yff,'g')
hold on
plot(t,YF2,'-.')
xlabel('czas (s)')
ylabel('amplituda (~)')
legend('Funkcja filtfilt','Własna implementacja')
title('Porównanie sygnałów w dziedzinie czasu')

figure(5)
plot(t,Y)
hold on
plot(t,yf)
plot(t,yff,'g')
plot(t,YF2,'-.')
xlabel('czas (s)')
ylabel('amplituda (~)')
legend('Sygnał wejściowy','Funkcja filter','Funkcja filtfilt','Własna implementacja')
title('Porównanie sygnałów w dziedzinie czasu')

%% Zadanie 1.3

% Okno o długości 11 próbek
h11 = zeros(11,1);
h11(:) = 1/11;

% Okno o długości 51 próbek
h51 = zeros(51,1);
h51(:) = 1/51;

figure(6)
freqz(h11,1,length(yc),fs)

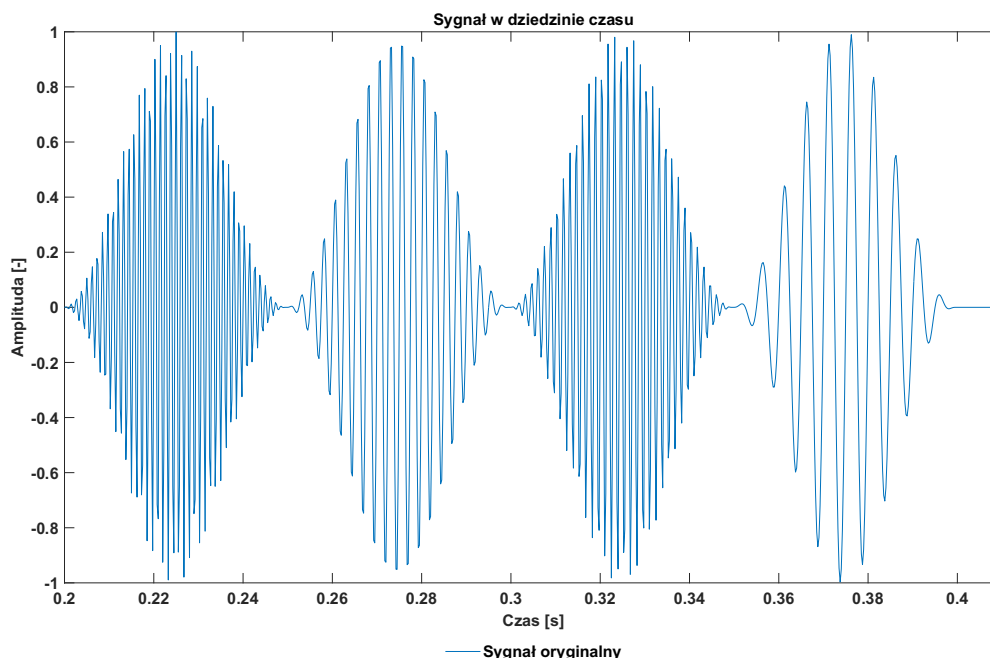
```

```
hold on
freqz(h,1,length(yc),fs)
freqz(h51,1,length(yc),fs)
line([0 500],[-3 -3])
lines = findall(gcf,'type','line');
lines(1).Color = 'black';
lines(2).Color = 'red';
lines(3).Color = 'green';
lines(4).Color = 'blue';

