

Projekt: Techniki Obliczeniowe

Temat: Implementacja różnych filtrów cyfrowych wraz z prezentacją na przykładowym sygnale

Autorzy: Hubert Mąka, Dominik Marszałek

Wstęp:

W projekcie do zaimplementowania mamy różnego rodzaju filtry cyfrowe za pomocą języka Matlab.

Filtry cyfrowe to układy elektroniczne lub algorytmy, które służą do przetwarzania sygnałów cyfrowych. Mogą one być używane do różnych celów, takich jak odfiltrowywanie szumów lub zakłóceń z sygnału, przekształcanie sygnału z jednej częstotliwości na inną, czy też wygładzanie sygnału. W przeciwieństwie do filtrów analogowych, filtry cyfrowe wykorzystują cyfrowe metody obliczeniowe do przetwarzania sygnału, co pozwala na bardziej precyzyjne i dokładne działanie. Dzielą się one na różne rodzaje, w zależności od sposobu ich implementacji i charakterystyki częstotliwościowej.

My postanowiliśmy w projekcie zawrzeć cztery wybrane przez nas filtry cyfrowe, których opis jest przedstawiony poniżej:

-Filtr FIR

Filtry FIR (ang. Finite Impulse Response) to rodzaj filtrów cyfrowych, które charakteryzują się tym, że mają skończoną odpowiedź impulsową. Oznacza to, że sygnał, który jest przetwarzany przez taki filtr, po przejściu przez niego ma skończony czas trwania. Filtry FIR są bardzo często używane w różnych aplikacjach przetwarzania sygnałów, ponieważ są one bardziej precyzyjne i dokładne niż inne rodzaje filtrów cyfrowych. Ponadto, filtry FIR nie wymagają stosowania dodatkowych układów sprzężenia zwrotnego, co czyni je bardziej prostymi w budowie i użyciu.

-Filtr IIR

Filtry IIR (ang. Infinite Impulse Response) to rodzaj filtrów cyfrowych, które charakteryzują się tym, że mają nieskończoną odpowiedź impulsową. Oznacza to, że sygnał, który jest przetwarzany przez taki filtr, po przejściu przez niego nie ma skończonego czasu trwania. Filtry IIR są często używane w różnych aplikacjach przetwarzania sygnałów, ponieważ ich budowa jest prostsza niż filtrów FIR i wymagają mniejszej ilości obliczeń. Jednak ze względu na swoją nieskończoną odpowiedź impulsową, filtry IIR mogą być mniej precyzyjne i dokładne niż filtry FIR.

-Filtr dolnoprzepustowy

Filtry dolnoprzepustowe są filtrami, które przetwarzają sygnał przez odfiltrowywanie wszystkich składowych o częstotliwościach wyższych niż określona wartość progowa. Służą one do zmniejszenia zawartości szumów i zakłóceń o wysokich częstotliwościach w sygnale, co pozwala na lepszą jakość dźwięku lub obrazu.

-Filtr medianowy

Filtr medianowy to rodzaj filtra cyfrowego, który działa na zasadzie wyznaczania mediany dla pewnego okna sąsiednich punktów sygnału wejściowego i przypisywania jej jako wartości punktu sygnału wyjściowego. Dzięki temu filtracja medianowa pozwala na usunięcie wartości odstających (outlierów) i innych niepożądanych składowych sygnału. Jest to szczególnie przydatne w przypadku sygnałów zawierających duże ilości szumów.

