

Lab. 7 Filtracja sygnałów, filtry FIR			
Nazwisko, Imię	Data wykonania ćwiczenia	Planowy dzień zajęć	Planowa godzina zajęć
Dziuba Wojciech	10.04.2019	Środa	08:00

1) Informacje na temat w jaki sposób projektujemy filtry IIR

Dolnoprzepustowe:

Procedura projektowania rozpoczyna się od określenia wymaganej funkcji transmitancji, po czym następuje obliczenie współczynników filtru, które dadzą taką funkcję transmitancji. Przy projektowaniu najczęściej stosowana jest jedna z dwóch metod: metoda okna i tzw. metoda optymalna.

Środkowoprzepustowe:

Metoda projektowania dolnoprzepustowych filtrów może być użyta jako pierwszy krok przy projektowaniu środkowoprzepustowego filtru. Jeżeli zdefiniujemy współczynniki dolnoprzepustowego filtru jako $h_{lp}(n)$ to naszym problemem będzie znalezienie współczynników $h_{bp}(n)$ środkowoprzepustowego filtru.

Górnoprzepustowe:

Możemy wykorzystać technikę projektowania środkowoprzepustowego filtru do projektowania filtrów górnoprzepustowych. Aby wyznaczyć współczynniki filtru górnoprzepustowego, musimy jedynie zmodyfikować ciąg przesuwający $s_{shift}(n)$, tak aby reprezentował on spróbkowaną sinusoidę o częstotliwości $f_s/2$.

2) Różnice między filtrami FIR i IIR

Właściwość	IIR	FIR
Liczba wymaganych mnożeń	Mała	Duża
Stabilność	Musi być projektowana	Zagwarantowana
Liniiowość fazy	Nie	Zagwarantowana
Sprzętowe wymagania dla pamięci	Małe	Duże
Złożoność sprzętowa układu sterowania filtru	Umiarkowana	Mała
Łatwość projektowania lub stopień złożoność oprogramowania wspomagającego projektowanie	Umiarkowana	Prosta
Możliwość filtracji adaptacyjnej	Tak	Tak

3) Kiedy bardziej wskazane jest użycie filtru FIR a kiedy IIR

Wychodząc ze sprzętowego punktu widzenia, gdzie różnice pomiędzy filtrami FIR i IIR są zasadnicze, nasz wybór musi wynikać z tych właściwości filtru, które są najbardziej i najmniej

dla nas istotne.

Na przykład gdy wymagany jest filtr o dokładnie liniowej fazie, to jedynym poprawnym wyborem będzie filtr FIR.

Jeżeli jednak wymagane jest, aby filtr pracował z bardzo wielką częstotliwością, a dopuszczalna jest niewielka nieliniowość fazy, możemy skłonić się ku filtrom IIR, z ich zredukowaną liczbą operacji mnożenia dla jednej próbki sygnału wyjściowego.

4) Czy są jakieś różnice w budowie filtrów FIR i IIR

Filtr IIR oprócz struktury zapożyczonej z filtru FIR posiada pętlę sprzężenia zwrotnego obejmującą bloki opóźnienia, mnożniki i sumatory

Table of Contents

Zadanie 1.1	1
Zadanie 1.2	1
Zadanie 2	2
Zadanie 3	3
Zadanie 4	4
Zadanie 5	5
Zadanie 6	6
Zadanie 7	7
Zadanie 8	9
zad9	11

Zadanie 1.1

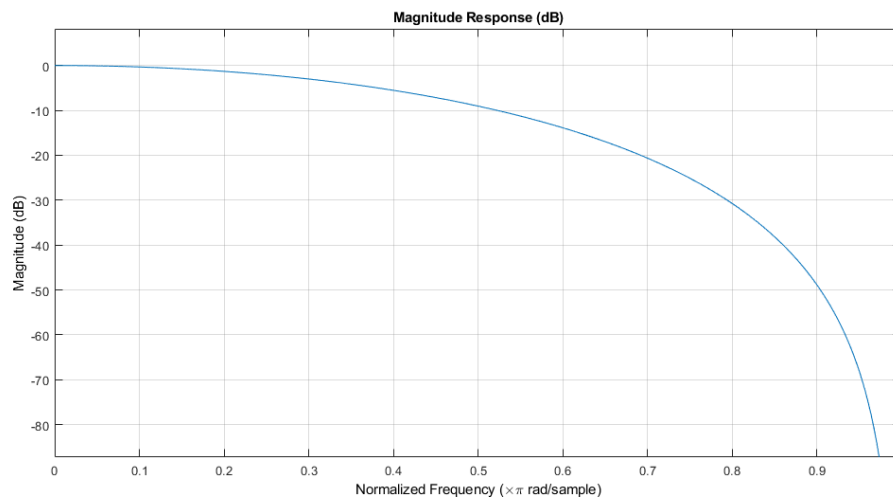
```
[b,a] = maxflat(4,1,0.3)
fvtool(b,a)
```