legend('11 próbek','21 próbek','51 próbek')
title('Porównanie długości okien')
```

## Zadanie 2 – Projektowanie filtrów FIR i IIR

W zadaniu przeprowadzono filtrację sygnału składającego się z czterech falek o różnych częstotliwościach chwilowych, które są odseparowane w czasie. Sygnał w dziedzinie czasu zawarto na **Rysunku 2.1**.

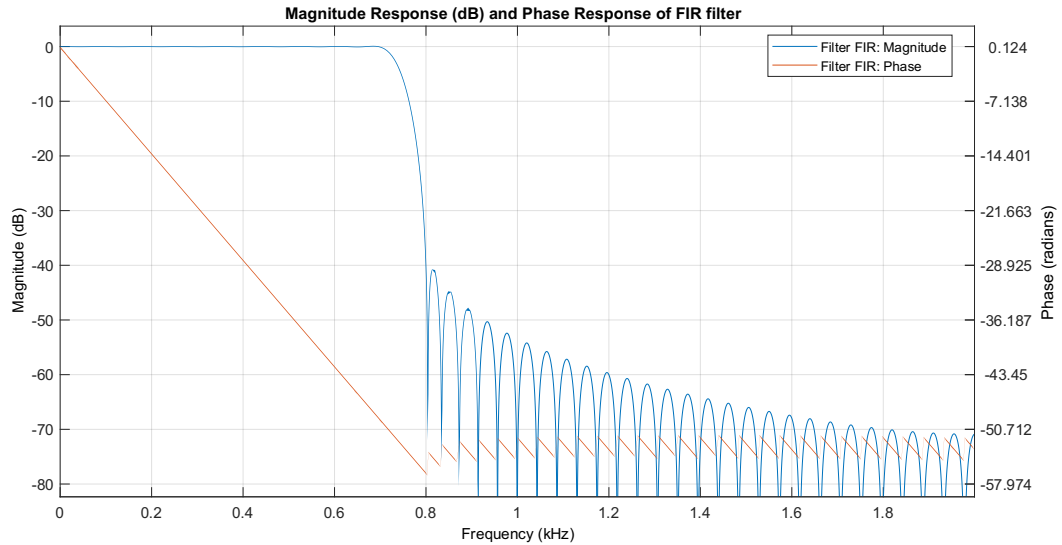


Rysunek 2.1 - Sygnał w dziedzinie czasu

Falka zlokalizowana jako pierwsza w dziedzinie czasu posiada częstotliwość **850 Hz**, kolejna to częstotliwość **400 Hz**, następnie **700 Hz** i ostatecznie falka o najmniejszej częstotliwości chwilowej – **200 Hz**.

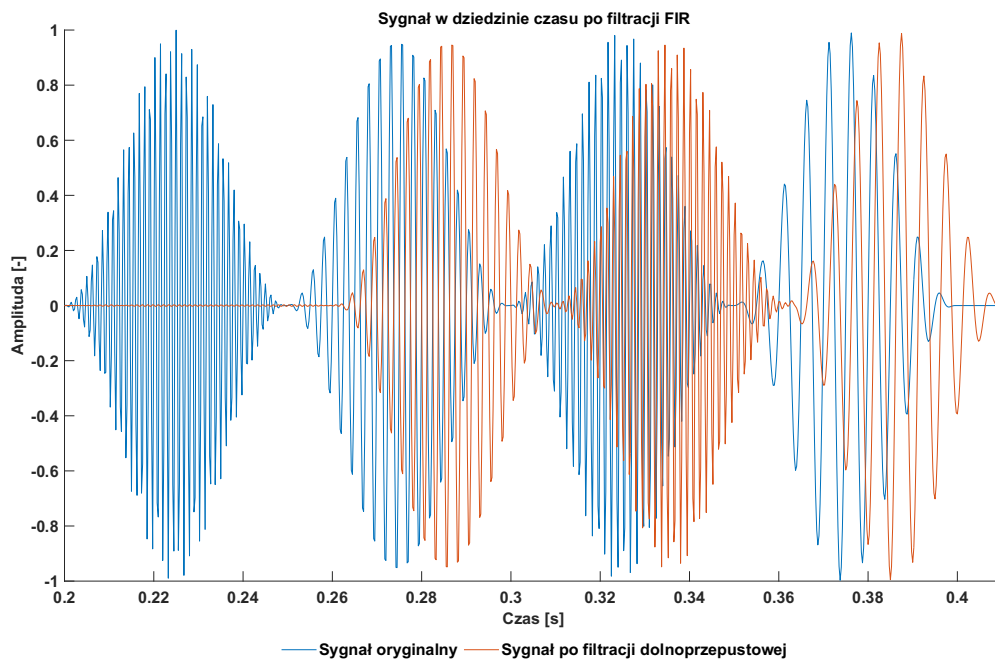
W pierwszej kolejności należało zaprojektować filtr dolnoprzepustowy FIR, aby odfiltrować falkę o częstotliwości środkowej 850 Hz zachowując pozostałe składowe sygnału. W tym celu skorzystano z narzędzia *filterDesigner*. Dobrano parametry: częstotliwości: Fpass 700 Hz oraz Fstop 800 Hz, spadek amplitudy: w paśmie zaporowym (Astop) 40 dB oraz w paśmie przepustowym (Apass) 1dB. Wybrano opcję utworzenia filtra odpowiadającego wymaganiom o minimalnym rzędzie. W ten sposób utworzono filtr FIR o rzędzie 90, który wyeksportowano do pliku o rozszerzeniu mat, a następnie wczytano do przestrzeni roboczej i dokonano filtracji sygnału. Na **Rysunku 2.2** przedstawiono charakterystykę amplitudowo-częstotliwościową i fazowo częstotliwościową filtra, a na **Rysunku 2.3** efekt filtracji sygnału.





Rysunek 2.2 - Charakterystyki częstotliwościowe filtru FIR

Zgodnie ze znaną teorią filtrów cyfrowych, filtr o skończonej odpowiedzi impulsowej wprowadza liniowe przesunięcie fazowe, a więc w efekcie filtracji wszystkie składowe sygnału powinny być opóźnione jednakowo. Pasma przepustowe odpowiada założonym wartościom zadeklarowanym podczas tworzenia filtru w narzędziu.