Solver'y:

Matlab posiada wiele solver'ów do cyfrowego przetwarzania/filtrowania sygnałów. Dzięki tym narzędziom można szybko i łatwo zaprojektować i zaimplementować różne rodzaje filtrów cyfrowych, takich jak filtry FIR, IIR, medianowe czy dolnoprzepustowe, i wykorzystać je do przetwarzania sygnałów w różnych aplikacjach. Oto przykładowe z nich:

- **fir1()** - służy do projektowania filtrów FIR o dowolnej charakterystyce częstotliwościowej
- **filter()** - służy do przetwarzania sygnałów przez filtr cyfrowy
- **freqz()** - służy do obliczania charakterystyki częstotliwościowej filtru cyfrowego
- **grpdelay()** - służy do obliczania opóźnienia grupowego filtru cyfrowego
- **impz()** - służy do obliczania odpowiedzi impulsowej filtru cyfrowego
- **medfilt1()** - służy do przetwarzania sygnałów przez filtr medianowy 1-wymiarowy

Posłużą one nam głównie do testowania poprawności napisanych przez nas algorytmów.

Podsumowanie:

Algorytmy przez nas napisane będą miały zadanie rozwiązanie potrzebnych do uzyskania poprawnego wyniku równań, oraz odpowiedniego przetworzenia sygnału wejściowego. Wyniki będą prezentowane w formie wykresów zestawiających działanie algorytmów (sygnał na wejściu i wyjściu) oraz opisu stworzonego przez nas kodu.

Raport końcowy:

Część 1:

Niniejszy raport przedstawia wyniki rozważań dotyczących filtracji medianowej oraz ruchomej średniej. Filtracje te są często stosowane w celu usunięcia szumów z danych oraz poprawy jakości obrazów. Poniżej zostaną porównane wyniki działania obu algorytmów na zamodelowanych zaszumionych sygnałach:

1. Szum Gauss'owski
2. Szum impulsowy

Krótki teoretyczny opis algorytmów:

Algorytm ruchoma średnia:

Wzór na obliczanie średniej:

$$y(i) = \frac{1}{W} \sum_{j=i-W+1}^i u(j)$$

Gdzie W jest szerokością okna (liczbą uśrednianych próbek).

Algorytm medianowy:

Algorytm filtracji medianowej przebiega następująco:

- a) Oknem prostokątnym o długości W wybiera się z sygnału W próbek.
- b) Porządkuje się wybrane próbki rosnąco.
- c) Oblicza się medianę uporządkowanych próbek (wartość środkową ciągu).
- d) Przesuwa się okno o jedną próbkę do przodu i powtarza punkty b) i c).

Wartości obliczane w punkcie c) są sygnałem przefiltrowanym.

Ponieważ na początku okno „nachodzi” na filtrowany sygnał, a na końcu „schodzi” to w wyniku filtracji otrzymujemy sygnał o długości $W - N + 1$, gdzie N to długość sygnału do filtracji. Z przefiltrowanego sygnału należy odrzucić $\left\lfloor \frac{W}{2} - 1 \right\rfloor$ pierwszych i ostatnich próbek.

Wzór na obliczanie mediany:

$$\text{med}(x_1, x_2, \dots, x_N) = \begin{cases} x_{k+1}, & N = 2k + 1 \\ \frac{1}{2}(x_k + x_{k+1}), & N = 2k \end{cases}$$

Parametry testowego sygnału:

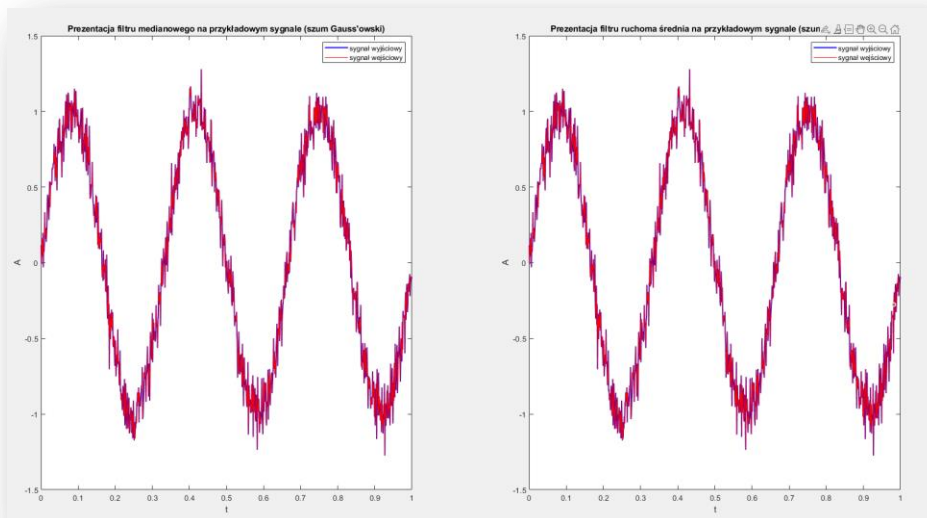
```
% generowanie sygnału sinusoidalnego
fs = 1000; % częstotliwość próbkowania (Hz)
f = 3; % częstotliwość sygnału (Hz)
t = 0:1/fs:1; % czas (s)
window_size = 4; % rozmiar okna
signal = sin(2*pi*f*t); % sygnał sinusoidalny

% szum Gauss'owski
mean = 0; % średnia zaszumienia
std = 0.1; % odchylenie standardowe szumu
noise = mean + std * randn(size(signal)); % generacja szumu
noisy_signal1 = signal + noise; % złożenie zaszumionego sygnału1

% szum impulsowy
p = 0.1; % prawdopodobieństwo szumu
noise = (rand(size(signal)) < p); % generacja szumu
noisy_signal2 = signal + noise; % złożenie zaszumionego sygnału2
```

Poniżej przedstawiamy wykresy z wynikami filtracji dla sygnału o stałych parametrach, zmieniając tylko okno próbkowania.

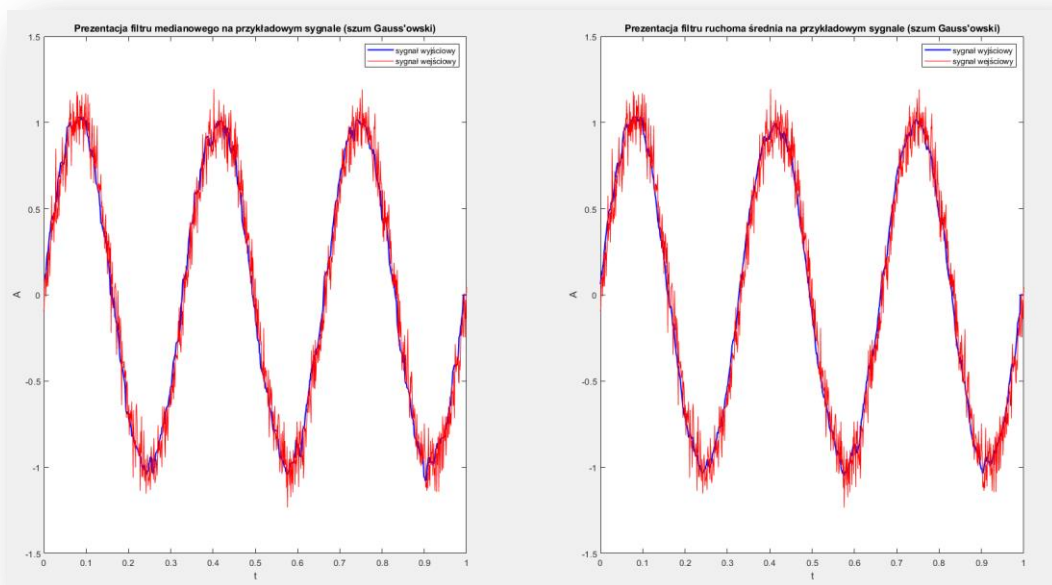
1. Sygnał z szumem Gauss'owskim



Rozmiar okna: 1

Komentarz:

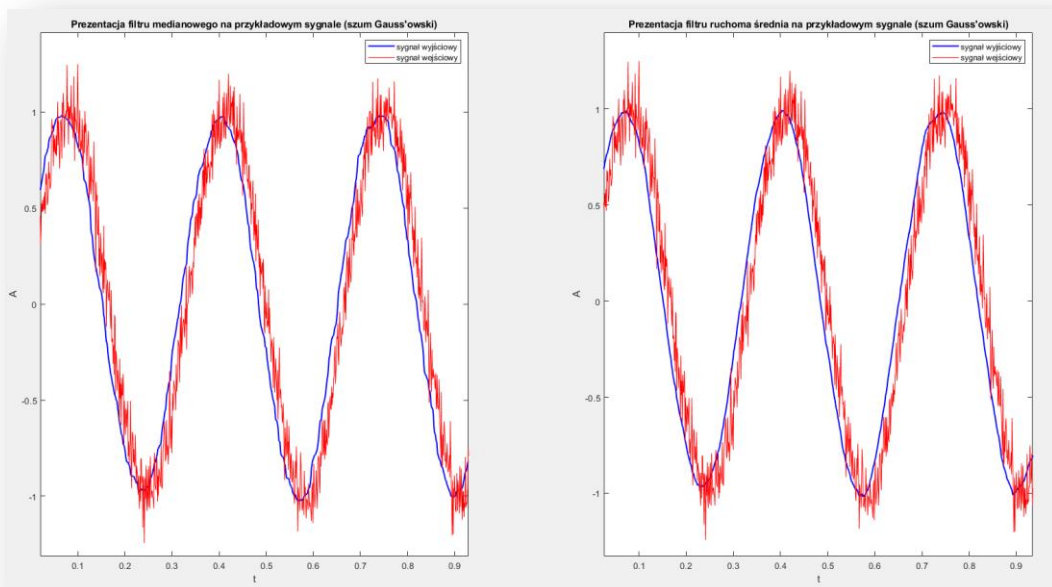
Rozmiar okna równy 1 jest równoważny z brakiem filtracji. Sygnał wejściowy równy sygnałowi wyjściowemu.



Rozmiar okna: 10

Komentarz:

Widać już zarysy filtracji, jednak nie można jeszcze stwierdzić, który filtr lepiej działa.

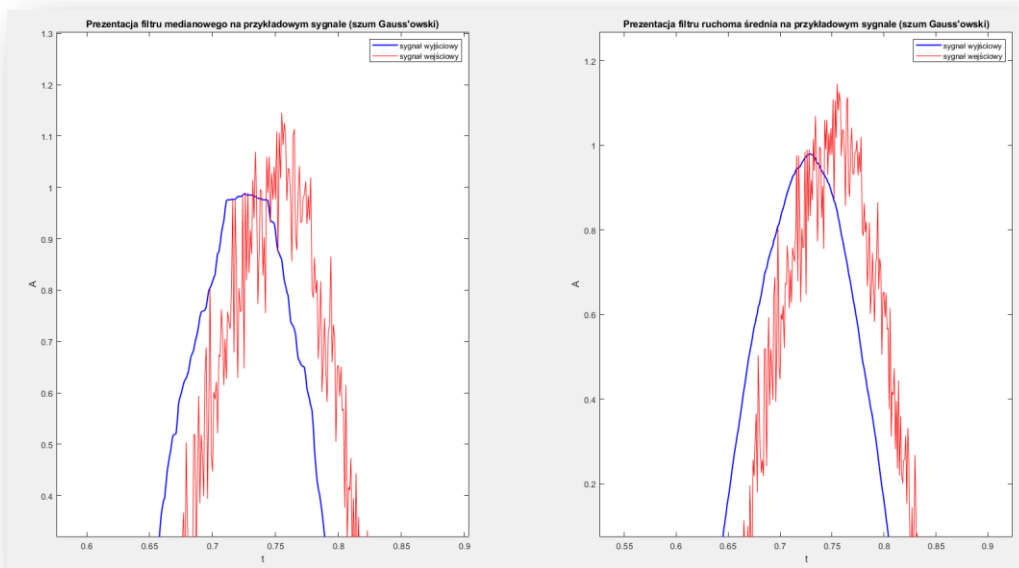


Rozmiar okna: 30

Komentarz:

Można zauważyć, że filtr typu ruchoma średnia dla większych okien działa lepiej na zaszumiony sygnał Gauss'owski.