$b =$

0.1187 0.4749 0.7123 0.4749 0.1187

$a =$

1.0000 0.8995



Zadanie 1.2

```
[b,a] = maxflat(4, 'sym', 0.3)
```

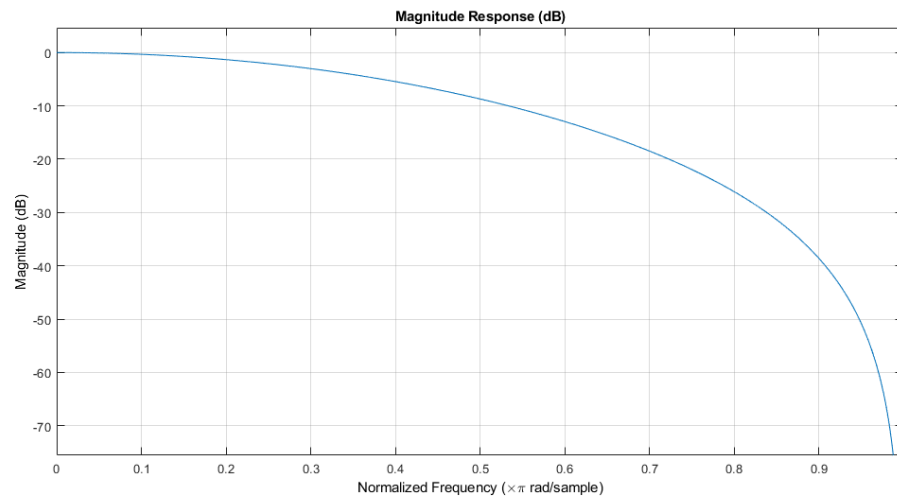
```
fvtool(b,a)
```

```
b =
```

```
0.0331    0.2500    0.4337    0.2500    0.0331
```

```
a =
```

```
1
```



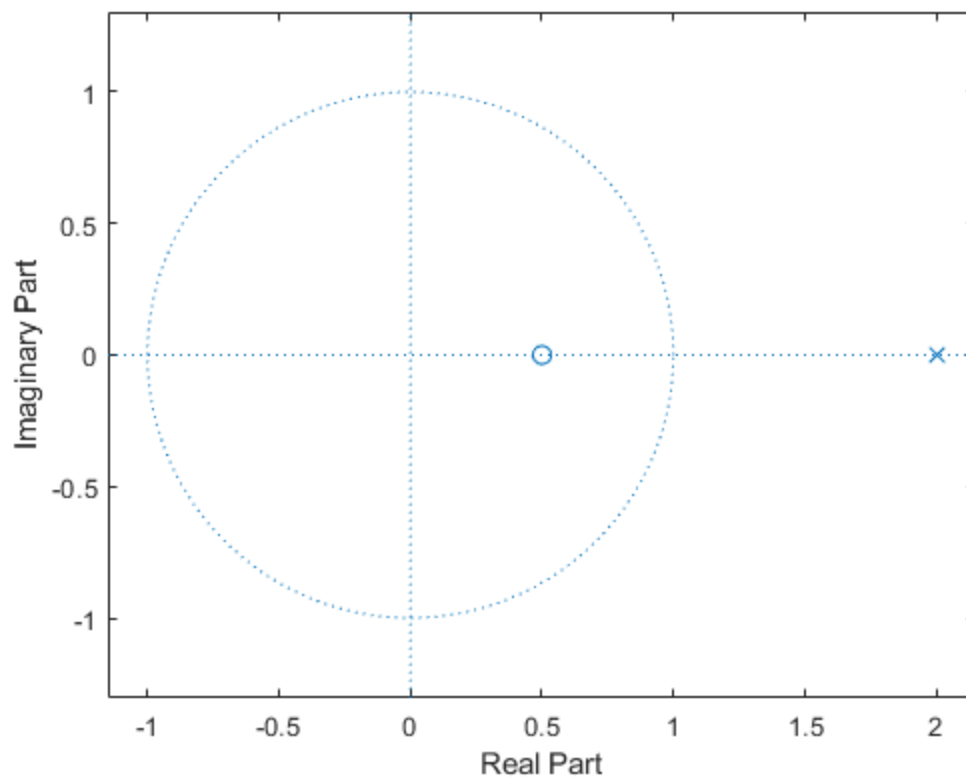
Zadanie 2

```
b = [1 -0.5]; a = [1 -2];  
act_flag1 = isstable(b,a)  
zplane(b,a)
```

```
act_flag1 =
```

```
logical
```

```
0
```