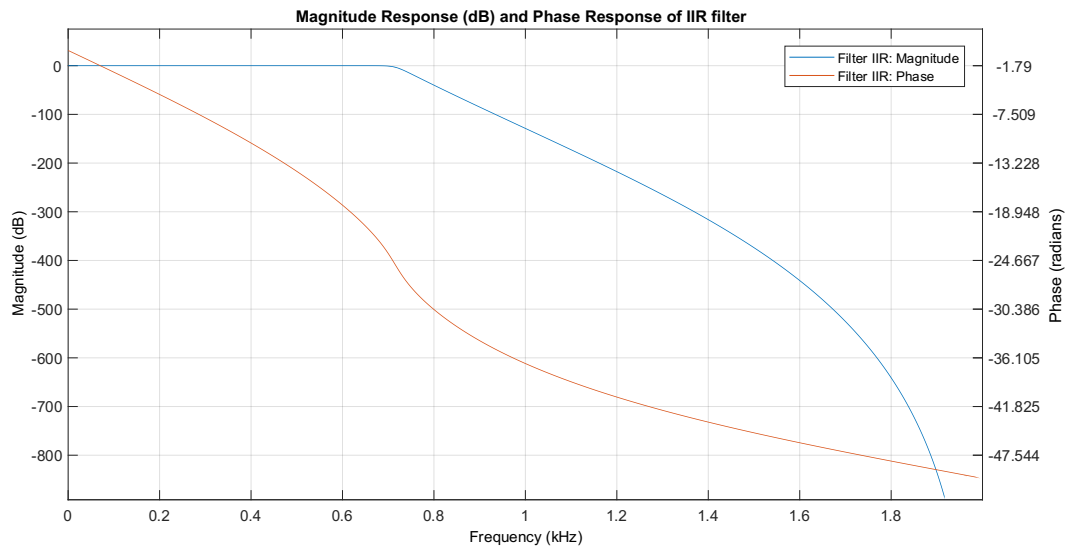
Rysunek 2.3 - Efekt filtracji z użyciem zaprojektowanego filtru FIR

Falka znajdująca się po lewej stronie o częstotliwości 850 Hz została odfiltrowana, a pozostałe falki w sygnale pozostały jednak zmieniła się ich faza i amplituda. Falka o najwyższej częstotliwości, która pozostała w sygnale – 700 Hz zgodnie z **Rysunkiem 2.2** posiada nieznacznie mniejszą amplitudę od sygnału oryginalnego. Wszystkie 3 falki są przesunięte w fazie względem oryginalnego sygnału jednakowo, co odpowiada charakterystyce filtru FIR – przesunięcie fazowe rośnie ze wzrostem

częstotliwości, a więc wszystkie składowe na wykresie w dziedzinie czasu są opóźnione jednakowo. Zgodnie z charakterystyką częstotliwościowo-amplitudową występują drobne fluktuacje amplitudy w miejscu sygnału gdzie znajdowała się falka o częstotliwości 850 Hz. Poniżej zamieszczono kod, który wykorzystano do uzyskania wykresów.

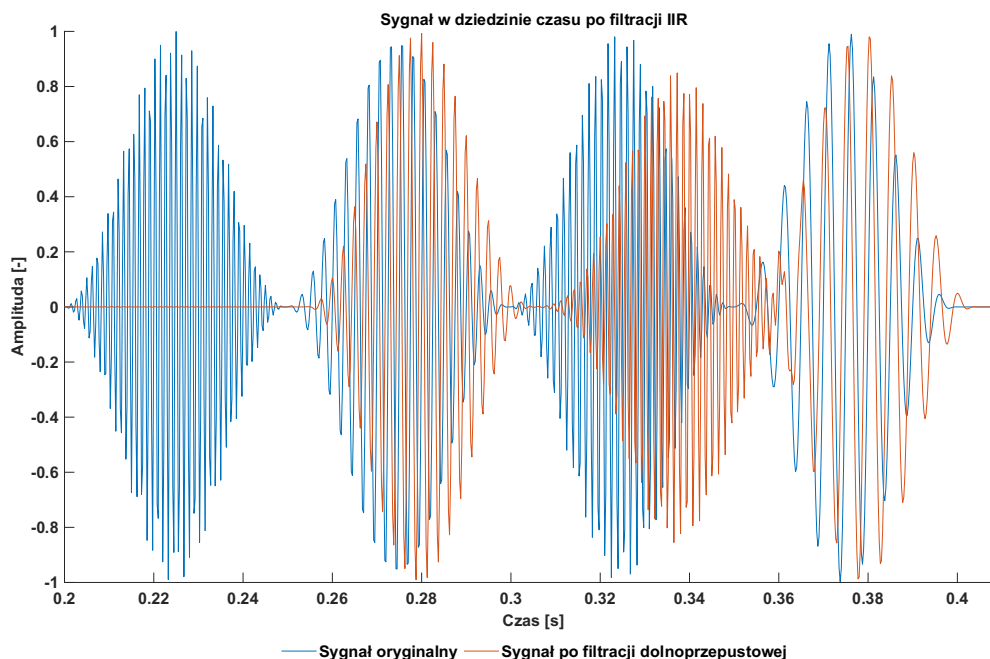
```
clear all
close all
clc
%% Tworzenie sygnału - kod z instrukcji
% częstotliwość próbkowania
fs = 4000;
% max. czas sygnału
tmax = 1-1/fs;
% wektor czasu
t = 0:1/fs:tmax;
% długość sygnału czasowego
s = length(t);
% wektor częstotliwości
f = linspace(0,fs-fs/s,s);
% częstotliwości środkowe falek
w = [200,400,700,850];
% okno hanna, 200 próbek
mask = hann(200);
% lokalizacje falek na wektorze czasu
inds = [1400,1000,1200,800];
% deklaracja wektora sygnału y
y = zeros(length(t),1);
% definicja kolejnych falek
for i = 1:length(w)
m = zeros(s,1);
m(inds(i):inds(i)+length(mask)-1) = mask;
y_temp = sin(2*pi*t*w(i)).*m.';
y = y+y_temp.';
end
%% Część właściwa
figure(1)
plot(t,y)
xlim([0.2,0.41]);
xlabel('Czas [s]');
ylabel('Amplituda [-]');
legend('Sygnał oryginalny','location','southoutside','orientation',...
'horizontal','box','off')
