

SPRAWOZDANIE 11

Zajęcia nr 12 - Filtry IIR

Michał Midor, gr. 5 - 10.12.2025, godz: 16:45

WSTĘP

W dziedzinie cyfrowego przetwarzania sygnałów, filtr cyfrowy jest kluczowym elementem systemów komunikacyjnych i pomiarowych. Jego zadaniem jest modyfikacja widma sygnału, najczęściej poprzez tłumienie niepożądanych częstotliwości (szumu) lub separowanie składowych sygnału. Filtry cyfrowe wykazują szereg istotnych zalet w stosunku do swoich analogowych odpowiedników, w tym:

- **Precyzja:** Oferują powtarzalną, wysoką precyzję, niezależną od tolerancji komponentów.
- **Stabilność:** Ich charakterystyki nie dryfują z powodu starzenia się komponentów czy zmian temperatury.
- **Elastyczność:** Mogą realizować bardziej złożone i precyzyjne charakterystyki bez konieczności kosztownej rekonfiguracji sprzętu.

Projektowanie filtrów IIR polega na wyznaczeniu współczynników licznika b i mianownika a oraz ich transmitancji $H(z)$, często poprzez digitalizację znanych filtrów analogowych (np. Butterwortha, Eliptycznego) za pomocą transformacji biliniowej. Alternatywnie, stosuje się metody aproksymacji pożądanej charakterystyki, takie jak Yule-Walker.

Filtry cyfrowe dzieli się na dwie główne kategorie różniące się istotnie budową i właściwościami:

- Filtr FIR (Finite Impulse Response) - budowa bez sprzężenia zwrotnego, odpowiedź impulsowa jest skończona (zanika po czasie równym długości filtru, czyli zachowanych próbek z przeszłości), możliwa jest faza liniowa bez zniekształceń czasowych, zawsze jest stabilny przy skończonych współczynnikach, lecz oferuje niższą wydajność
- Filtr IIR (Infinite Impulse Response) - budowa ze sprzężeniem zwrotnym (rekursywna), odpowiedź impulsowa jest teoretycznie nieskończona, wprowadza fazę nieliniową ze zniekształceniami czasowymi, wymaga weryfikacji stabilności (wszystkie bieguny muszą leżeć wewnątrz okręgu jednostkowego na płaszczyźnie Z , lecz oferuje wyższą wydajność (osiąga daną stromość przy niższym rzędzie).

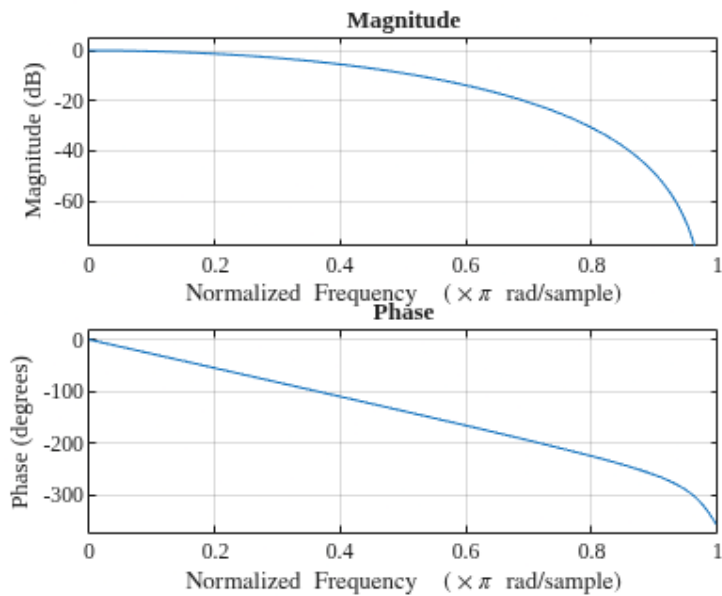
IIR jest wskazany, gdy: Krytyczna jest minimalizacja złożoności obliczeniowej lub pamięci (wymagany jest niski rząd filtru N) oraz gdy nieliniowość fazy jest dopuszczalna.

FIR jest wskazany, gdy absolutnie wymagana jest liniowa faza (np. w aplikacjach telekomunikacyjnych, pomiarowych, obróbce obrazu) lub gdy stabilność jest krytycznym wymogiem.