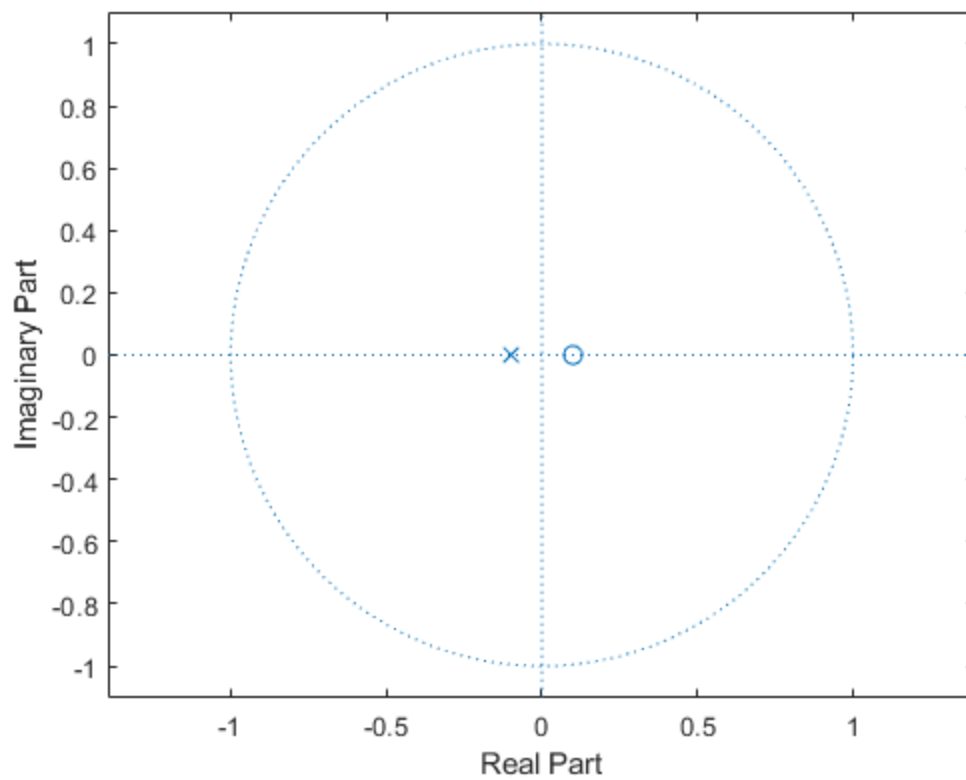
Zadanie 3

```
b = [1 -0.1]; a = [-1 -0.1];  
act_flag1 = isstable(b,a)  
zplane(b,a)
```

```
act_flag1 =
```

```
logical
```

```
1
```



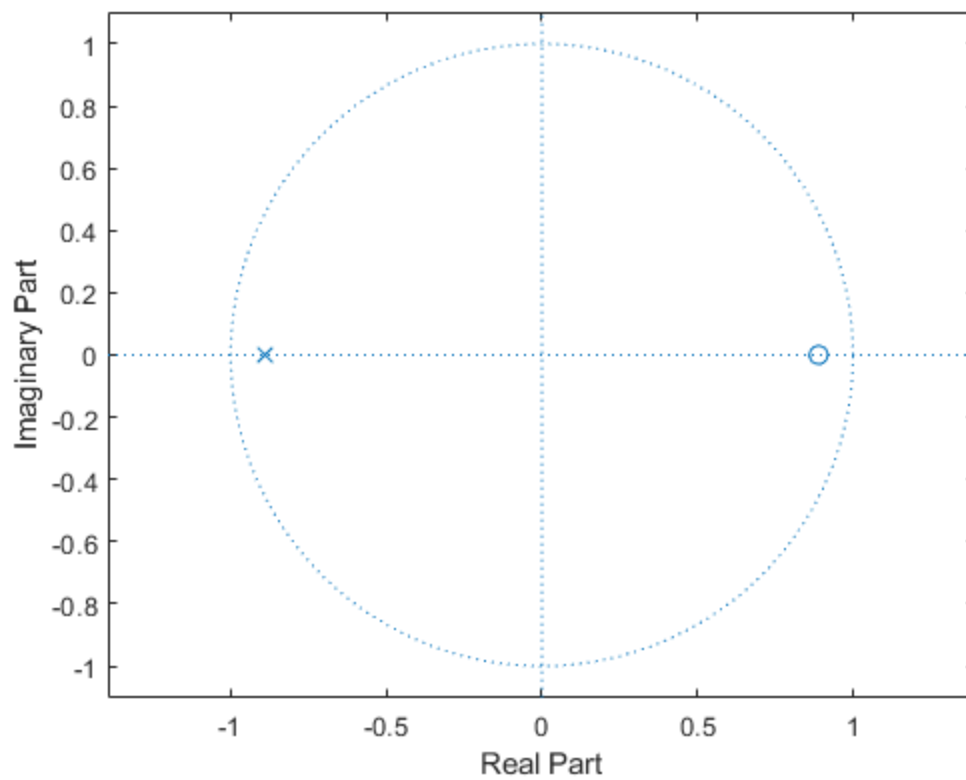
Zadanie 4

```
b = [0.9 -0.8]; a = [-0.9 -0.8];  
act_flag1 = isstable(b,a)  
zplane(b,a)
```

```
act_flag1 =
```

```
logical
```

```
1
```