title('Sygnał w dziedzinie czasu')
%% Zadanie 2.1
load('Filtr2_1.mat');
yf=filter(Hd,y);
figure(2)
hold on
plot(t,y)
plot(t,yf)
hold off
xlim([0.2,0.41]);
xlabel('Czas [s]');
ylabel('Amplituda [-]');
legend('Sygnał oryginalny','Sygnał po filtracji
dolnoprzepustowej','location','southoutside','orientation',...
'horizontal','box','off')
title('Sygnał w dziedzinie czasu po filtracji FIR')
figure(3)
freqz(Hd)
```

W kolejnej części zadania ponownie skorzystano z narzędzia *filterDesigner* aby utworzyć filtr o jednakowych parametrach, jednak tym razem o nieskończonej odpowiedzi impulsowej – IIR. Poniżej podobnie jak dla filtru FIR przedstawiono charakterystyki częstotliwościowo-amplitudową i częstotliwościowo-fazową filtru oraz efekt filtracji sygnału: **Rysunek 2.4, 2.5**.



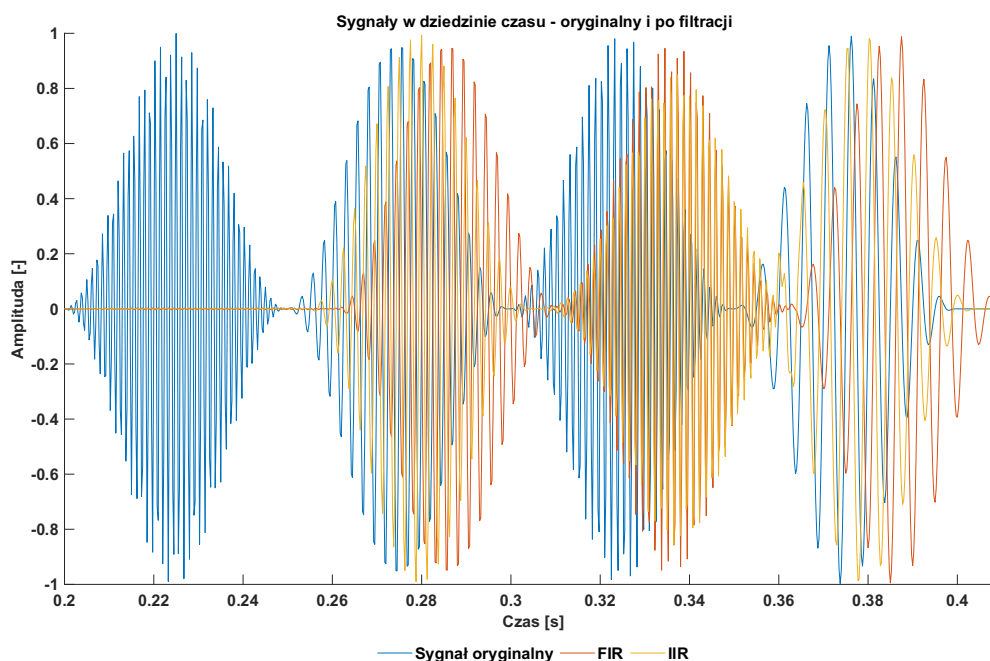
Rysunek 2.4 - Charakterystyki częstotliwościowe filtru IIR

Na **Rysunku 2.4** można zaobserwować, że faza nie zmienia się liniowo jak to miało miejsce w przypadku filtru FIR, a spadek amplitudy osiąga wartości rzędu -1000 dB dla częstotliwości zbliżającej się do częstotliwości Nyquista. Można zatem spodziewać się, że nie wszystkie składowe o różnych częstotliwościach będą jednakowo przesunięte w czasie, oraz że amplituda falki o częstotliwości 700 Hz będzie mniejsza niż oryginalnego sygnału ze względu na ostre zbocze charakterystyki częstotliwościowo-amplitudowej. Warto zwrócić uwagę na rząd filtru, który wynosi 32 i jest prawie 3 razy mniejszy od filtru FIR o takich samych parametrach. Ukazuje to, że filtry IIR są bardziej wydajne obliczeniowo od filtrów FIR. Dodatkowo można zauważyć w narzędziu *filterDesigner*, że filtr jest stabilny, a nie zawsze musi tak być, ze względu na występujące sprzężenie zwrotne. Oba utworzone filtry w plikach o rozszerzeniu *fd*a zostaną spakowane i przesłane wraz ze sprawozdaniem.