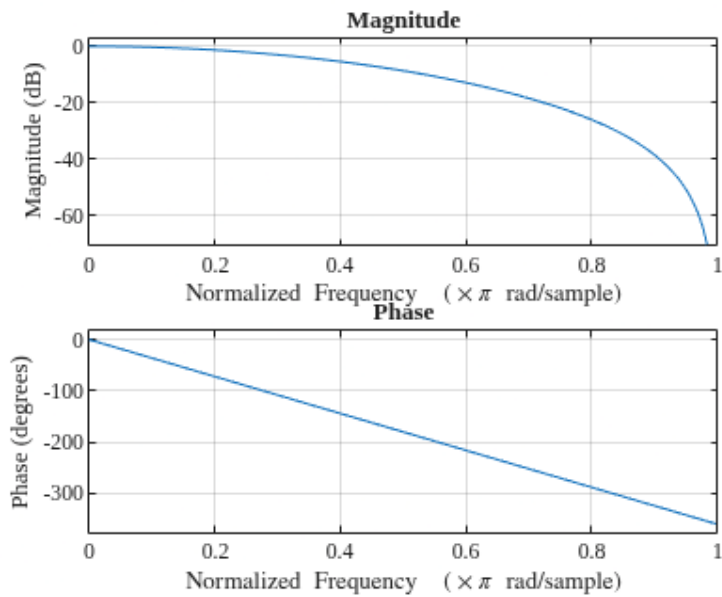
WYKONANIE ĆWICZEŃ

1)

```
[b, a] = maxflat(4, 1, 0.3);  
[b_sym, a_sym] = maxflat(4, 'sym', 0.3); % sym oznacza liniową fazę  
  
freqz(b, a)
```



```
freqz(b_sym, a_sym);
```

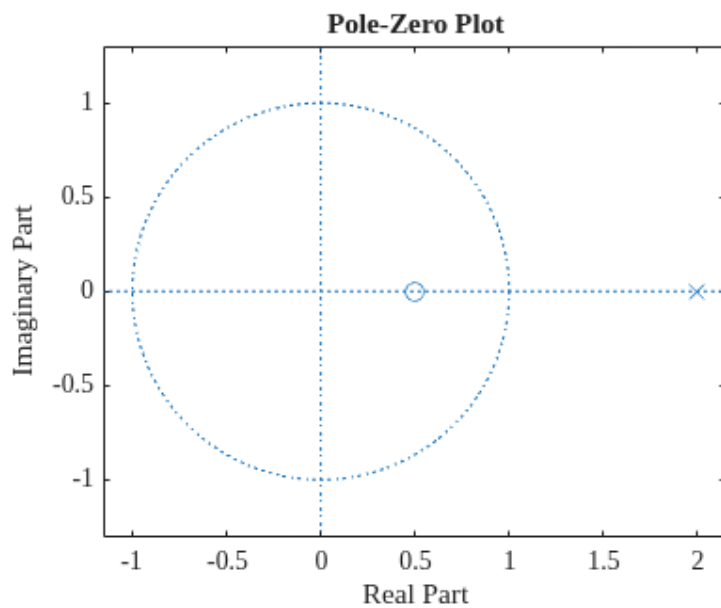


2)

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

```
flag = logical
0
```

```
zplane(b, a)
```



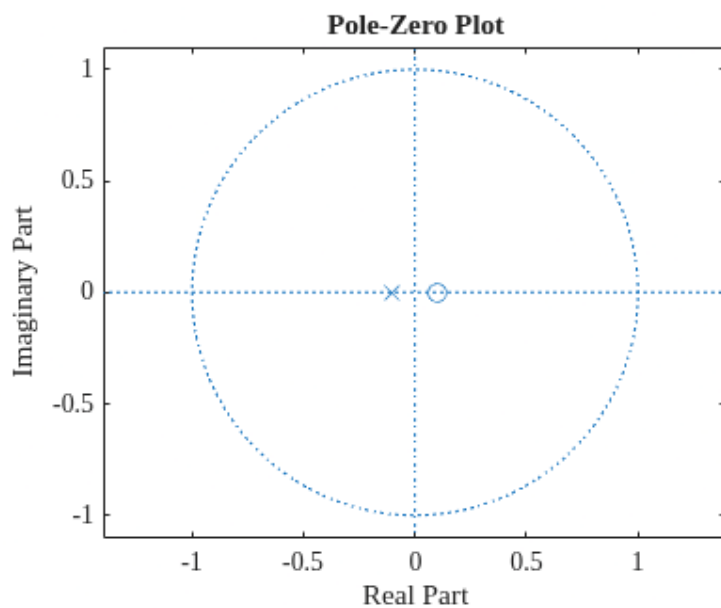
Ten filtr IIR nie jest stabilny.

