



Filtry cyfrowe w pakiecie Matlab

IMPELEMENTACJA, ZASTOSOWANIE, ANALIZA.
RAPORT KOŃCOWY

Mateusz Oleksy, Paweł Urban | Techniki Obliczeniowe | 21.01.2024

Filtry cyfrowe – wprowadzenie

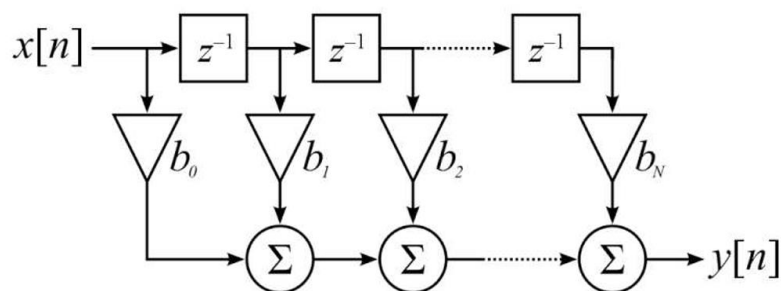
Wszelkiej maści filtry znajdują szerokie zastosowanie w dzisiejszym świecie. Od telekomunikacji aż po muzykę. Na przykład gdy musimy przesłać jakiś sygnał i tym celu używamy modulacji, a przed i po tym procesie używamy filtrów dolno-przepustowego i pasmowo-zaporowego, bądź o innej charakterystyce. Podobnie postępujemy przy procesie odwrotnym- demodulacji. W tym raporcie omówimy co to dokładnie filtr cyfrowy oraz pokażemy jego implementację. Po co używamy filtry cyfrowe?

- Poprawa sygnału poprzez usunięcie szumów poprzez odfiltrowanie częstotliwości z widma sygnału.
- Wyodrębnianie danych częstotliwości z sygnału.
- Przygotowanie sygnału do transmisji.

Filtr cyfrowy to program komputerowy, mający na celu zrealizować to samo zadanie co filtr analogowy jednakże na sygnale w postaci cyfrowej, czyli zmiana widma sygnału. Rozróżniamy dwa rodzaje filtrów cyfrowych typu FIR (Finite impulse response) oraz IIR (Infinite impulse response). Odpowiedź filtru FIR na pobudzenie o skończonej długości również jest skończona. Inaczej mówimy, że filtr ma skończoną odpowiedź impulsową. Natomiast w IIR, w odpowiedzi na skończone pobudzenie otrzymujemy nieskończoną odpowiedź- filtr ma nieskończoną odpowiedź impulsową. W rozważanym projekcie zaimplementowaliśmy filtr typu FIR ponieważ nie realizujemy aktywnego sprzężenia zwrotnego (jak w przypadku IIR), FIR ma większą stabilność oraz mniejszy poziom skomplikowania. Pokażemy przydatność takiego filtru na sygnale cyfrowym, nagrany mikrofonem i wygenerowanym sztucznie.

ZASADA DZIAŁANIA FILTRU FIR

Schemat blokowy przedstawiający filtr FIR wygląda następująco:



Na rysunku powyżej przedstawiono na czym polega filtr FIR. W dziedzinie zmiennej zespolonej z^{-1} odpowiada opóźnieniu sygnału o jedną próbkę. Stąd bierzmy z^{-1} przemnażamy razy n-ty współczynnik b i podajemy do blocka sumacyjnego. Ostatecznie otrzymujemy na wyjściu sumę przekształconych N próbek.

Przypomina to coś w rodzaju filtru o średniej ruchomej (moving average filter) tyle, że zamiast stałych współczynników $1/N$, obliczamy współczynniki z odwrotnej

transformaty fouriera. Warto wspomnieć o tym, że w filtrze FIR ważna jest długość jak i rząd filtru. Rząd filtru oznacza najwyższą potęgę w transmitancji, z kolei długość to ilość współczynników w odpowiedzi impulsowej. Im większa długość filtru, tym jest on dokładniejszy. W naszym projekcie długość jest nieparzysta, tak aby nie pojawiały się problemy z niesymetrycznością.

Odpowiedź impulsowa filtru FIR

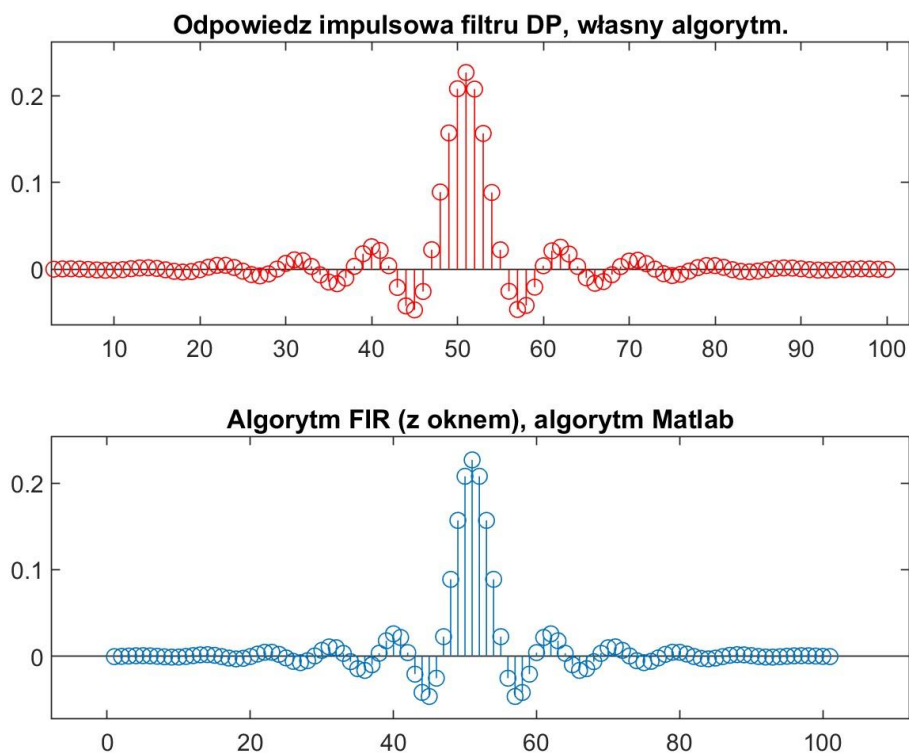
Odpowiedź impulsowa filtru FIR, czyli współczynniki b . Większość filtrów operuje w dziedzinie częstotliwości. W związku z tym trzeba by obliczyć transformatę fouriera odpowiedzi impulsowej i następnie kolejną FFT (fast fourier transform) od sygnału pierwotnego i potem odwrotną transformatę fouriera od sygnału przetworzonego. W związku z tym w naszej implementacji operujemy w dziedzinie czasu i użyliśmy splotu do przemnożenia współczynników i próbek filtru. Dodatkowo wszystkie współczynniki mnożymy razy okno Hamminga o długości odpowiadającej N . Do obliczenia odpowiedzi impulsowej filtru dolnoprzepustowego posłużył nam wzór:

$$DP = \sin(2 \cdot \pi \cdot n \cdot f_g / f_s) / (n \cdot \pi)$$

f_g – częstotliwość górna (odcięcia)

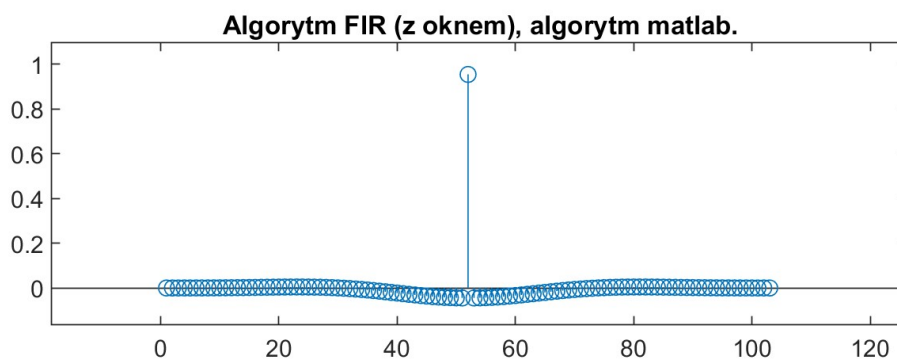
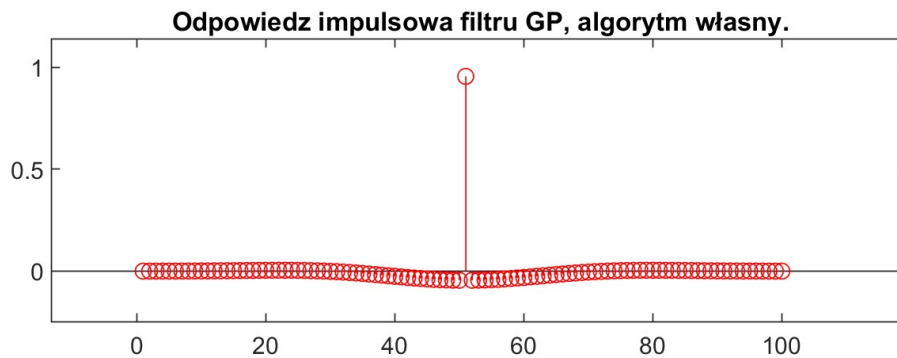
f_s – częstotliwość próbkowania