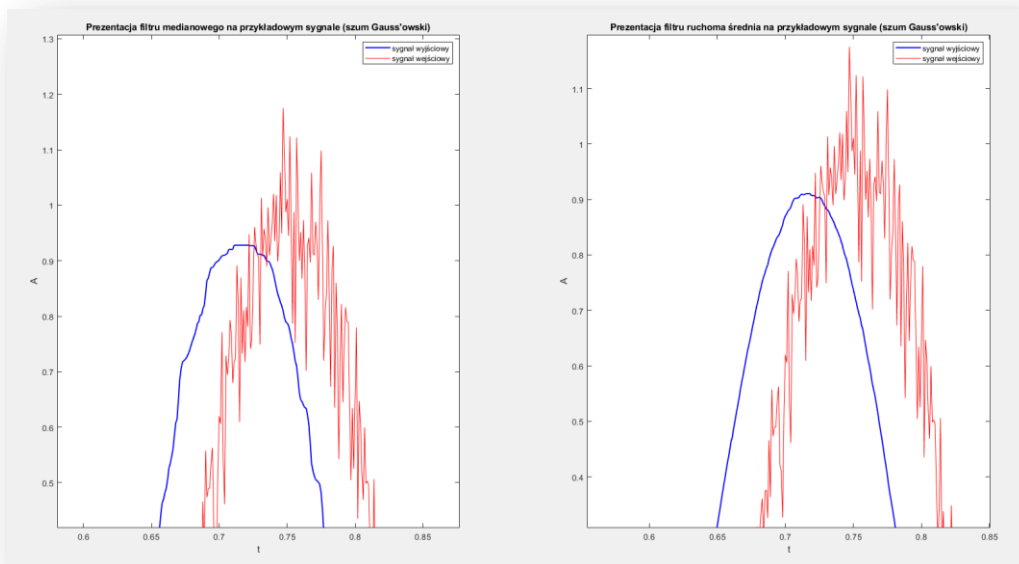
Poniżej w celu zdobycia lepszego obrazu sytuacji przyjrzelśmy się tylko jakości wierzchołków sygnału w celu lepszego uwidocznienia zachodzących zmian.



Rozmiar okna: 50

Komentarz:

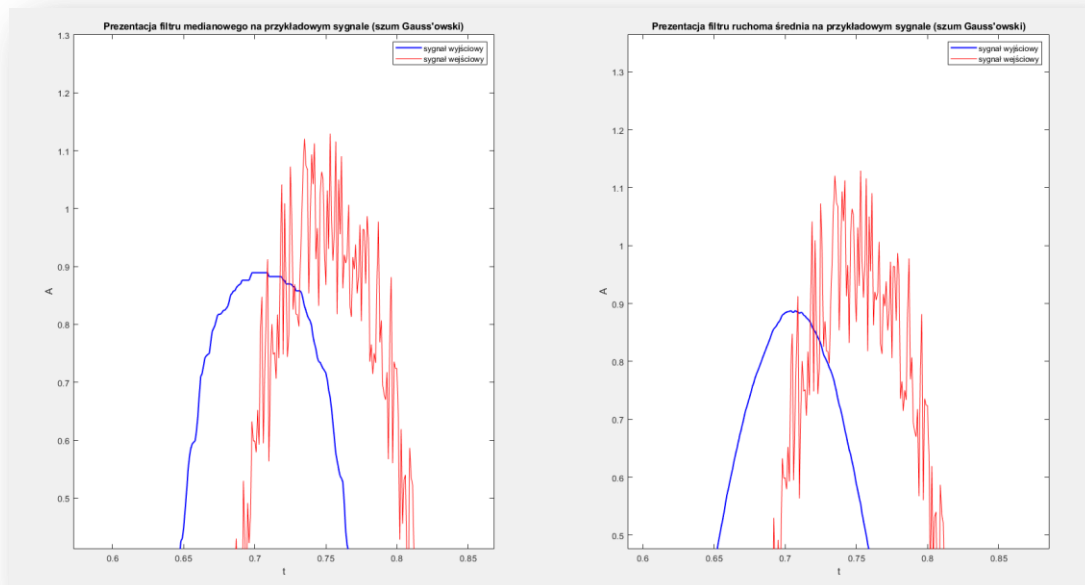
Można zauważyć znaczną przewagę filtru typu ruchoma średnia. Sygnał jest bardziej gładki/ ma mniej nagłych skoków.