3)

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

```
flag = logical
      1
```

```
zplane(b, a);
```



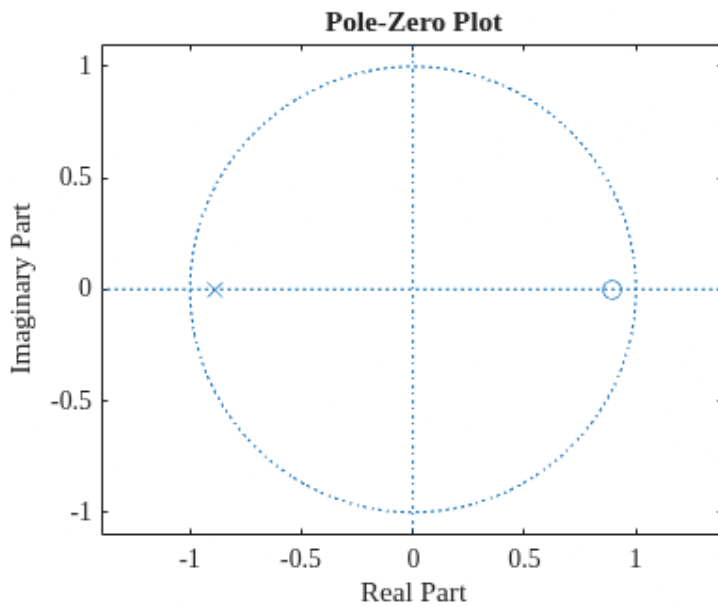
Ten filtr IIR jest stabilny.

4)

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

```
flag = logical  
1
```

```
zplane(b, a);
```



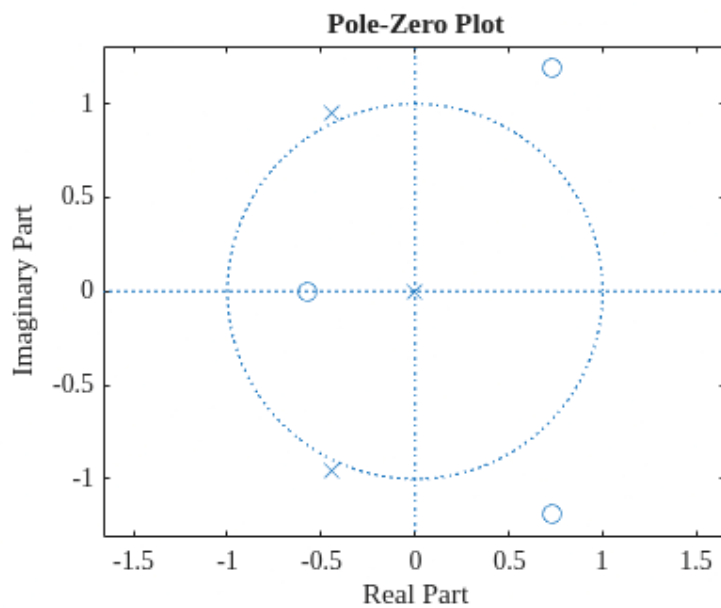
Ten filtr IIR jest stabilny.