n – numer danego współczynnika, od 1 do N



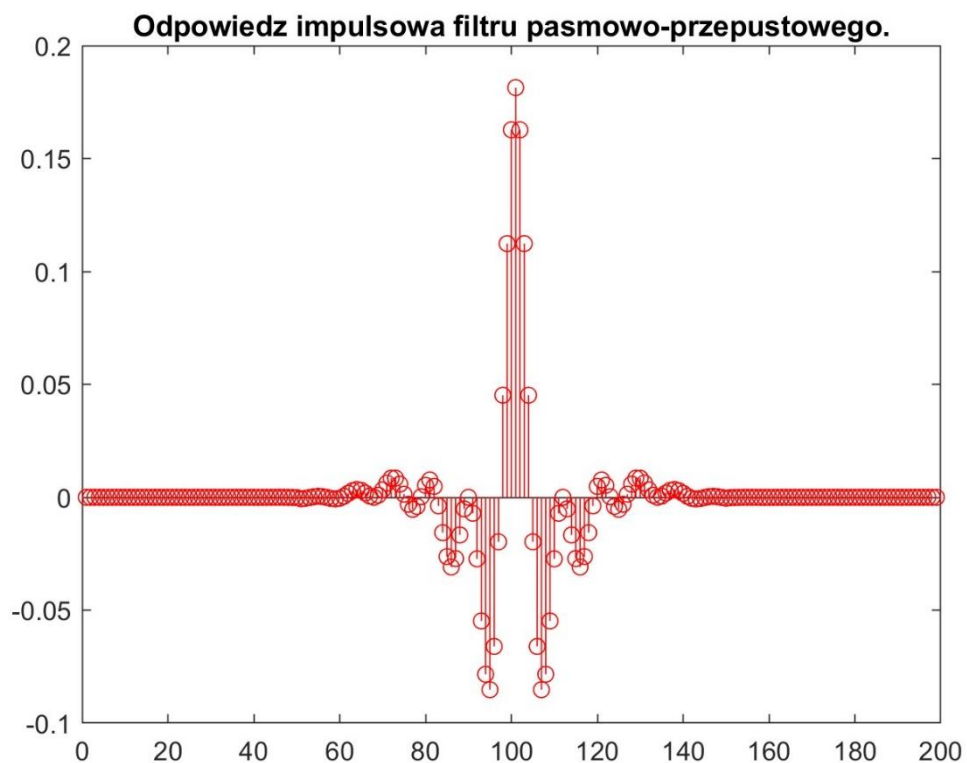
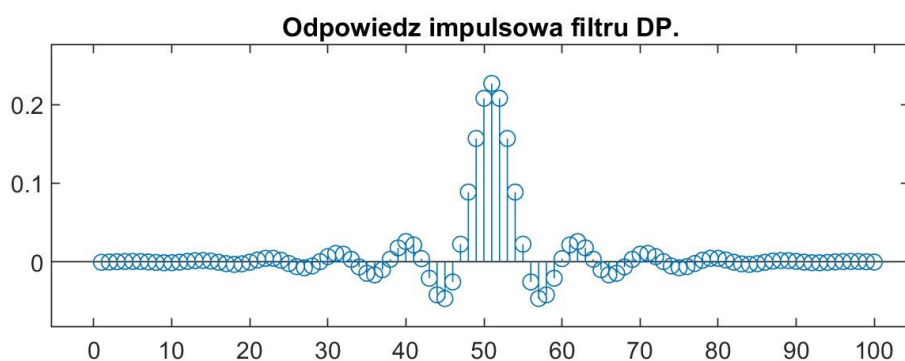
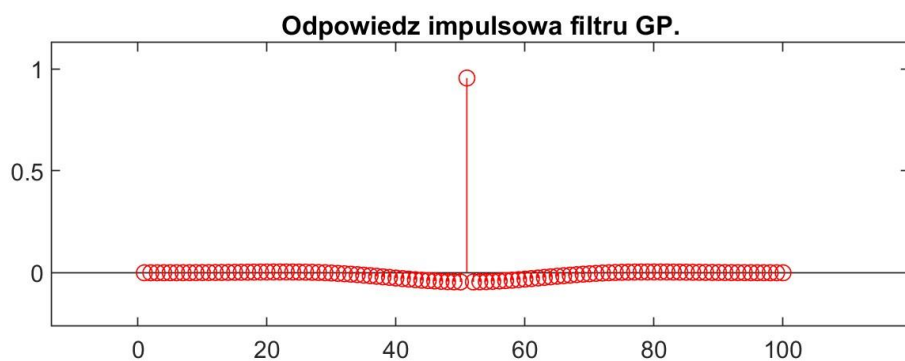
1. Porównanie własnej implementacji odpowiedzi impulsowej dla filtra dolnoprzepustowego vs algorytm Matlab, gdzie $N=101$, $f_g = 5000$ Hz

Dla filtra górnoprzepustowego, można posłużyć się wzorem $GP(t) = 1 - DP(t)$.



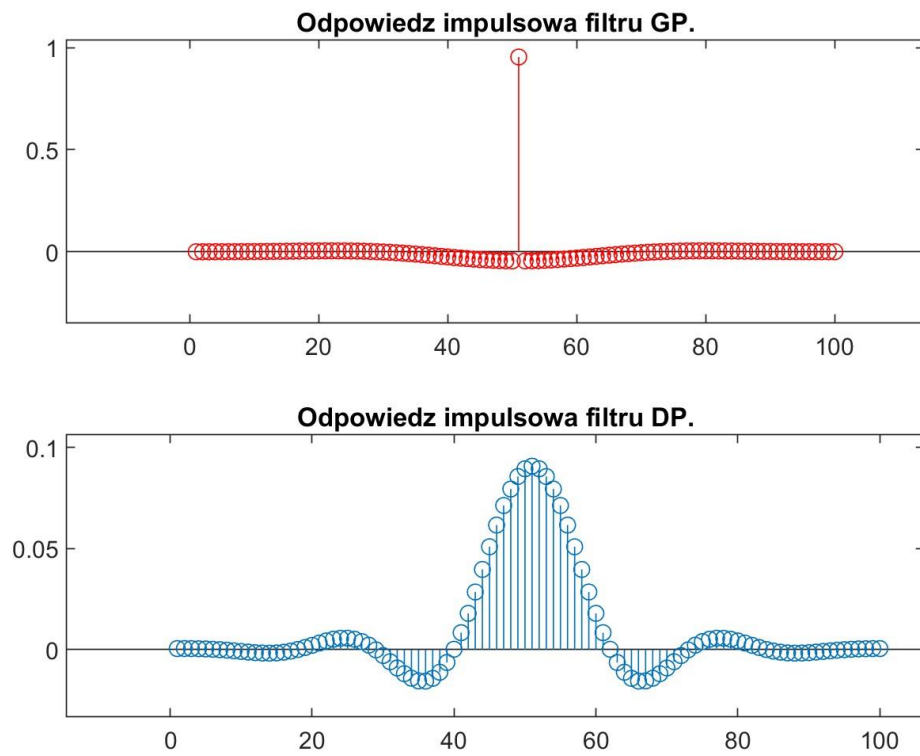
2. Porównanie własnej implementacji odpowiedzi impulsowej dla filtra górnoprzepustowego i algorytmu Matlab, gdzie $N=101$, $f_d = 1000$ Hz