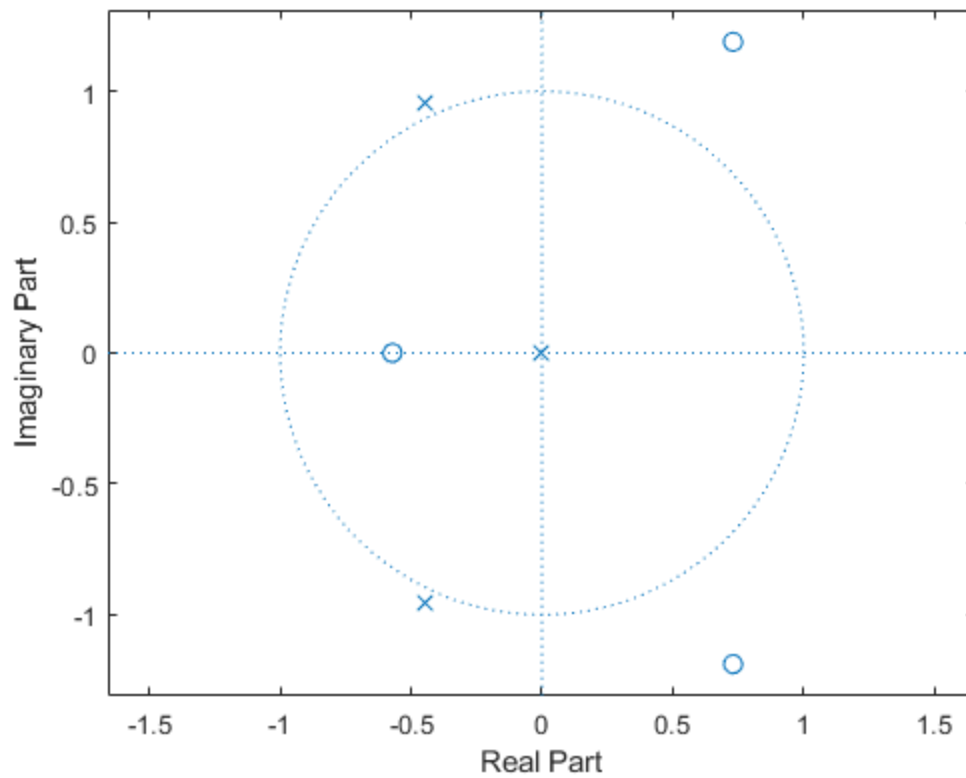
Zadanie 5

```
b = [0.9 -0.8 1 1]; a = [-0.9 -0.8 -1];  
act_flag1 = isstable(b,a)  
zplane(b,a)
```

```
act_flag1 =
```

```
logical
```

```
0
```