5)

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

```
flag = logical  
0
```

```
zplane(b, a);
```



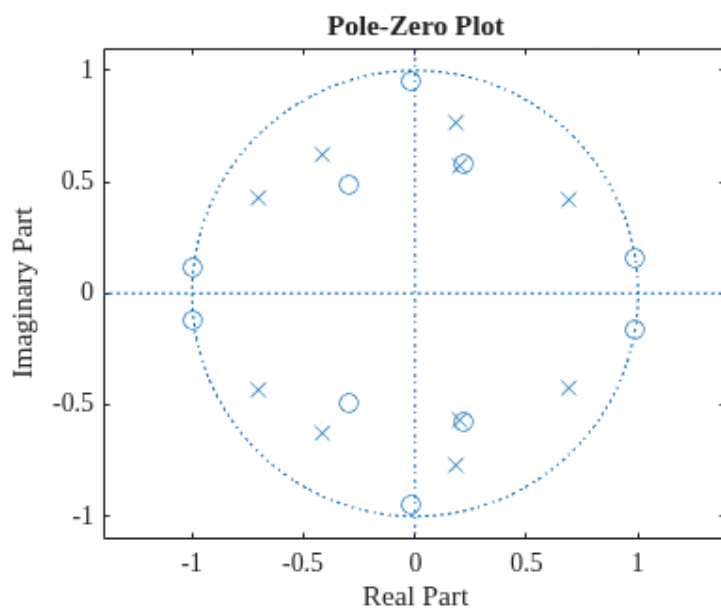
Ten filtr IIR nie jest stabilny

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);
flag = isstable(b, a)
```

```
flag = logical
      1
```

```
zplane(b, a)
```



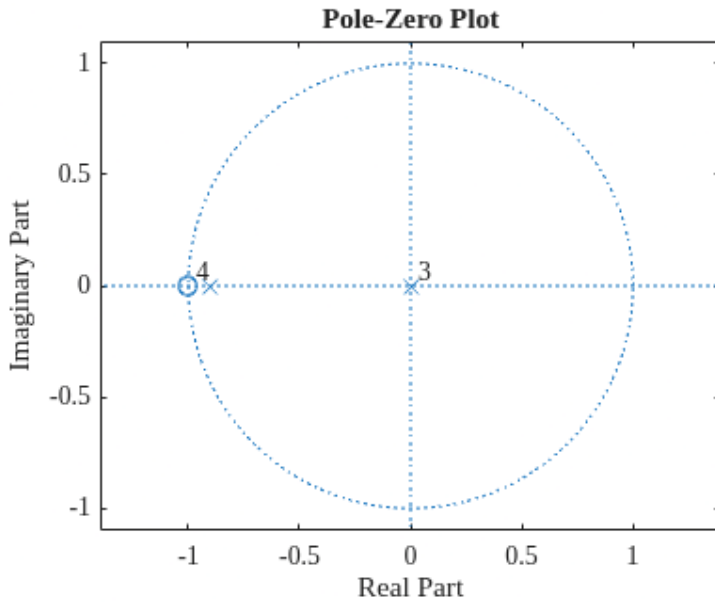
Ten filtr IIR jest stabilny.

7)

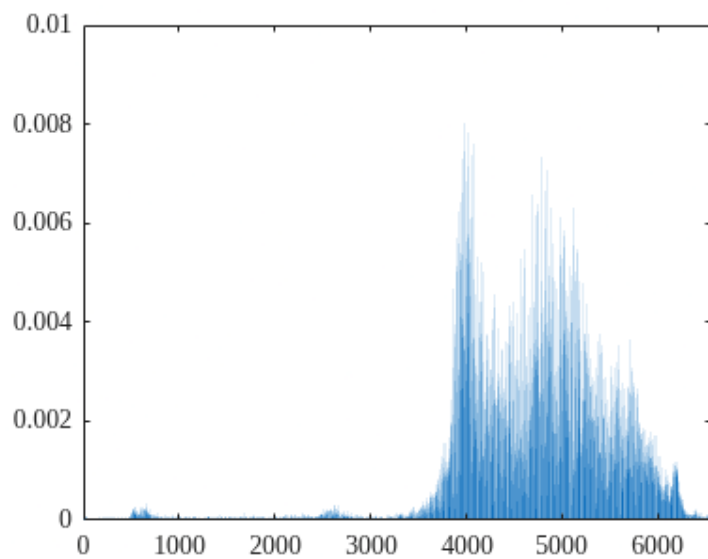
```
[b, a] = maxflat(4, 1, 0.3);  
flag = isstable(b, a)
```