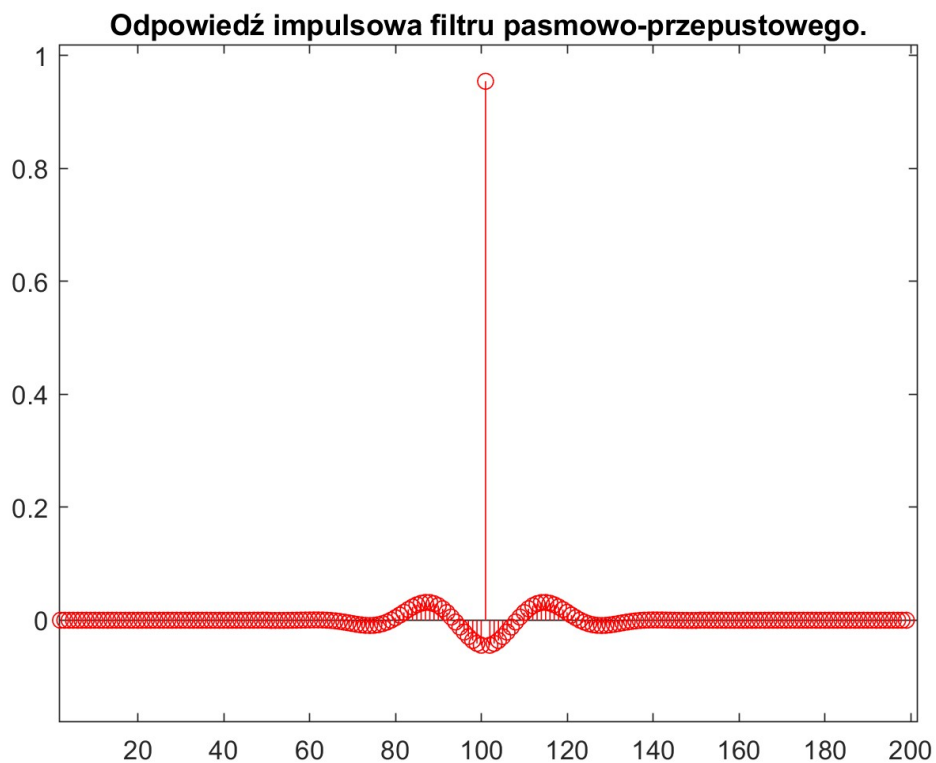
Filtr pasmowo-przepustowy można uzyskać z wzoru $DP(f)*GP(f)$. Choć gdy operujemy w dziedzinie czasu to mnożenie zastępujemy splotem.



3. Porównanie filtrów o zadanych własnościach, następnie odpowiedź impulsowa dla filtru pasmowo-przepustowego, gdzie $N=101$, $f_d = 1000$ Hz, $f_g = 5000$ Hz

Filtr pasmo-zaporowy to $1-PP(t)$, czyli pasmowo-zaporowy można uzyskać poprzez odjęcie od 1 filtru pasmo-przepustowego.





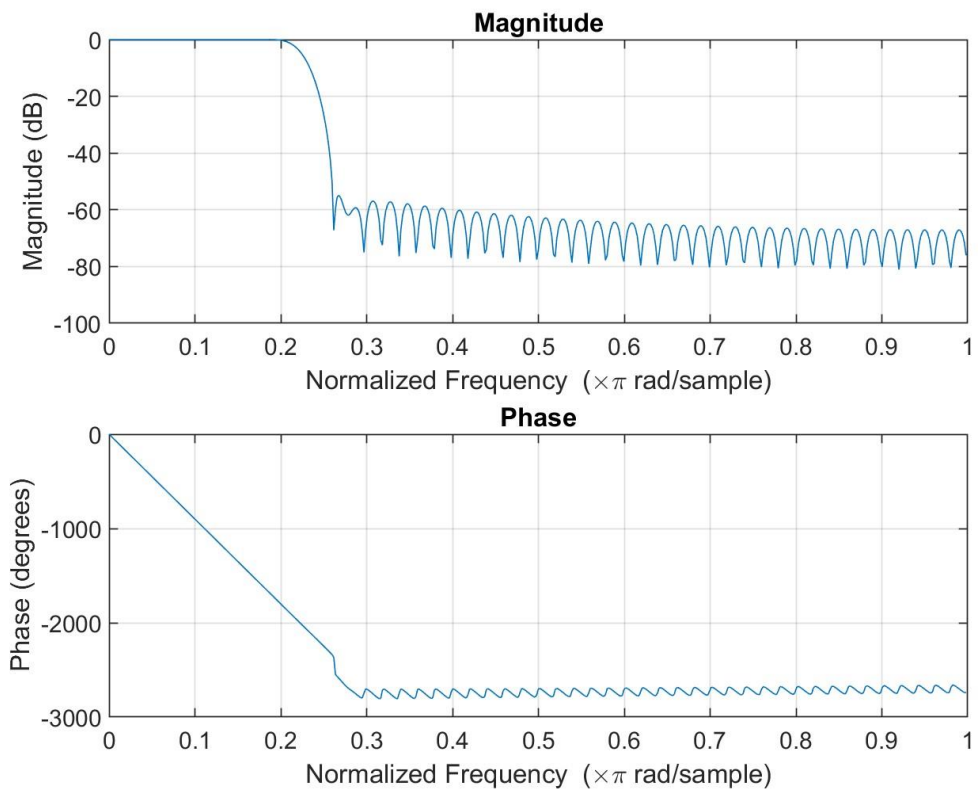
4. Porównanie filtrów o zadanych parametrach, następnie odpowiedź impulsowa dla filtru pasmowo-przepustowego, gdzie $N=101$, $f_d = 1000$ Hz, $f_g = 2000$ Hz

Charakterystyk fazowej i częstotliwościowej filtru FIR

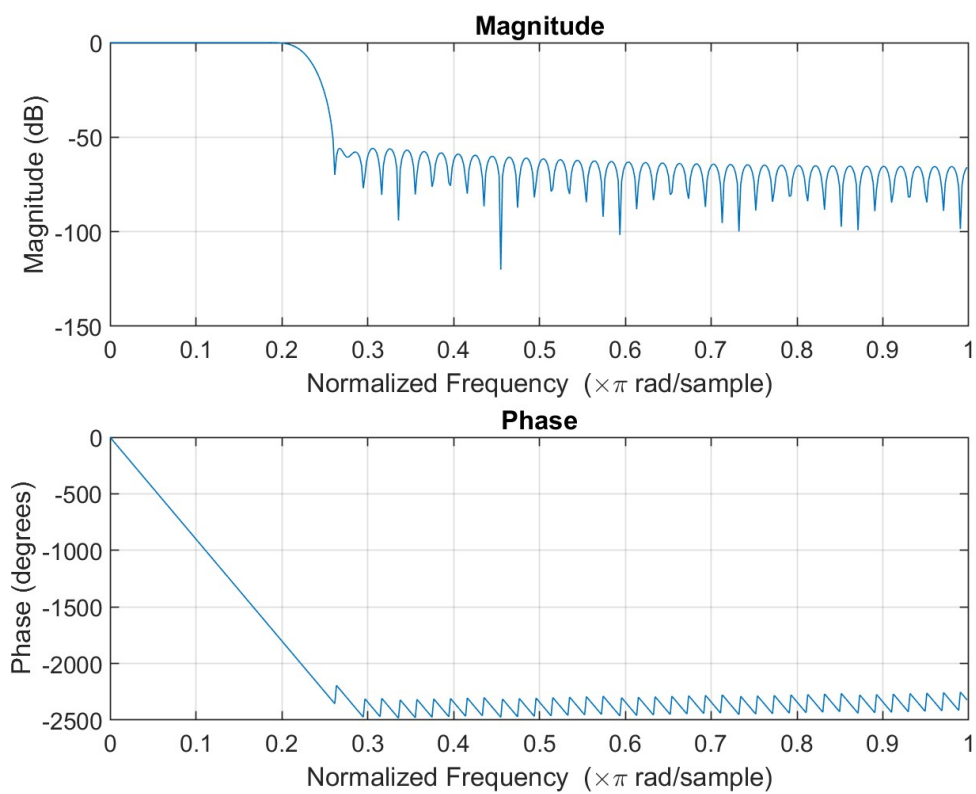
Wszystkie wykresy poniżej były analizowane na sygnale nagrany mikrofonem laptopa. Stąd zakres częstotliwości wynosi 0-24000 Hz.