Rozmiar okna: 70

Komentarz:

Dalsze testy nadal wykazują przewagę filtru typu ruchoma średnia.



Rozmiar okna: 90

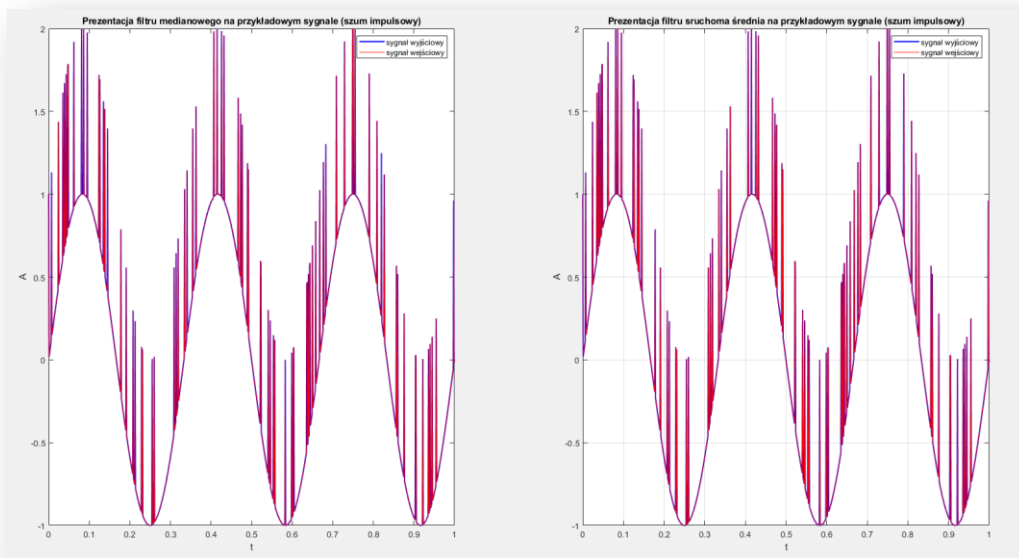
Komentarz:

Dalsza przewaga filtru ruchoma średnia.

Wnioski:

Jak można zauważyć filtr typu ruchoma średnia świetnie sprawdza się w filtrowaniu zakłóceń szumu losowego. Ze zwielokrotnianiem rozmiaru okna idzie znaczna poprawa sygnału wyjściowego czego nie można powiedzieć o filtrze medianowym. Ponadto można zauważyć zmniejszanie się amplitudy sygnału spowodowane coraz większym uśrednianiem wartości wokół wierzchołka.