```
flag = logical  
      1
```

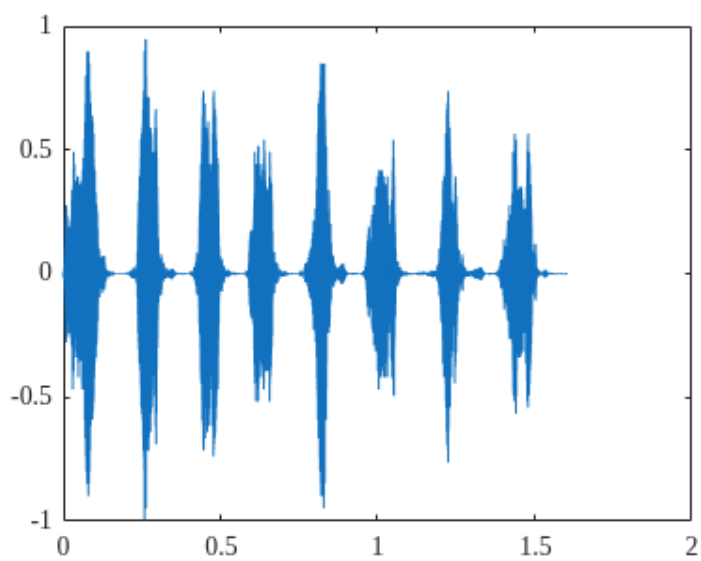
```
zplane(b, a) % stabilny
```



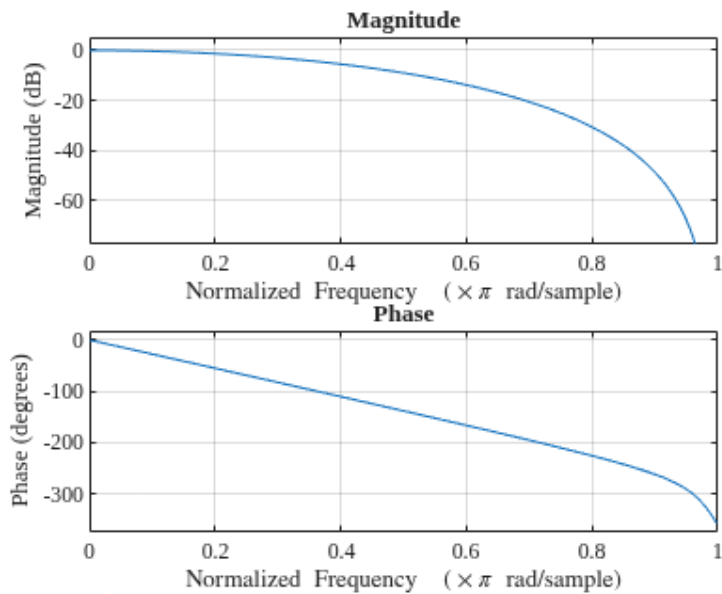
```
load chirp  
t = (0:length(y)-1)/Fs; % 1.6 sekundy  
  