1.

a)



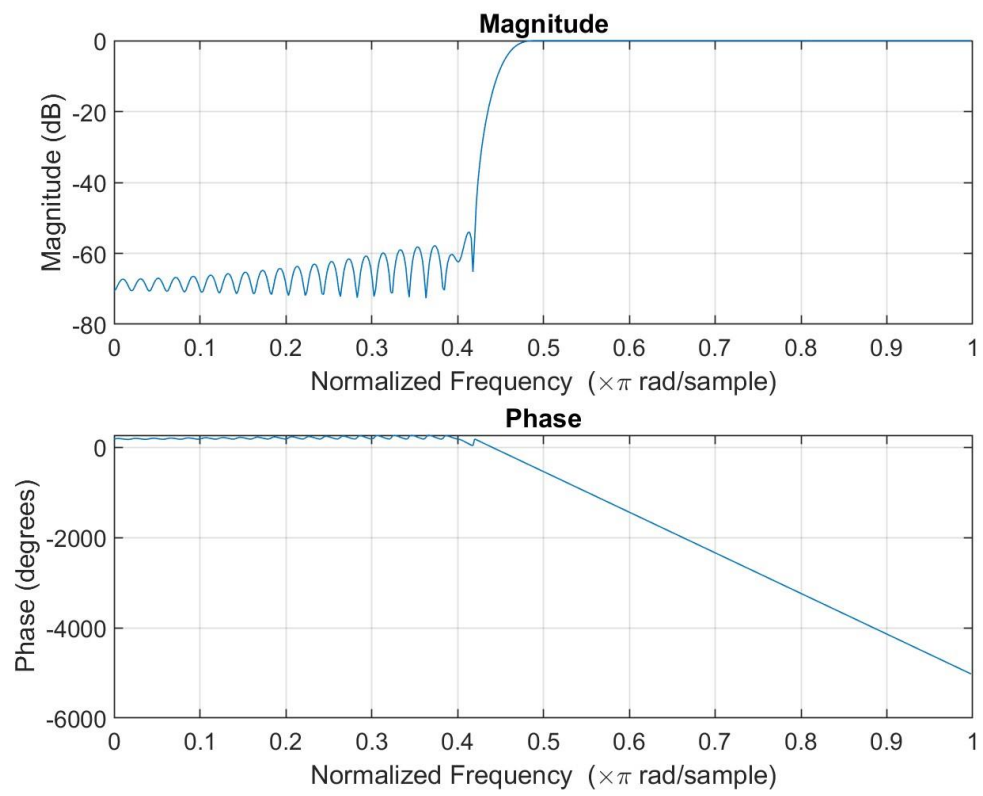
b)



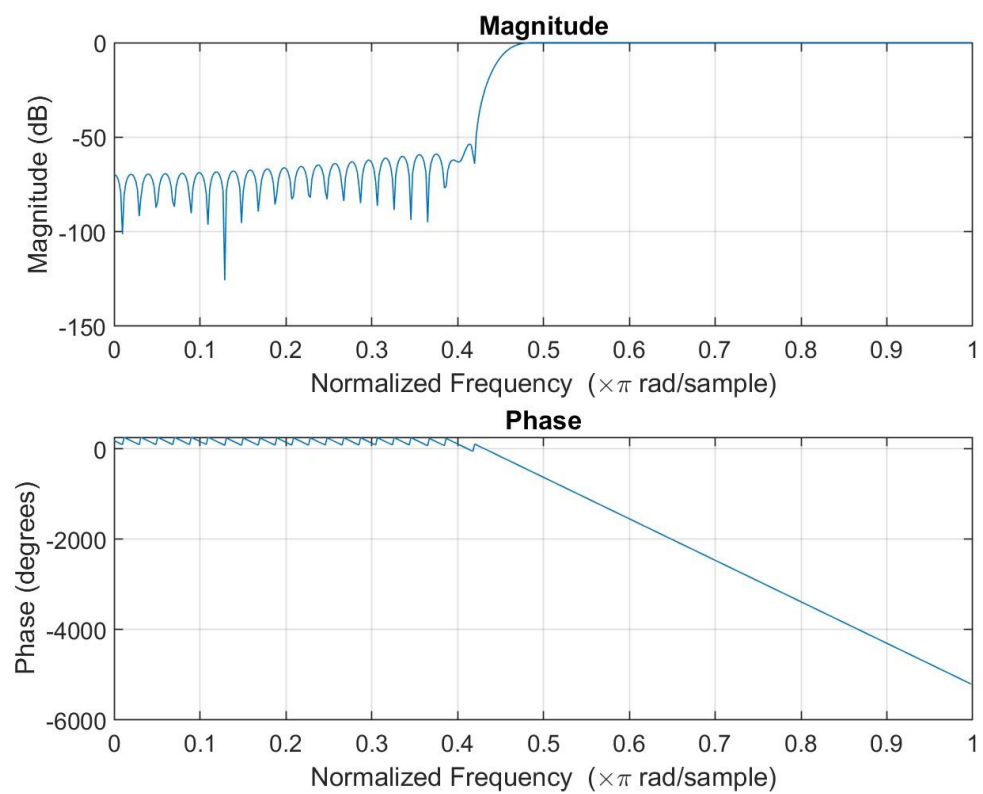
Porównanie charakterystyki modułu transmitancji oraz fazowej dla filtra dolnoprzepustowego z implementacją własnego algorytmu a) oraz algorytmu Matlab b). $N=101$, $f_g = 5000$ Hz

2.

a)



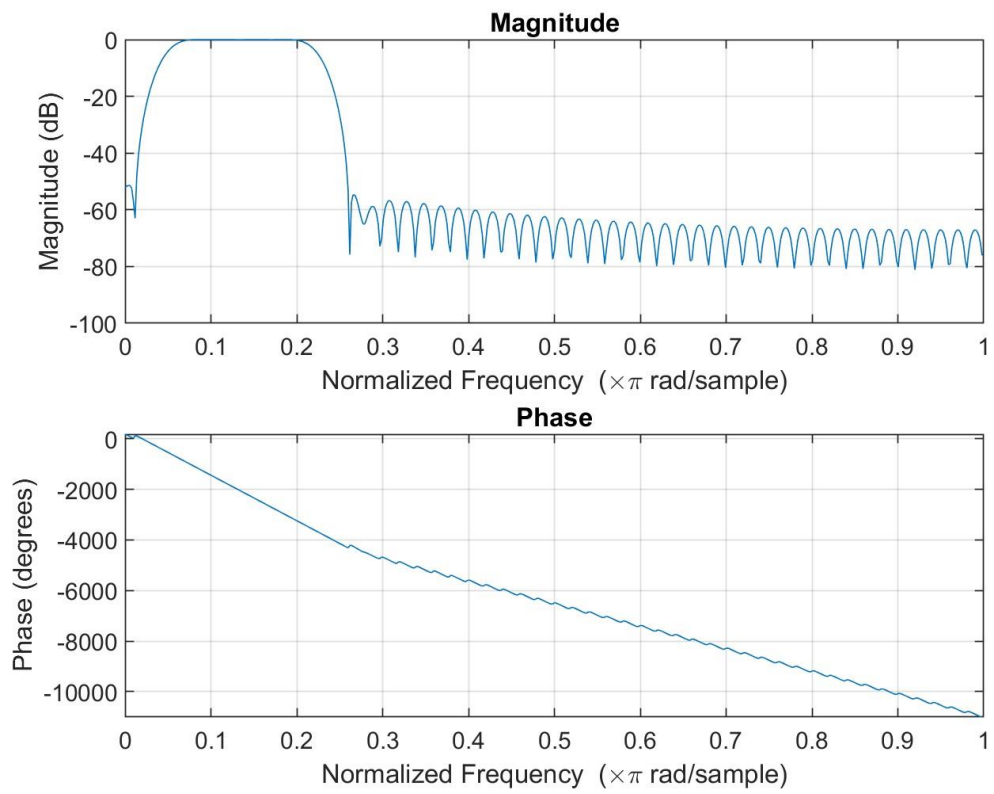
b)



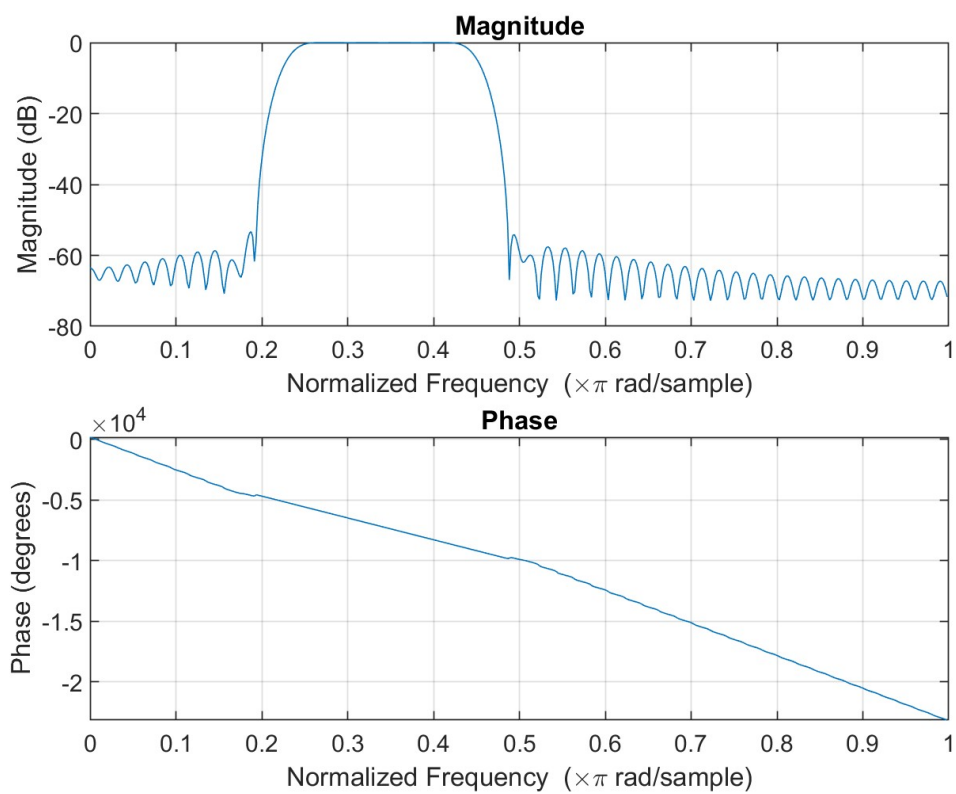
Porównanie charakterystyki modułu transmitancji oraz fazowej dla filtra dolnoprzepustowego z implementacją własnego algorytmu a) oraz algorytmu Matlab b). $N=101$, $f_g = 10000$ Hz