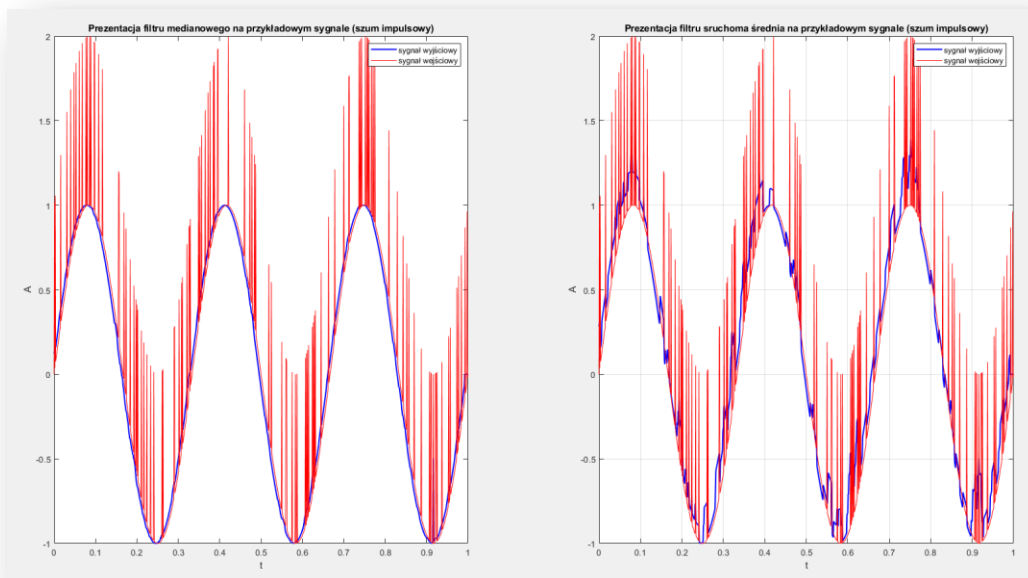
2. Sygnał z szumem impulsowym



Rozmiar okna: 1

Komentarz:

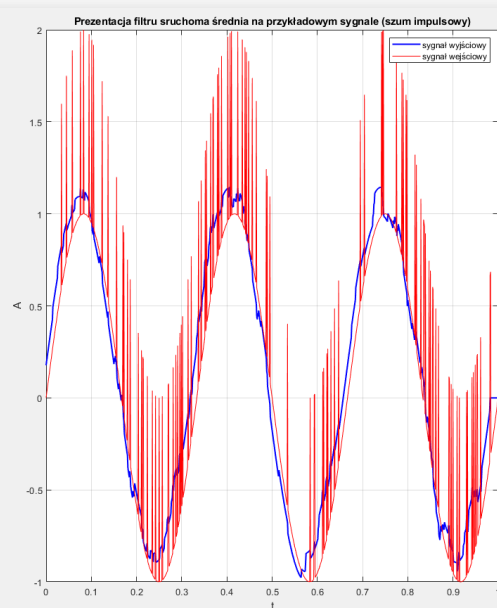
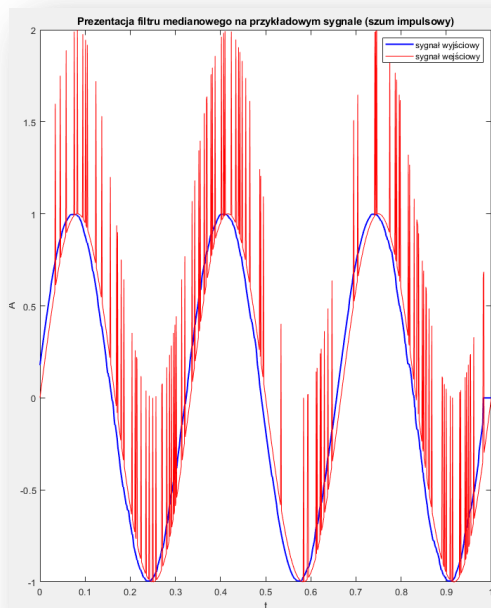
Rozmiar okna równy 1 jest równoważny z brakiem filtracji. Sygnał wejściowy równy sygnałowi wyjściowemu.