Zadanie 6

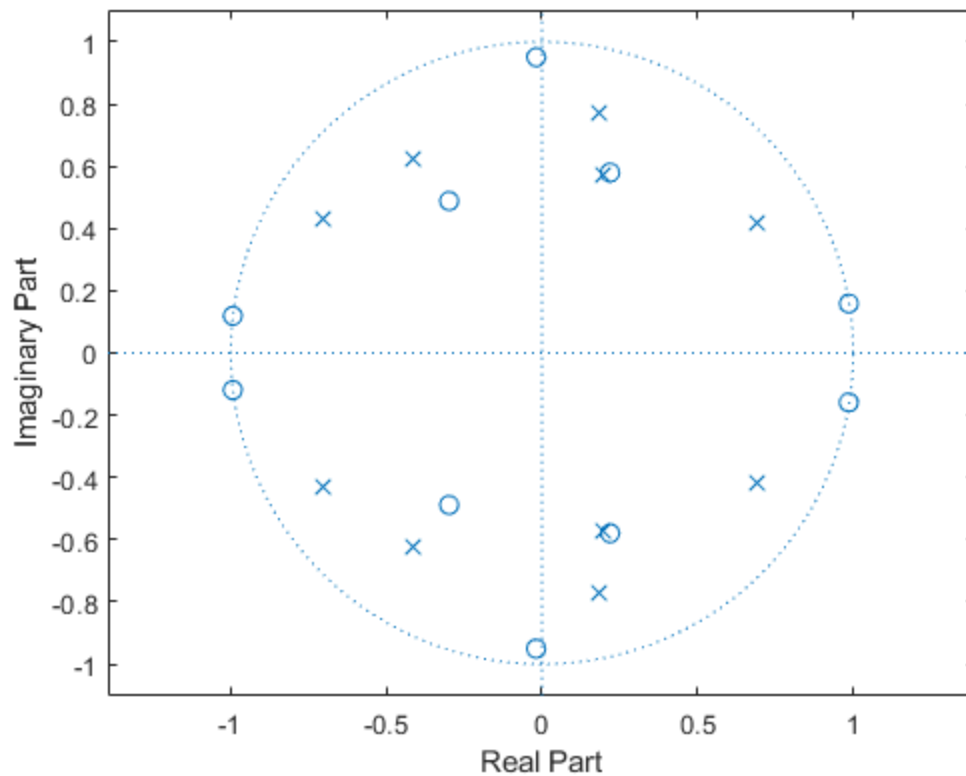
```
m = [0 0 1 1 1 0 1 1 0 0];  
f = [0 0.1 0.2 0.3 0.4 0.5 0.7 0.8 0.9 1];  
[b,a] = yulewalk(10,f,m);
```

```
act_flag1 = isstable(b,a)  
zplane(b,a)
```

```
act_flag1 =
```

```
logical
```

```
1
```

Zadanie 7

```
[b,a] = maxflat(4,1,0.3);

% act_flag1 = isstable(b,a)
% zplane(b,a)

% wynik flag = 0

load chirp
t = (0:length(y)-1)/Fs;
% 1.6 sekundy
xfft=abs(fft(y));
xfft=xfft/13129;

x1=1:1:6564;

% bar(x1(1:6564), xfft(1:6564));
% axis([0,6564, 0,0.01]) ;

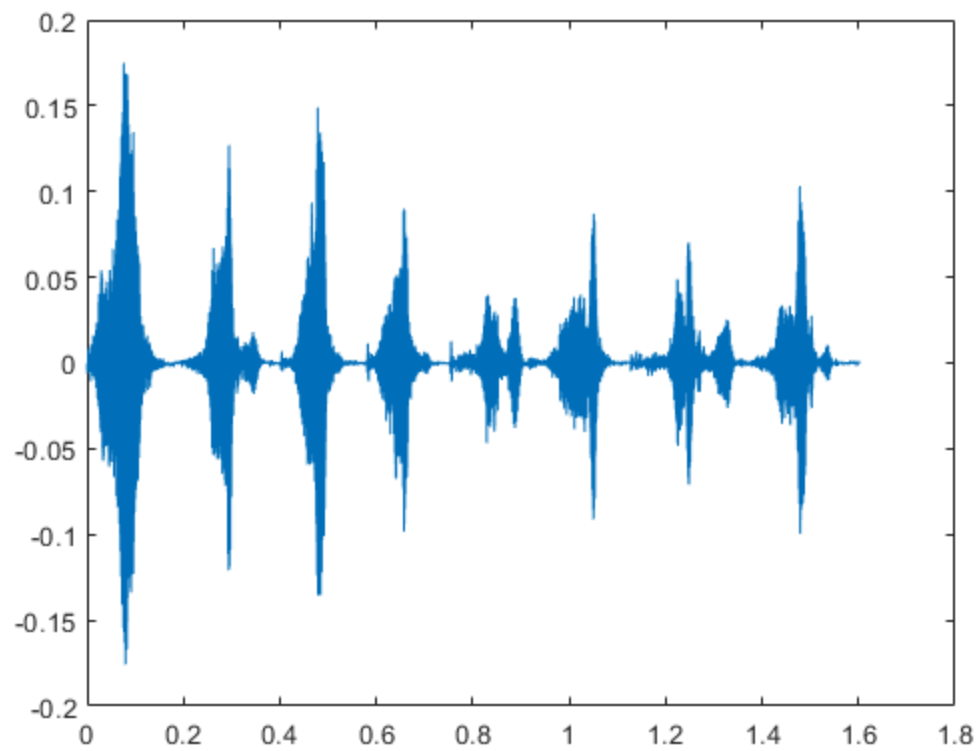
% plot(t,y);
% fvtool(b,a)

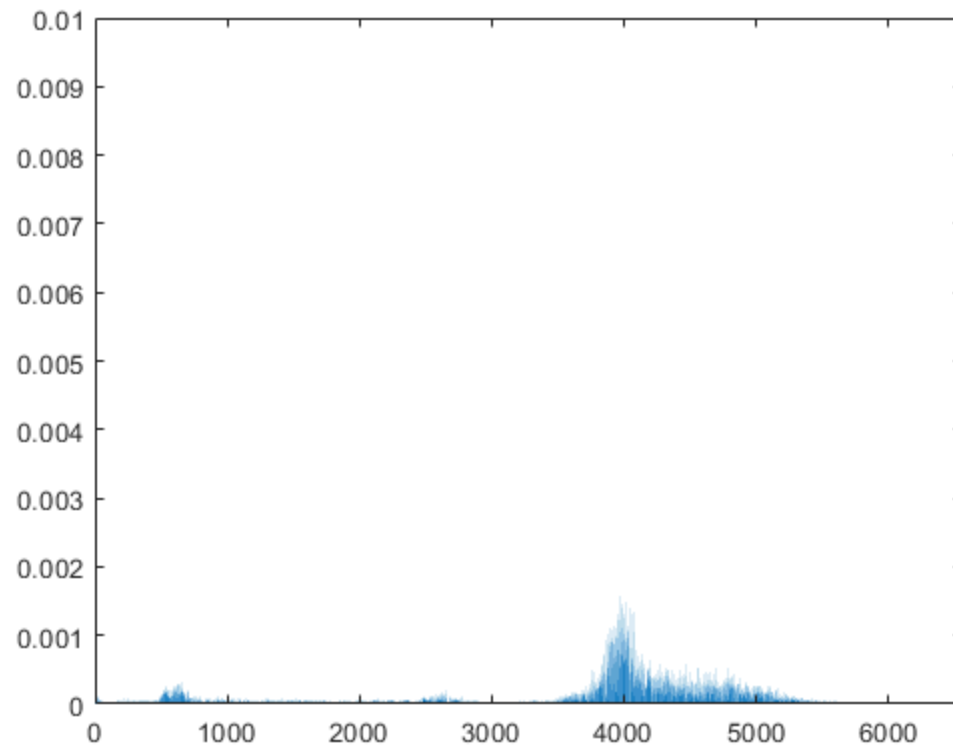
outsignal = filter(b,a,y);
```

```
figure()
plot(t, outsignal);

xfft=abs(fft(outsignal));
xfft=xfft/13129;
x1=1:1:6564;

figure()
bar(x1(1:6564), xfft(1:6564));
axis([0,6564, 0,0.01]) ;
```