3.

a)



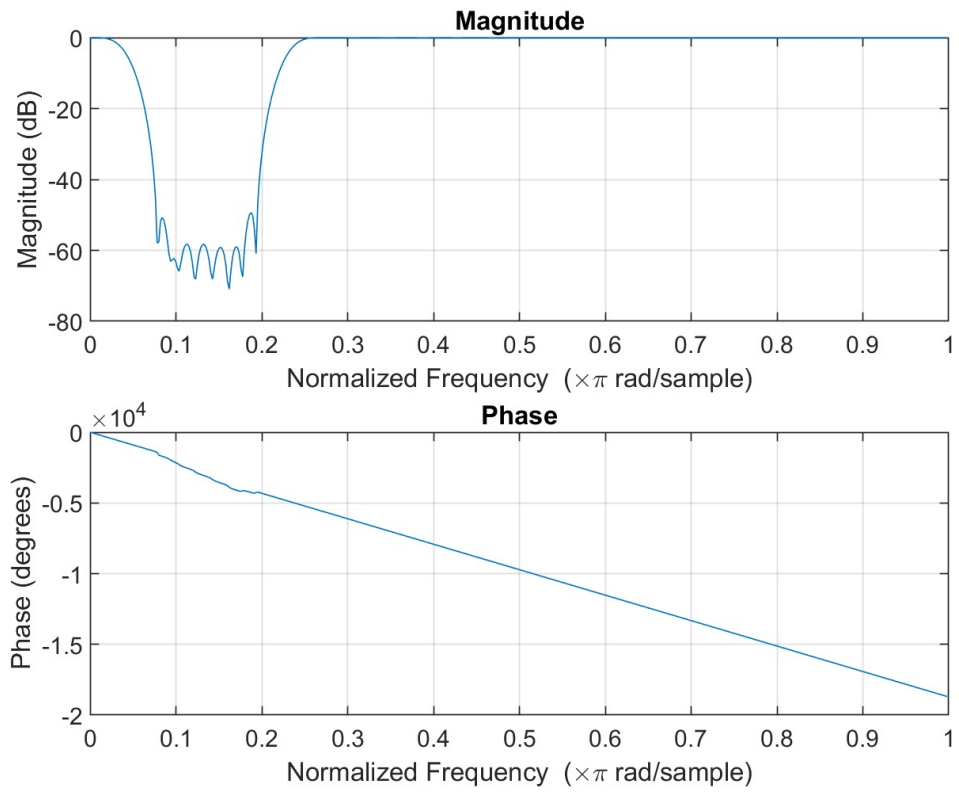
b)



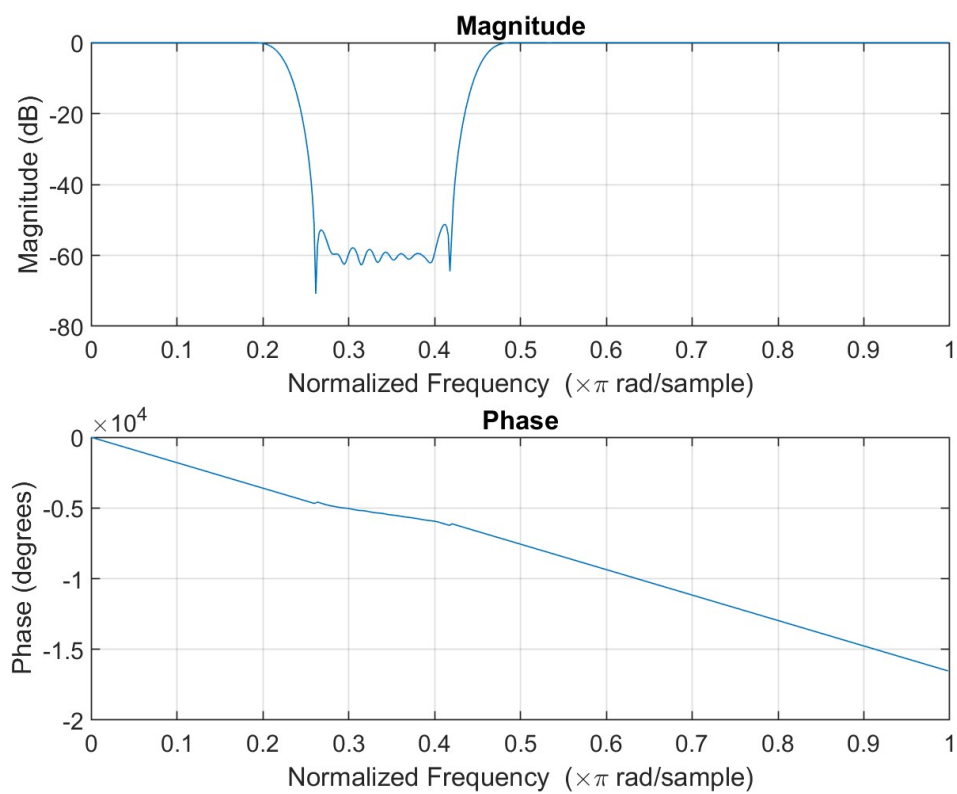
Porównanie charakterystyki modułu transmitancji oraz fazowej dla filtru pasmo-przepustowego a) $N=101$, $f_g = 5000$ Hz, $f_d = 1000$ Hz oraz b). $N=101$, $f_g = 10000$ Hz, $f_d = 5000$ Hz

4.

a)



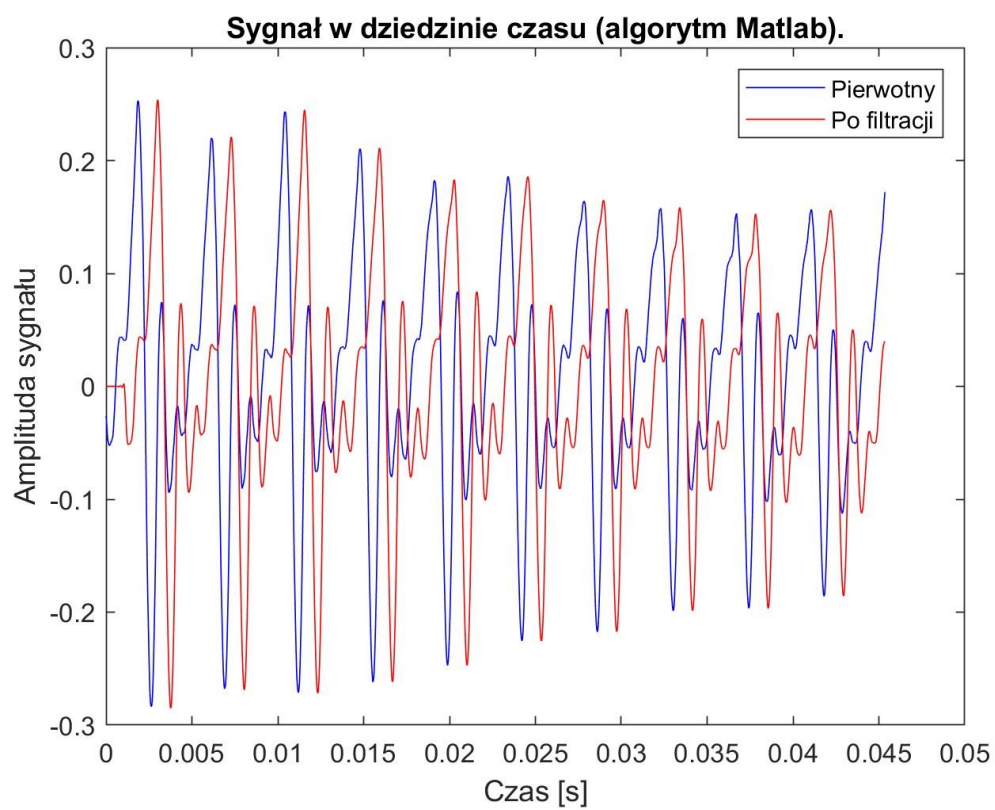
b)