Rysunek 2.5 - Efekt filtracji z użyciem zaprojektowanego filtra IIR

W efekcie filtracji, z sygnału usunięta została falka o częstotliwości 850 Hz, a pozostałe ponownie zostały zmodyfikowane pod względem fazy i amplitudy. Falka o częstotliwości 700 Hz posiada zmniejszoną amplitudę zgodnie z **Rysunkiem 2.4**, a amplituda pozostałych składowych pozostała bez zmian. Ze względu na nieliniową zmianę fazy, można zauważyć, że falka o najwyższej częstotliwości w sygnale po filtracji posiada większe przesunięcie względem oryginalnego sygnału niż pozostałe falki, które posiadają podobne przesunięcie ze względu na fakt, że w zakresie ich częstotliwości na **Rysunku 2.4** zmiana fazy jest prawie liniowa. W celu porównania dwóch filtrów naniesiono efekty filtracji względem oryginalnego sygnału obydwu filtrów na jeden wykres – **Rysunek 2.6**.



Rysunek 2.6 - Porównanie efektów filtracji

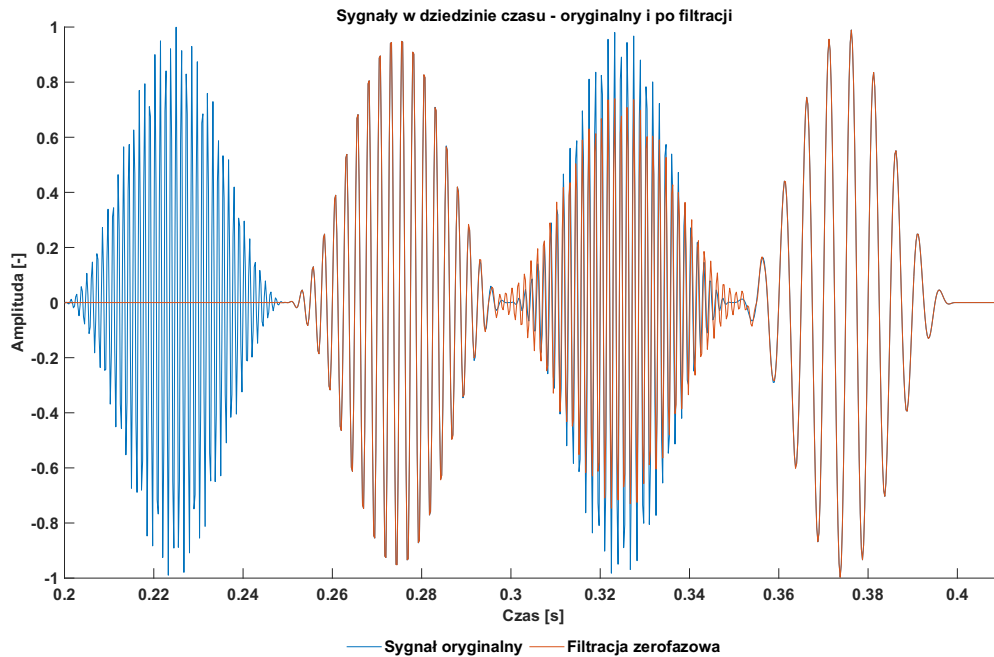
Dla falki o częstotliwości 700 Hz, przesunięcie fazowe jest podobne, jednak filtr IIR spowodował większe zniekształcenie amplitudy. Dla pozostałych falek amplituda jest zbliżona, jednak przy filtracji IIR przesunięcie fazowe jest mniejsze, ale nie stałe dla wszystkich składowych.

Podsumowując, filtry FIR posiadają liniową fazę, mniej zniekształcają amplitudę, jednak ze względu na wyższy rząd, a więc więcej współczynników są mniej wydajne i wymagają większej mocy obliczeniowej. Filtry IIR, posiadają mniejszy rząd dla takich samych parametrów, a więc są wydajniejsze obliczeniowo, jednak posiadają nieliniową fazę, która skutkuje w różnych przesunięciach komponentów o różnych częstotliwościach oraz ze względu na pętlę sprzężenia zwrotnego (wykorzystywanie próbek z sygnału wejściowego i wyjściowego) mogą być niestabilne. Poniżej zawarto kod wykorzystany do wykonania wykresów:

```
%% Zadanie 2.2
load('Filtr2_2.mat');
yf_IIR=filter(Hd_2_2,y);
figure(3)
hold on
plot(t,y)
plot(t,yf_IIR)
hold off
xlim([0.2,0.41]);
xlabel('Czas [s]');
ylabel('Amplituda [-]');
legend('Sygnał oryginalny','Sygnał po filtracji dolnoprzepustowej','location','southoutside','orientation',... 'horizontal','box','off')
title('Sygnał w dziedzinie czasu po filtracji IIR')
figure(4)
freqz(Hd)

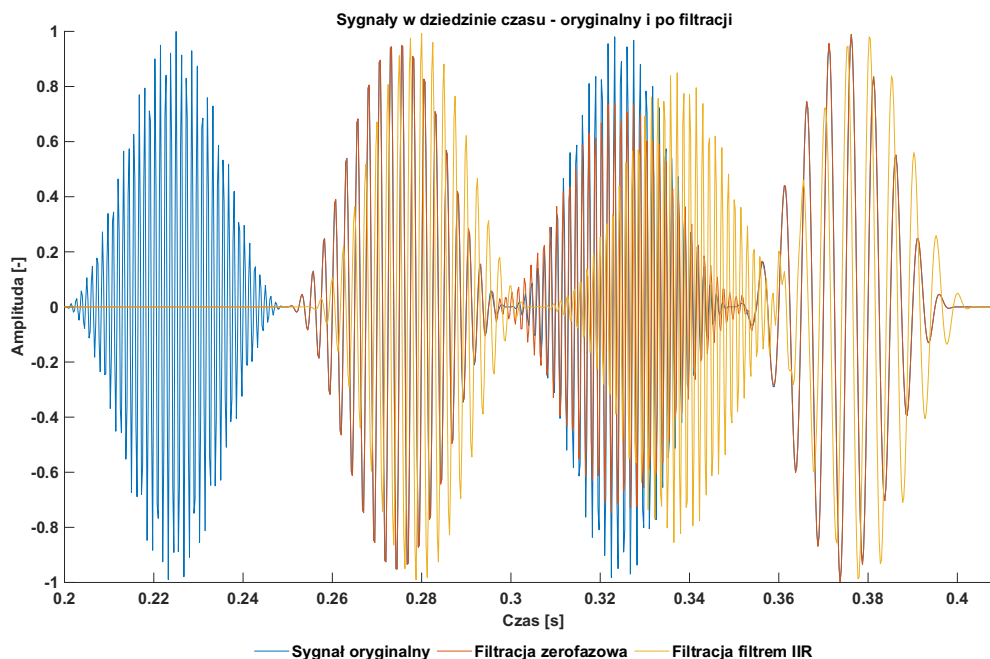
%% Złożenie
figure(5)
hold on
plot(t,y)
plot(t,yf)
plot(t,yf_IIR)
hold off
xlim([0.2,0.41]);
xlabel('Czas [s]');
ylabel('Amplituda [-]');
legend('Sygnał oryginalny','FIR','IIR','location','southoutside','orientation',... 'horizontal','box','off')
title('Sygnały w dziedzinie czasu - oryginalny i po filtracji')
```

W ostatniej części zadania przeprowadzono filtrację zero fazową z wykorzystaniem filtra IIR. Wyniki filtracji przedstawiono na **Rysunku 2.7**.



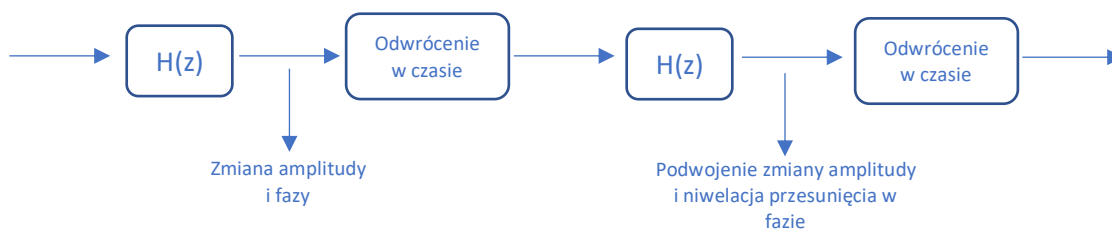
*Rysunek 2.7 - Filtracja zero fazowa z użyciem filtra IIR*

Poprawnie odfiltrowana została falka o częstotliwości 850 Hz. Zgodnie z charakterystyką filtra amplituda w tym miejscu jest w przybliżeniu równa zero i nie zmienia się jak to miało miejsce w przypadku filtra FIR. Pozostałe falki pod względem fazy są zgodne z oryginalnym sygnałem co jest charakterystycznym efektem w przypadku filtracji zero fazowej. Jednak ze względu na podwójną filtrację dzięki której niwelują się przesunięcie w fazie, podwaja się efekt zniekształcenia amplitudy. Patrząc na charakterystykę z **Rysunku 2.4** widać, że następuje zmiana amplitudy dla częstotliwości 700 Hz. Porównując do filtracji IIR przeprowadzonej w poprzedniej części zadania – **Rysunek 2.8** dobrze widać, różnicę w poziomie amplitudy dla tej falki.



Rysunek 2.8 - Porównanie filtracji IRR zero fazowej i zwykłej

Spoglądając na różnicę amplitudy dla falki o częstotliwości 700 Hz widać opisany wcześniej efekt podwójnego zniekształcenia amplitudy. Dodatkowo zaobserwować można, jak filtracja zero fazowa pozwoliła zachować oryginalną fazę sygnału, a pojedyncze zastosowanie filtra IIR zmieniło fazę. Przedstawiono to schematycznie na **Rysunku 2.9**.



Rysunek 2.9 - Schemat filtracji zero fazowej

Poniżej zawarto kod wykorzystany do wykonania ostatniej części zadania.