xfft=abs(fft(y));  
xfft=xfft/13129;  
x1=1:1:6564;  
bar(x1(1:6564), xfft(1:6564)); % sygnał oryginalny - dziedziła częstotliwości  
axis([0,6564, 0,0.01]) ;
```



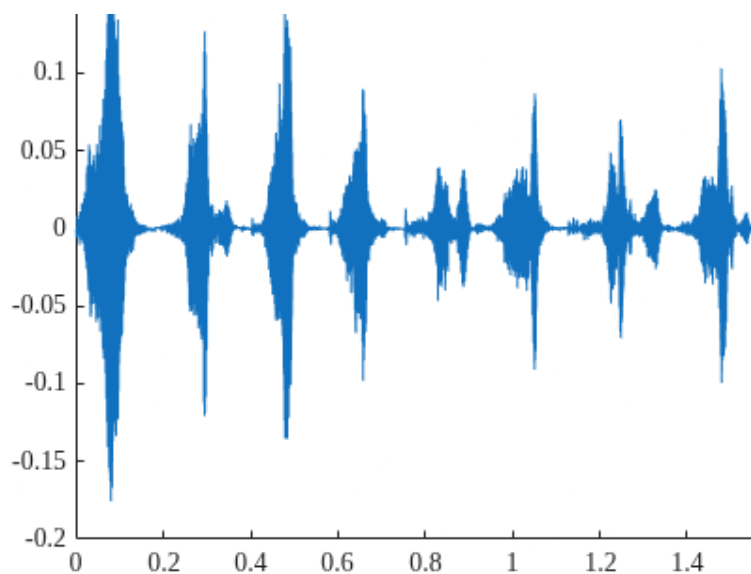
```
plot(t,y); % sygnał oryginalny - dziedzina czasu
```



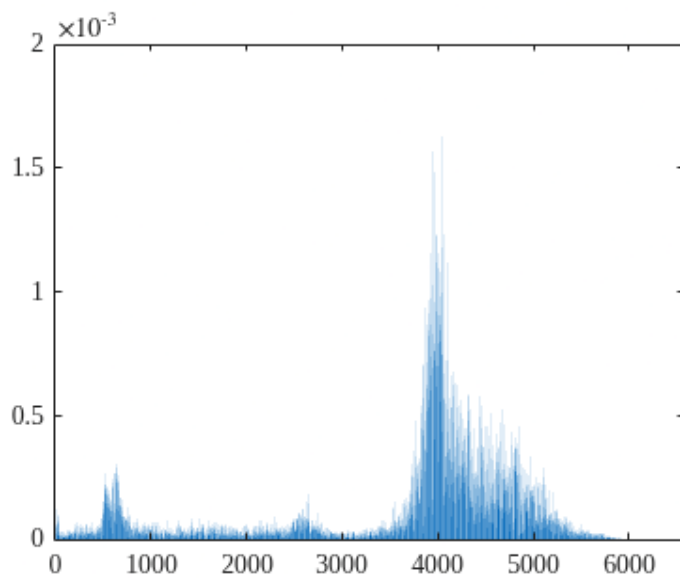
```
freqz(b, a)
```



```
outsignal = filter(b,a,y);
plot(t, outsignal) % sygnał przefiltrowany - dziedzina czasu
```



```
xfft=abs(fft(outsignal));
xfft=xfft/13129;
x1=1:1:6564;
bar(x1(1:6564), xfft(1:6564)); % sygnał przelitrowany - dziedzina
częstotliwości
axis([0,6564, 0,0.002]);
```

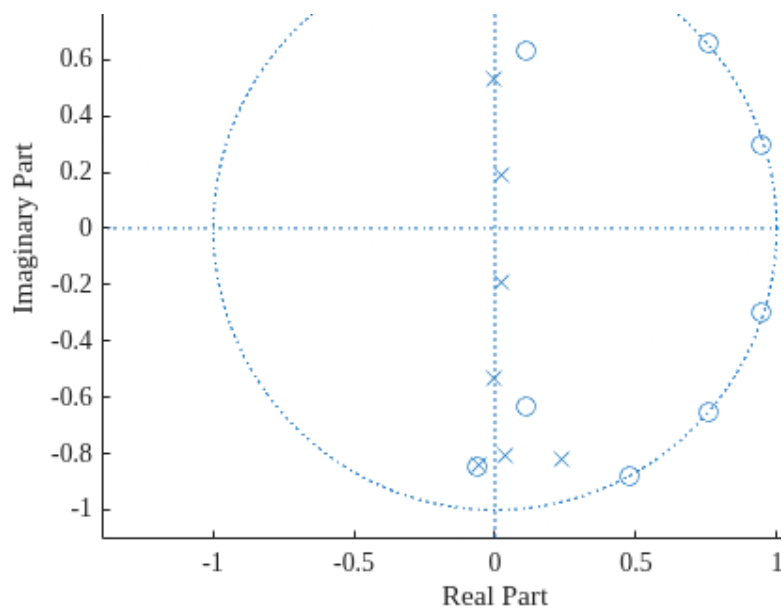



8)

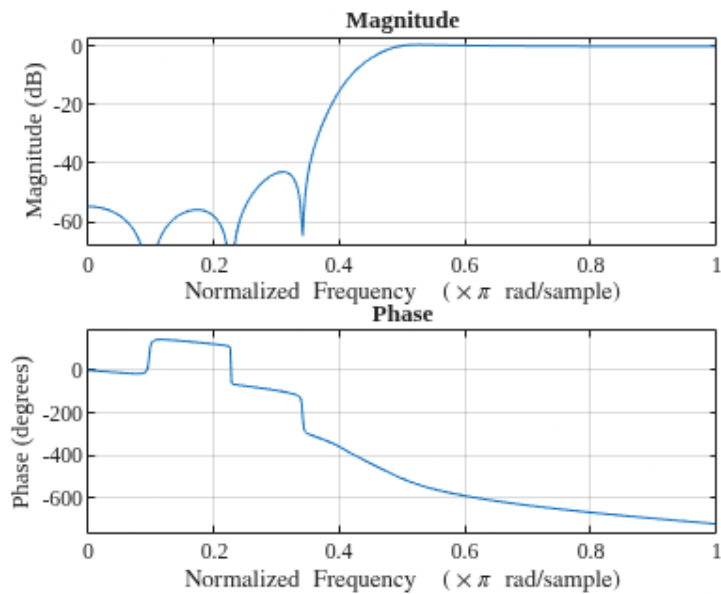
```
m = [0 0 0 0 0 1 1 1 1 1];
f = [0 0.1 0.2 0.3 0.4 0.5 0.7 0.8 0.9 1];
[b,a] = yulewalk(10,f,m);
flag = isstable(b, a)
```