Zadanie 8

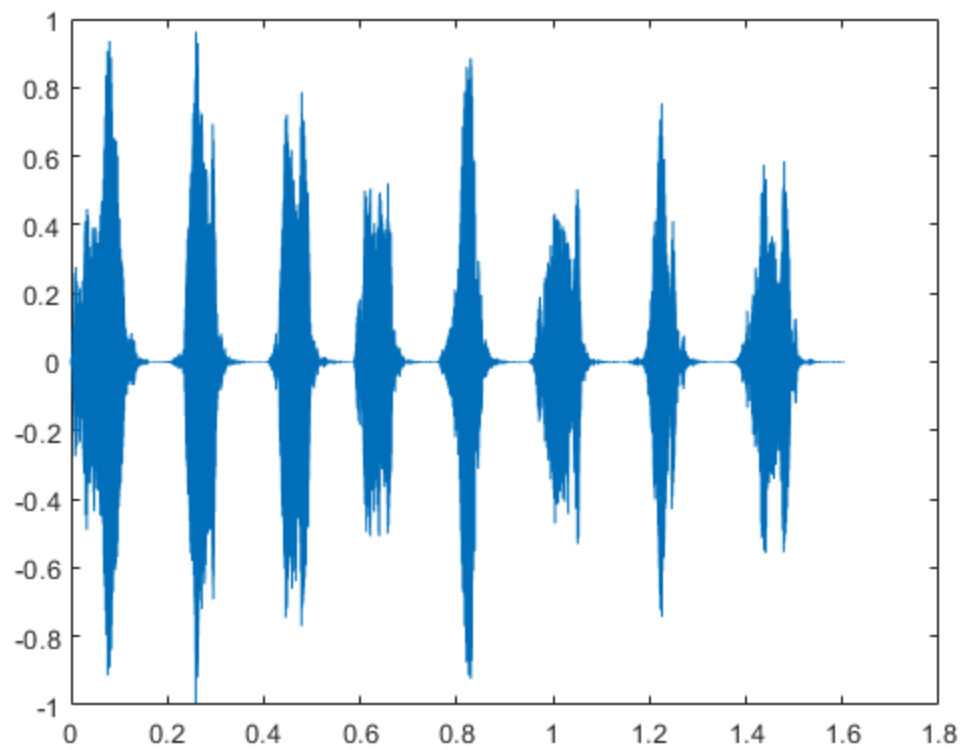
```
m = [0 0 0 0 0 1 1 1 1 1];  
f = [0 0.1 0.2 0.3 0.4 0.5 0.7 0.8 0.9 1];  
[b,a] = yulewalk(10,f,m);  
  
% act_flag1 = isstable(b,a)  
% zplane(b,a)  
  
% wynik flag = 0  
  
load chirp  
t = (0:length(y)-1)/Fs;  
% 1.6 sekundy  
xfft=abs(fft(y));  
xfft=xfft/13129;  
  
x1=1:1:6564;  
  
% bar(x1(1:6564), xfft(1:6564));  
% axis([0,6564, 0,0.01]) ;  
  
% plot(t,y);  
% fvtool(b,a)
```