```
%% Filtracja zerofazowa
load('Filtr_2_2_wspol.mat');
yfb=filtfilt(SOS,G,y);
figure(6)
hold on
plot(t,y)
plot(t,yfb)
hold off
xlim([0.2,0.41]);
xlabel('Czas [s]');
ylabel('Amplituda [-]');
legend('Sygnał oryginalny','Filtracja
zerofazowa','location','southoutside','orientation',...
'horizontal','box','off')
title('Sygnały w dziedzinie czasu - oryginalny i po filtracji')
```

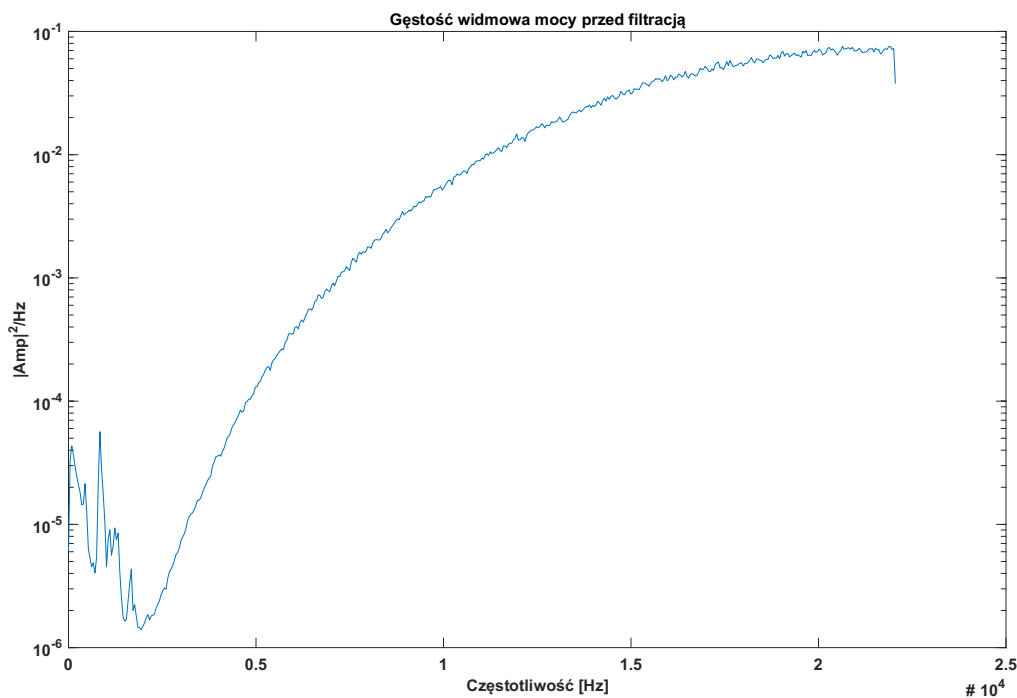
```