Rozmiar okna: 10

Komentarz:

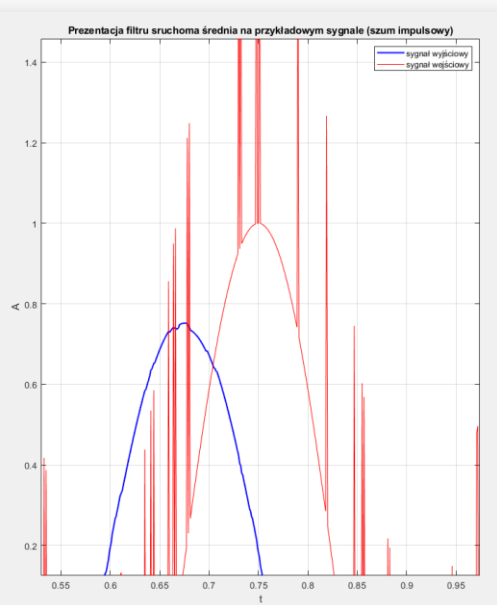
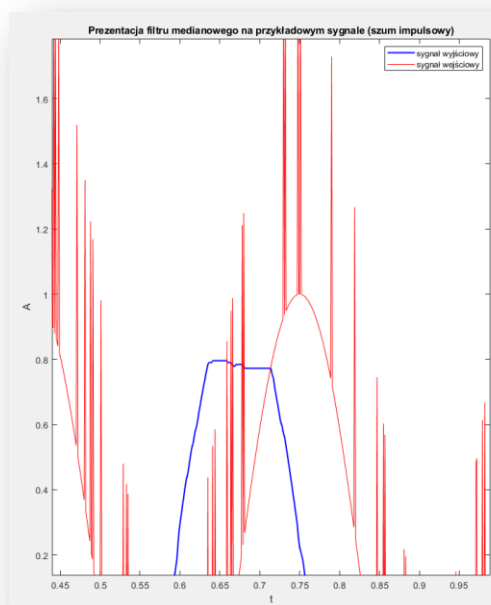
Już przy rozmiarze okna 10 można zauważyć znaczną przewagę filtru medianowego, który dzięki liczeniu mediany z okna nie wyłapuje nagłych skoków jednostkowych w przeciwieństwie do liczenia średniej, która jak można uśrednia skoki jednostkowe.



Rozmiar okna: 30

Komentarz:

Zwiększenie okna tylko potwierdza wcześniejsze wnioski.



Rozmiar okna: 150

Komentarz:

Jednak można zauważyć, że dla ekstremalnie dużych okien filtr medianowy mocno obcina sygnał.

Wnioski:

Filtr medianowy dla sygnału z szumem impulsowym działa znacznie lepiej niż filtr ruchoma średnia już przy bardzo małych rozmiarach okna. Jednak przy ekstremalnie dużych rozmiarach znacznie obcina on końcówki sygnału. Należy go więc stosować w zakresie rozsądnych rozmiarów okien.

Końcowe wnioski i podsumowanie:

1. Filtry ruchomej średniej i medianowe są przeznaczone do usuwania zaszumienia impulsowego lub szumu losowego.
2. Filtry ruchomej średniej są skuteczniejsze w usuwaniu szumu losowego, który jest rozłożony równomiernie na całym zakresie danych.
3. Filtry medianowe natomiast są skuteczniejsze w usuwaniu szumu impulsowego, który składa się z pojedynczych, nietypowych wartości, które odbiegają znacznie od otoczenia.

Bibliografia:

„Cyfrowe przetwarzanie sygnałów od teorii do zastosowań” - Tomasz P. Zieliński

„MATLAB w zastosowaniu do obliczeń obwodowych i przetwarzania sygnałów” – Stanisław Osowski, Andrzej Cichocki, Krzysztof Siwek

<https://home.agh.edu.pl/~mkl/lab5.pdf>