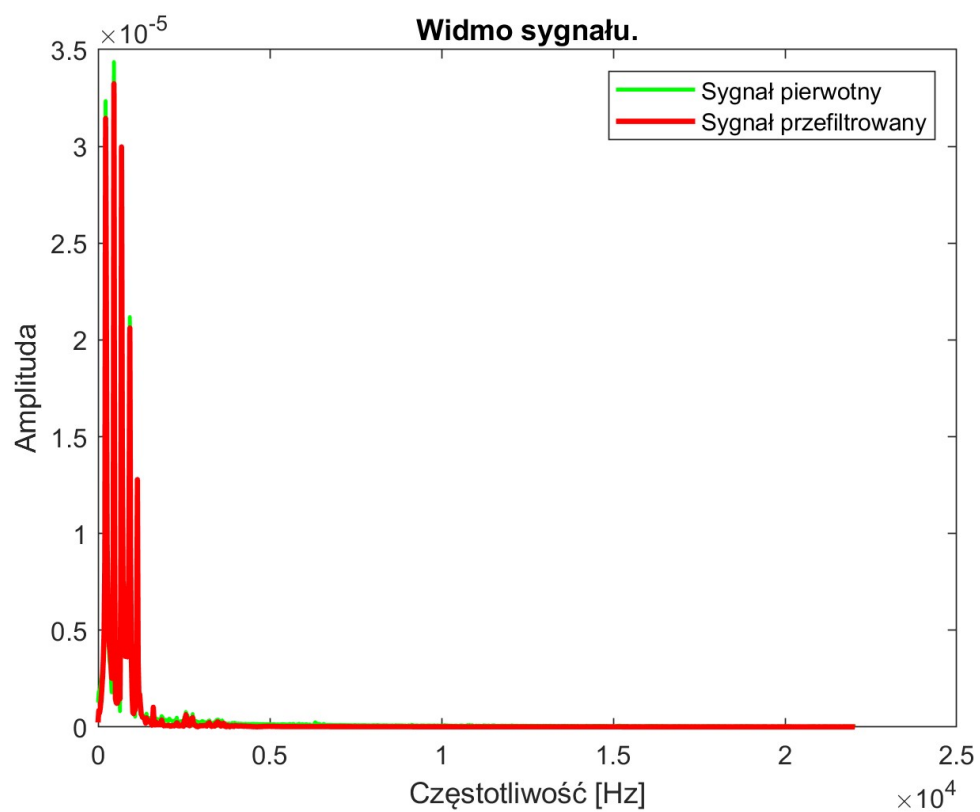
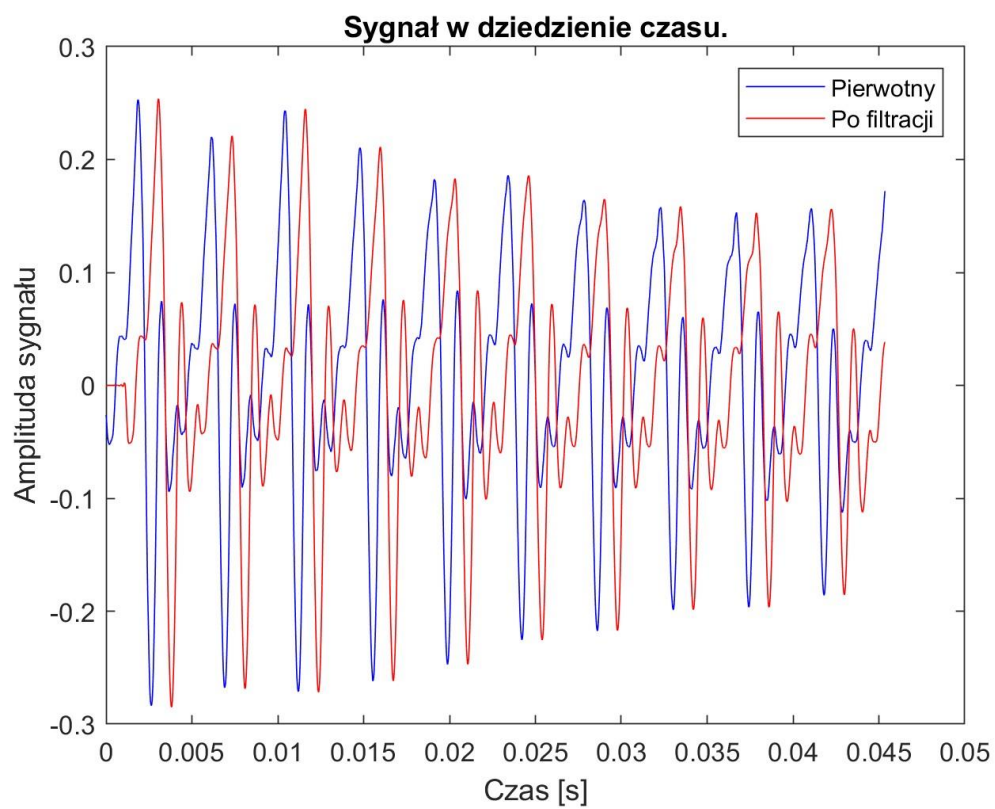


Porównanie charakterystyki modułu transmitancji oraz fazowej dla filtra pasmo-zaporowego a) $N=101$, $f_g = 5000$ Hz, $f_d = 1000$ Hz oraz b). $N=101$, $f_g = 10000$ Hz, $f_d = 5000$ Hz

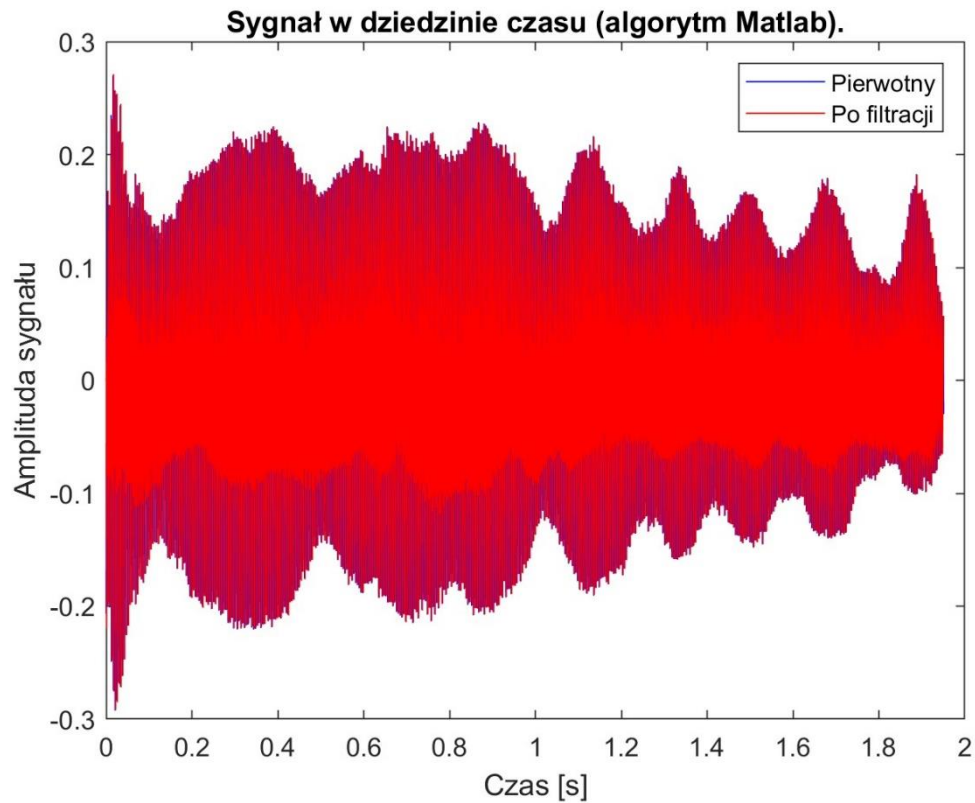
Podsumowując, jak widać udało nam się osiągnąć nawet lepsze tłumienie sygnału w paśmie zaporowym porównując filtry dolnoprzepustowy oraz górnoprzepustowy.