%% Złożenie z IIR
figure(6)
hold on
plot(t,y)
plot(t,yfz)
plot(t,y_IIR)
hold off
xlim([0.2,0.41]);
xlabel('Czas [s]');
ylabel('Amplituda [-]');
legend('Sygnał oryginalny','Filtracja zerofazowa','Filtracja z
wykorzystaniem filtru IIR','location','southoutside','orientation',...
'horizontal','box','off')
title('Sygnały w dziedzinie czasu - oryginalny i po filtracji')

```

### Zadanie 3 – Filtracja szumu

Pierwszym krokiem przed zaprojektowaniem filtra było wykreślenie gęstości widmowej mocy sygnału – **Rysunek 3.1**. Widoczne jest, iż od ok. 1800 Hz składowe sygnału narastają logarytmicznie co może świadczyć o występowaniu dodanego szumu.



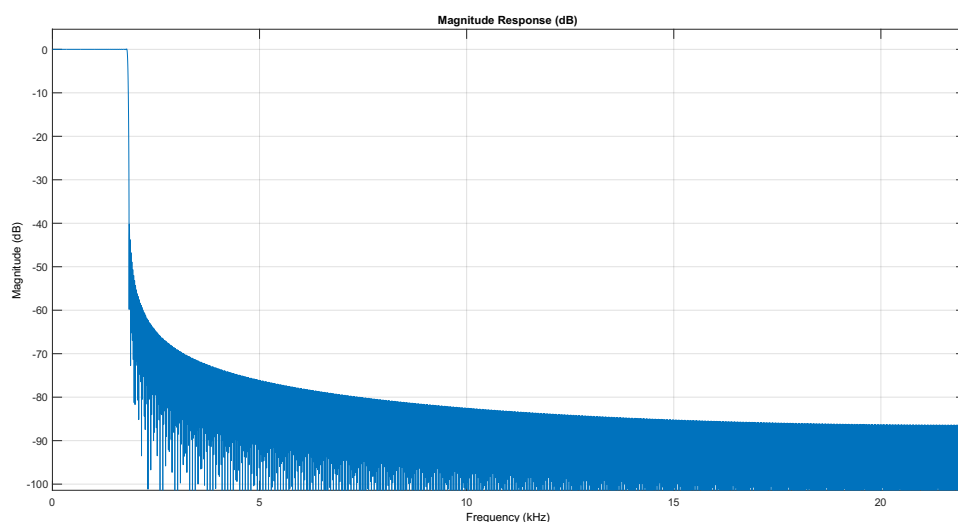
Rysunek 3.3.1 Gęstość widmowa mocy kanału pierwszego sygnału przed filtracją

Do jego odfiltrowania został wykorzystany filtr o parametrach widocznych na **Rysunku 3.2** oraz charakterystyce widocznej na **Rysunku 3.3**. Jest to dolnoprzepustowy filtr FIR, którego parametry odpowiadają odczytanym z **Rysunku 3.1** częstotliwością.



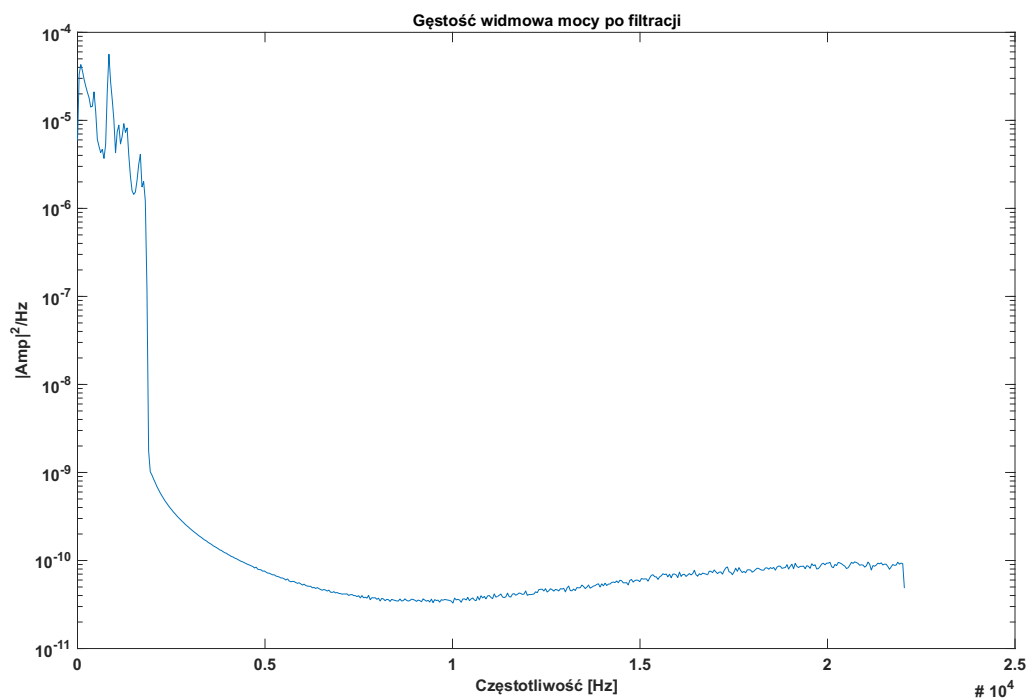
<b>Response Type</b> <input checked="" type="radio"/> Lowpass <input type="radio"/> Highpass <input type="radio"/> Bandpass <input type="radio"/> Bandstop <input type="radio"/> Differentiator <b>Design Method</b> <input type="radio"/> IIR Butterworth <input checked="" type="radio"/> FIR Window	<b>Filter Order</b> <input type="radio"/> Specify order: 10 <input checked="" type="radio"/> Minimum order <b>Options</b> <input checked="" type="checkbox"/> Scale Passband Window: Kaiser	<b>Frequency Specifications</b> Units: Hz Fs: 44100 Fpass: 1800 Fstop: 1850	<b>Magnitude Specifications</b> Units: dB Apass: 1 Astop: 40
--	--	---	---

Rysunek 3.3.2 Parametry filtru

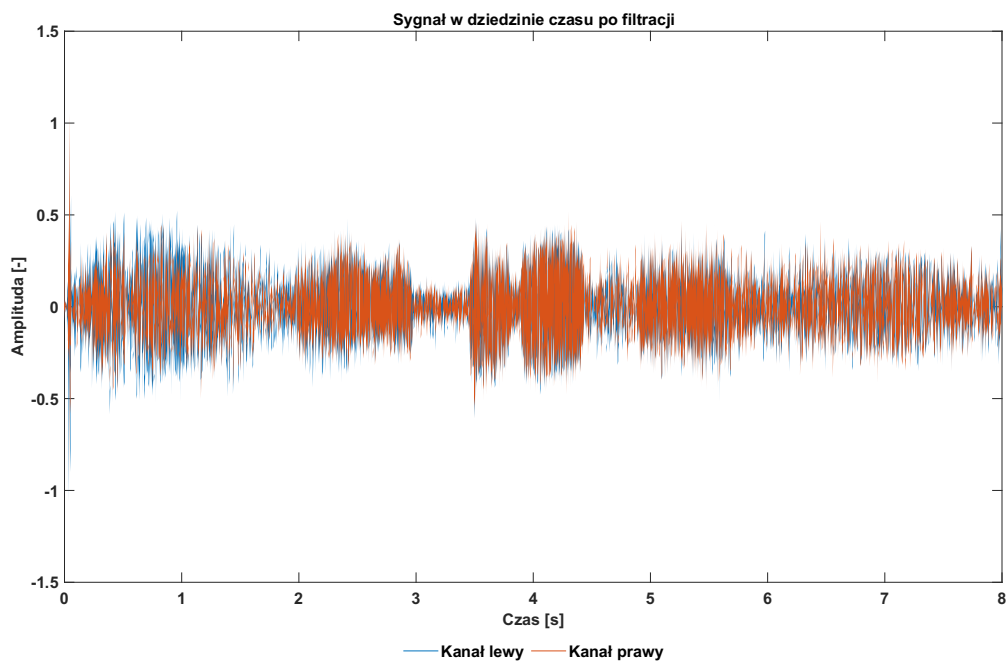


Rysunek 3.3.3 Charakterystyka filtru

Po zastosowaniu filtracji wykreślono ponownie PSD – **Rysunek 3.4**, oraz sygnał dźwiękowy w dziedzinie czasu – **Rysunek 3.5**.



Rysunek 3.3.4 Gęstość widmowa mocy kanału pierwszego sygnału po filtracji



Rysunek 3.5 - Sygnał dźwiękowy w dziedzinie czasu po filtracji

Po zastosowaniu odsłuchaniu sygnału słyszalna jest kwestia z filmu Władca Pierścieni: „You shall not pass” wypowiedziana przez postać graną przez Iana McKellena.

Kod wykorzystany do wykonania zadania:

```
clear all
close all
load('sound.mat')

% Wczytanie pliku
[data,fs] = audioread(['ktoto.wav']);
s = length(data);
yf = abs(fft(data(:,1)))/length(s)*2;
x = linspace(0,fs,s);

[Pxx,F] = pwelch(data(:,1),1000,1000/2,1000,fs);
figure(1)
semilogy(F,Pxx)
title('Gęstość widmowa mocy przed filtracją')
xlabel('Częstotliwość [Hz]')
ylabel('|Amp|^2/Hz')

% Num - współczynniki filtru FIR
dataF(:,1) = filter(Num,1,squeeze(data(:,1)));
dataF(:,2) = filter(Num,1,squeeze(data(:,2)));

sound(dataF, fs)

[Pxx2,F2] = pwelch(dataF(:,1),1000,1000/2,1000,fs);
figure(2)
```

```
semilogy(F2,Pxx2)
title('Gęstość widmowa mocy po filtracji')
xlabel('Częstotliwość [Hz]')
ylabel('|Amp|^2/Hz')
```