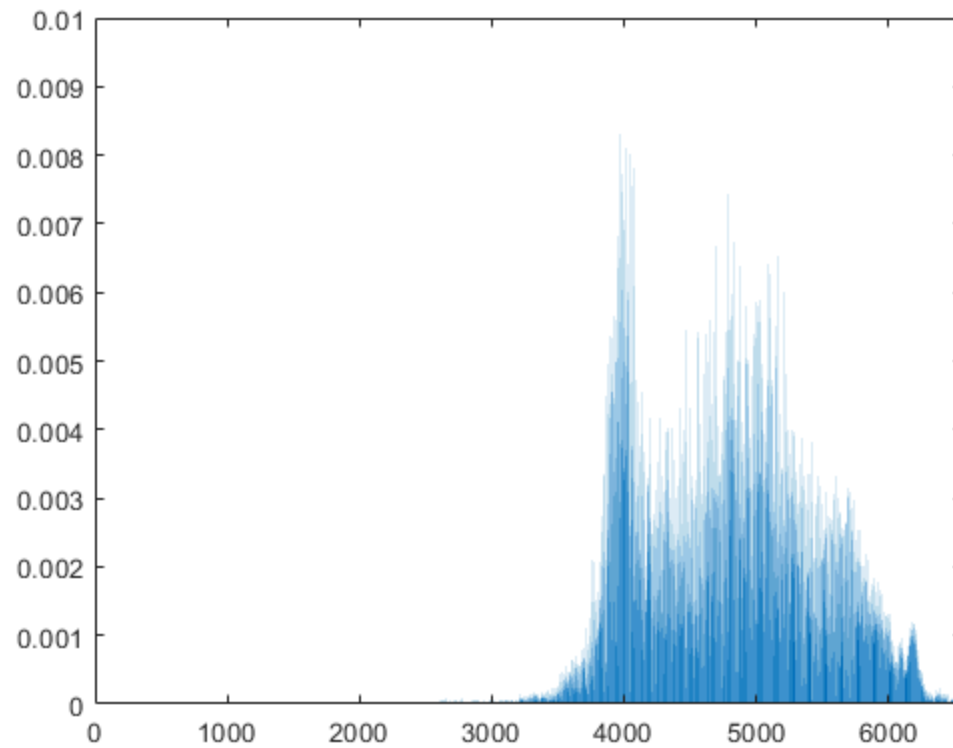
```
outsignal = filter(b,a,y);

figure()
plot(t, outsignal);

xfft=abs(fft(outsignal));
xfft=xfft/13129;
x1=1:1:6564;

figure()
bar(x1(1:6564), xfft(1:6564));
axis([0,6564, 0,0.01]) ;
```