Sygnał przed i po filtracji, porównanie widma

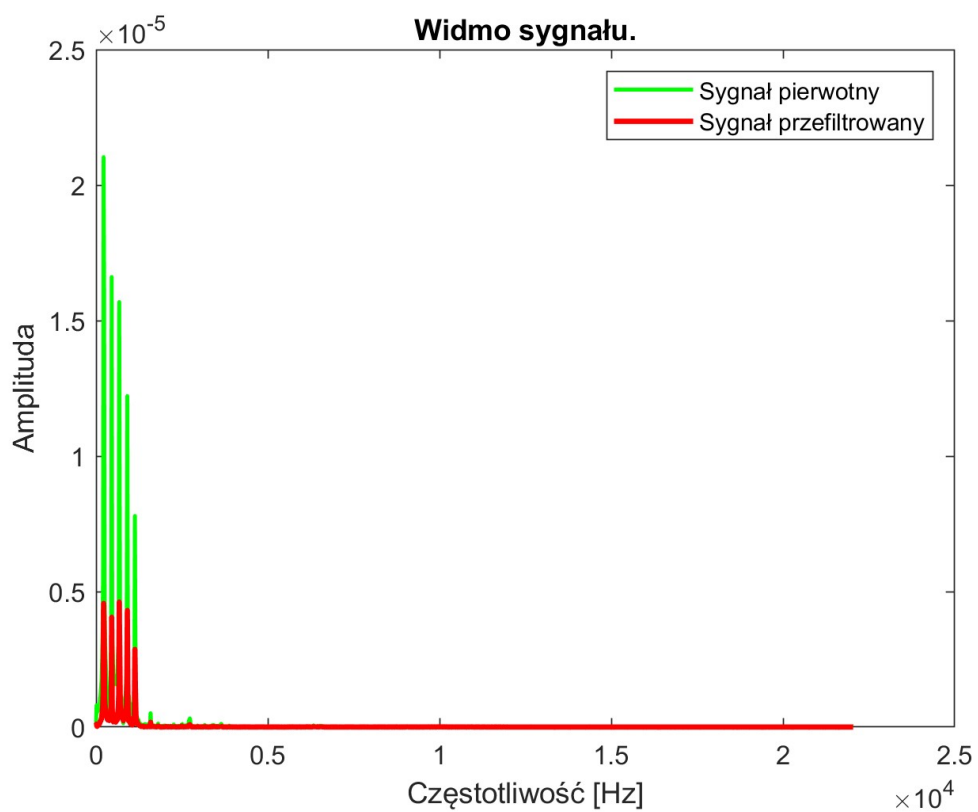
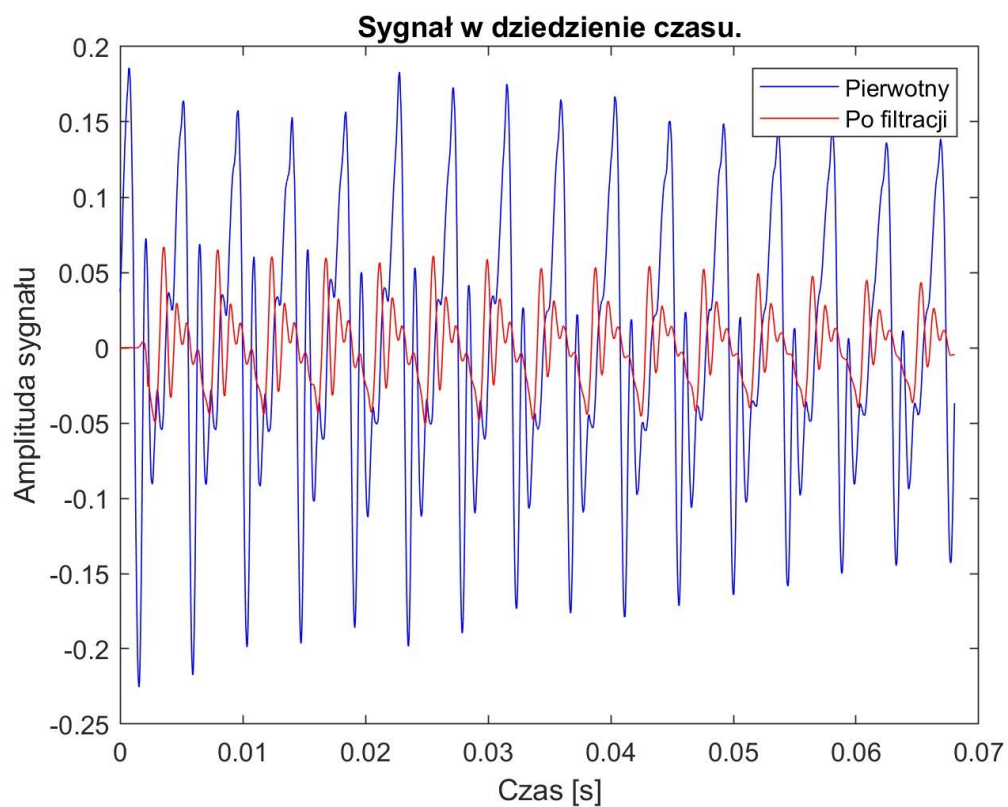




To porównanie sygnału dotyczy filtra dolnoprzepustowego oraz fragmentu sygnału nagranych za pomocą mikrofonu. $N=101$, $f_g = 5000$ Hz. Ten sygnał składa się jak widać w głównej mierze z bardzo niskich częstotliwości, dlatego też nie widać różnicy. Poniżej znajduje się przykład dla tych samych parametrów filtra, ale dla całego sygnału:

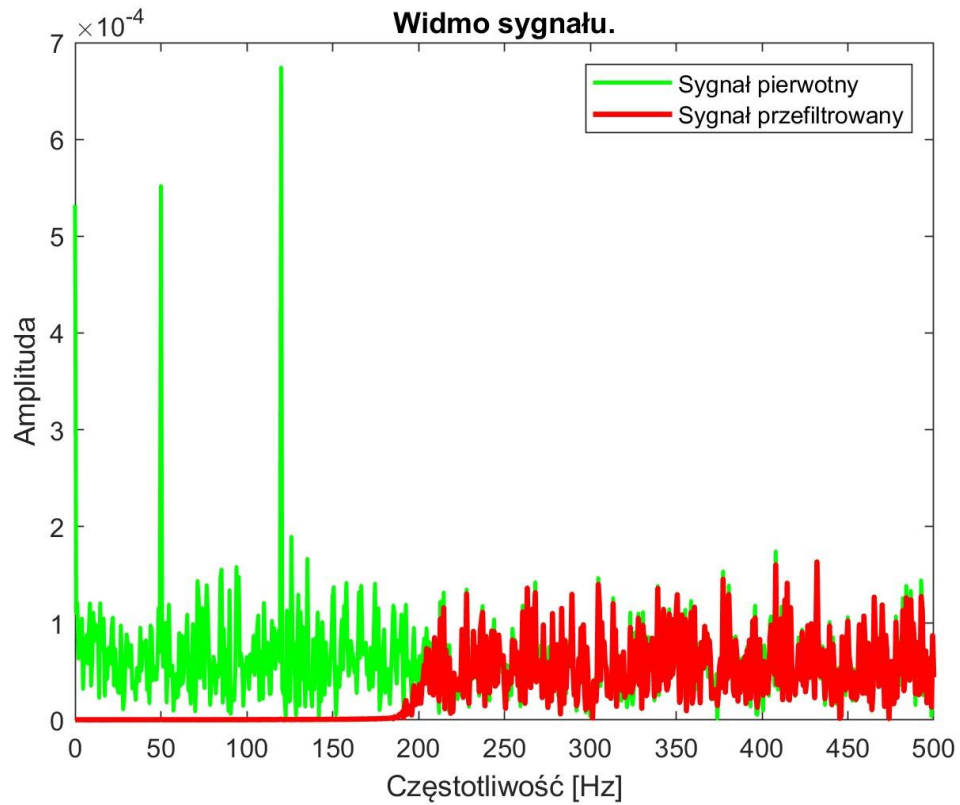


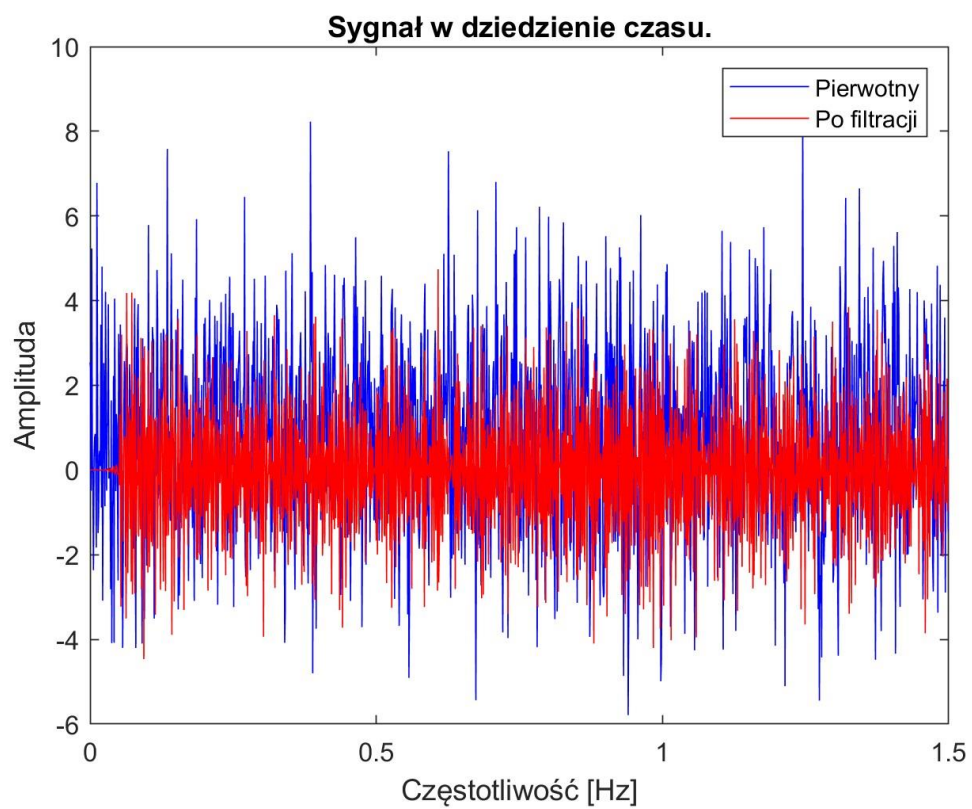
Przykład poniżej dotyczy filtra pasmo-przepustowego $N = 101$, $f_d = 1000$ Hz $f_g = 10000$ Hz. Można zauważyć, że częstotliwości początkowe zostały wszystkie prawie odfiltrowane. Stąd amplituda sygnału na wyjściu pomniejszyła się.



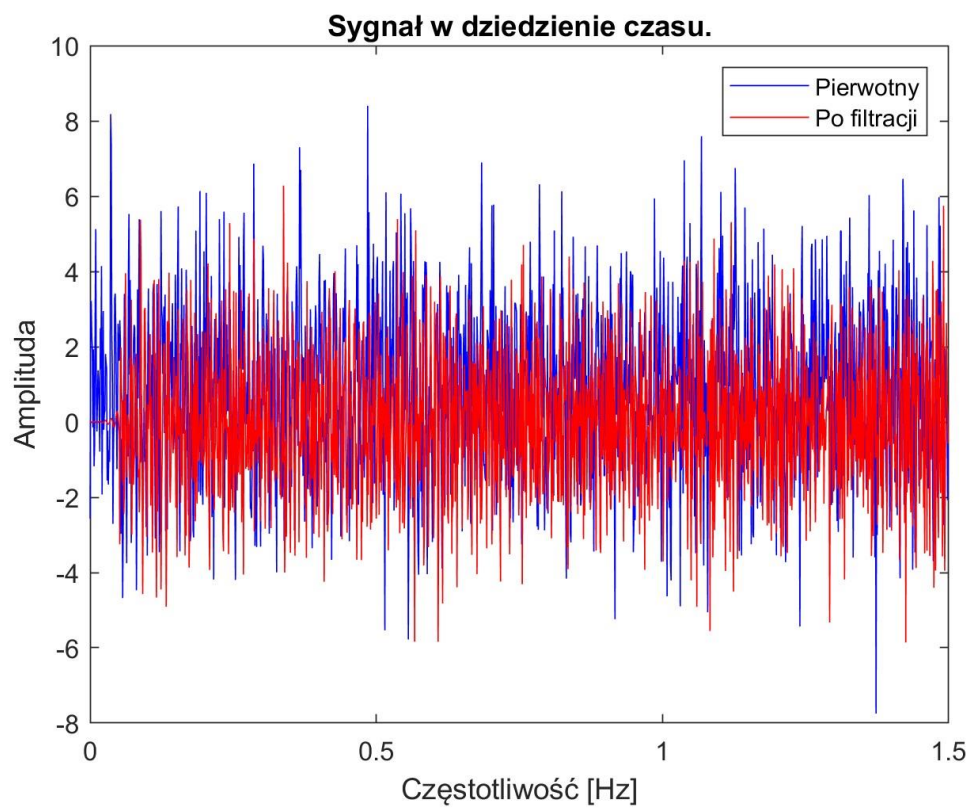
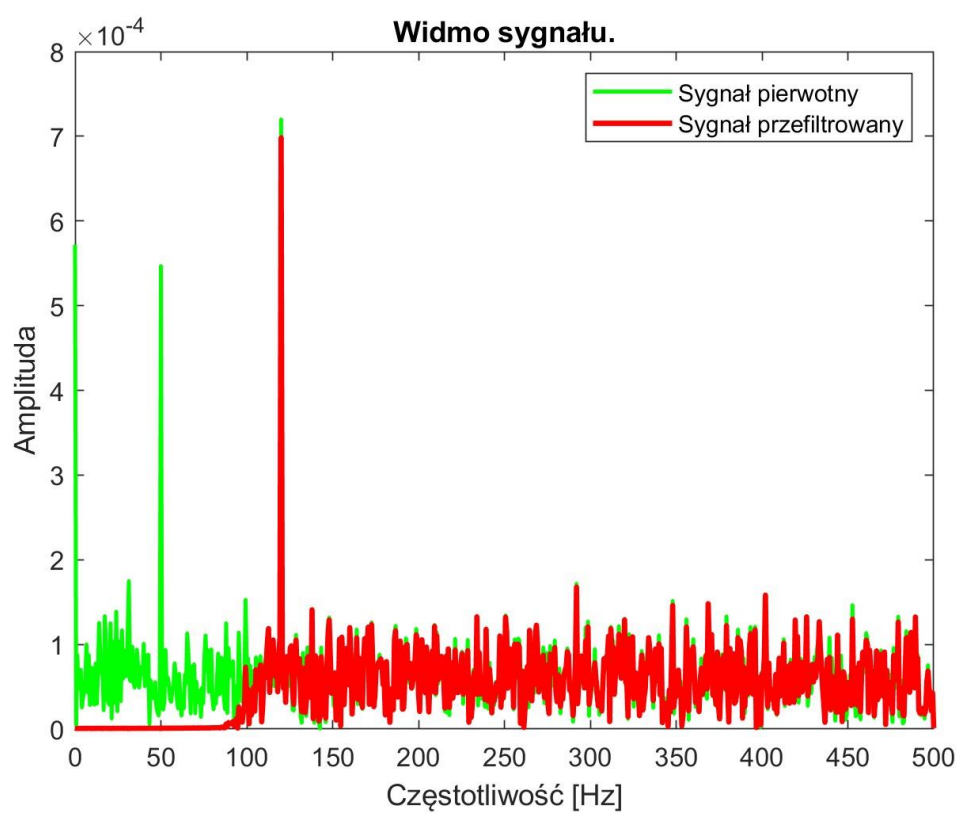
To były przykłady zrealizowane na sygnale pochodzącym z mikrofonu. Natomiast poniżej wygenerowaliśmy sygnał składający się z składowej stałej, dwóch różnych funkcji sinus + szumu. Na poniższych przykładach, można bardzo dobrze zauważyć działanie filtru. Filtr górnoprzepustowy $N = 101$.

a) $f_d = 200$ Hz





b) $f_d = 100 \text{ Hz}$



Powyższe przykłady idealnie ilustrują działanie filtru, usuwanie niechcianych częstotliwości z pasma sygnału. Można również zauważyć, że na początku odfiltrowanego sygnału pojawia się zero, dzieje się tak dlatego, że przycinamy sygnał do granic i zastępujemy resztę zerem. Poza tym do obliczenia n-tej próbki potrzebujemy N poprzednich próbek sygnału początkowego. Na koniec zakończenia algorytmu zapisujemy wynik do pliku 'wynik.wav'.

Podsumowanie

Przeanalizowane wyniki implementacji dowodzą skuteczności i użyteczności filtrów cyfrowych. Wynik wykonania zależy w głównej mierze od projektanta. Najpierw należy wybrać charakterystykę filtru, a następnie zdecydować się jakie częstotliwości chcemy usunąć bądź przepuścić.

Funkcje gotowe (Matlab) użyte w projekcie:

- FIR, filter (do porównania)
- FFT (widmo)
- conv (splot)
- freqz (ch-styka fazowa, częstotliwościowa)
- hamming (generacja okna hamminga)

Multimedia:

<https://multimed.org/student/zps/o9-Filtry.pdf>

<https://100ologos.net/matlab-logo/>