zad9

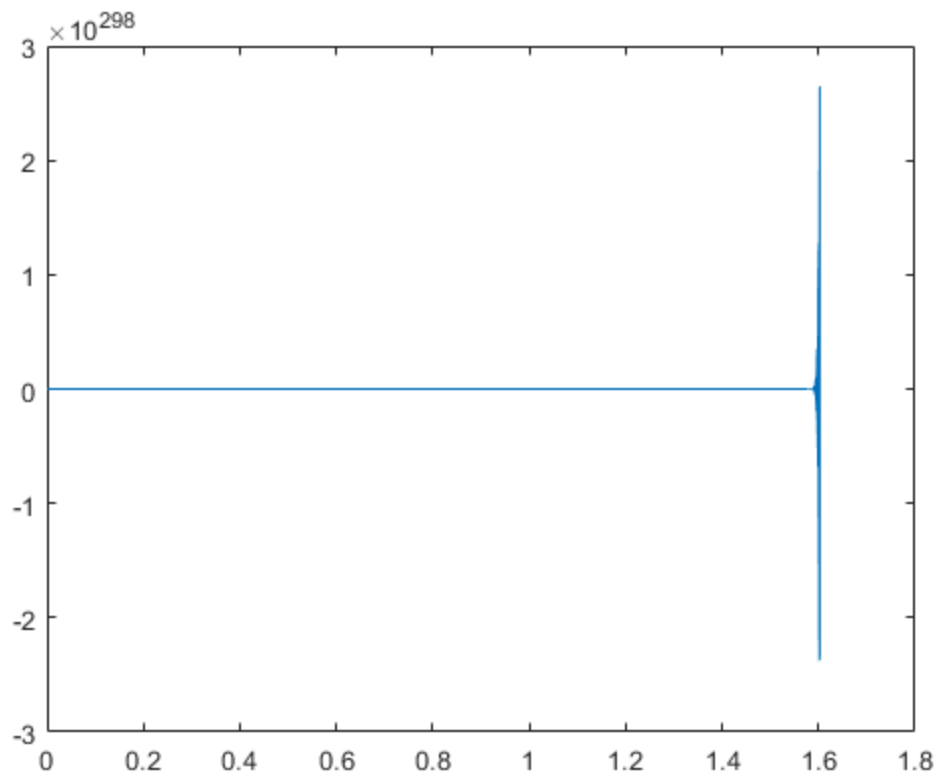
```
b = [0.9 -0.8 1 1]; a = [-0.9 -0.8 -1];  
% niestabilny  
  
% act_flag1 = isstable(b,a)  
% zplane(b,a)  
  
% wynik flag = 0  
  
load chirp  
t = (0:length(y)-1)/Fs;  
% 1.6 sekundy  
xfft=abs(fft(y));  
xfft=xfft/13129;  
  
x1=1:1:6564;  
  
% bar(x1(1:6564), xfft(1:6564));  
% axis([0,6564, 0,0.01]) ;  
  
% plot(t,y);  
% fvtool(b,a)
```

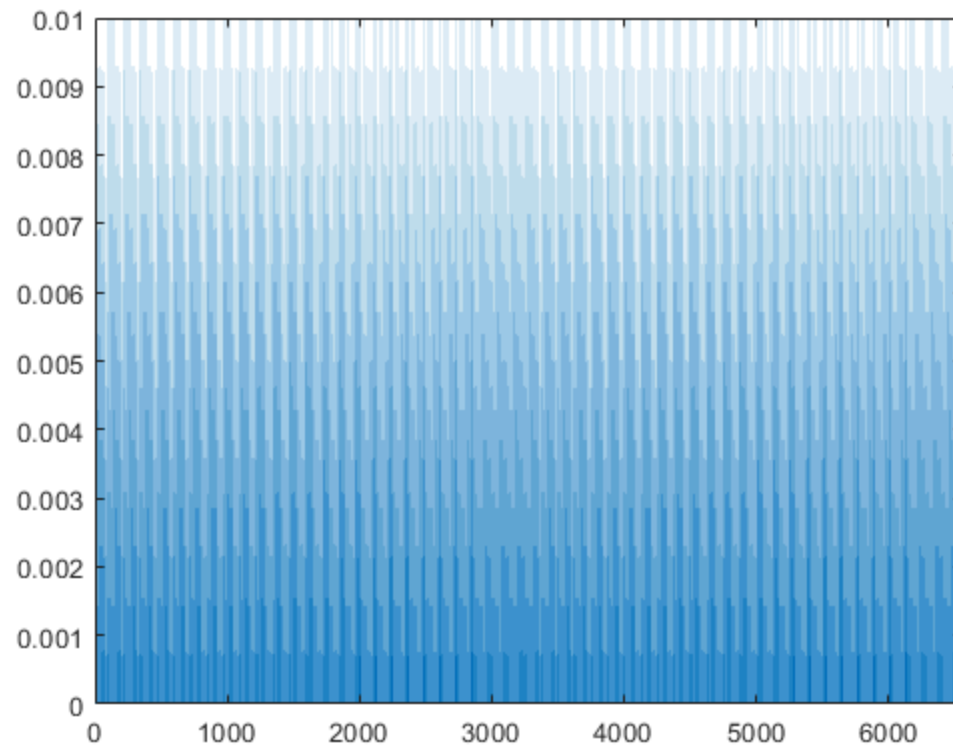
```
outsignal = filter(b,a,y);

figure()
plot(t, outsignal);

xfft=abs(fft(outsignal));
xfft=xfft/13129;
x1=1:1:6564;

figure()
bar(x1(1:6564), xfft(1:6564));
axis([0,6564, 0,0.01]) ;
```





Published with MATLAB® R2019a