```
flag = logical
      1
```

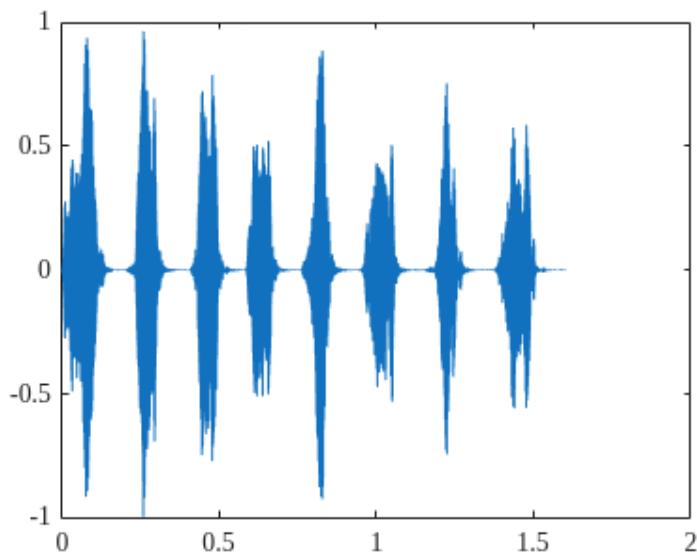
```
zplane(b, a) % jest stabilny
```



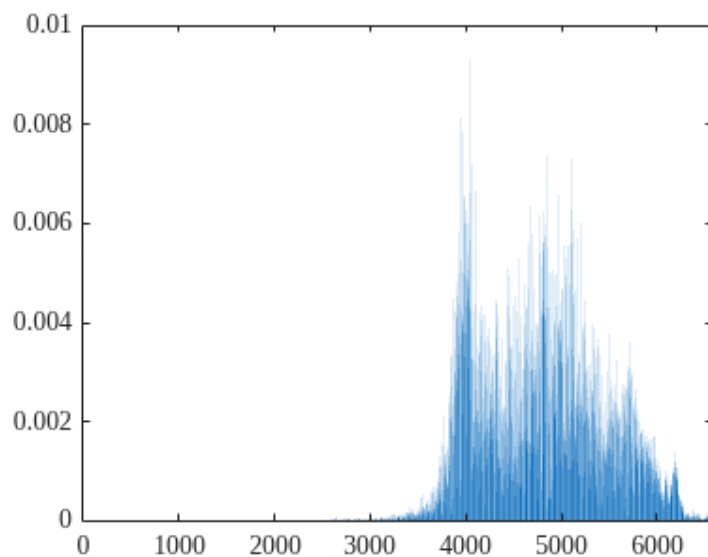
```
freqz(b, a)
```



```
outsignal = filter(b,a,y);
plot(t, outsignal) % sygnał przefiltrowany - dziedzina czasu
```



```
xfft=abs(fft(outsignal));
xfft=xfft/13129;
x1=1:1:6564;
bar(x1(1:6564), xfft(1:6564)); % sygnał przelitrowany - dziedzina
częstotliwości
axis([0,6564, 0,0.01]);
```

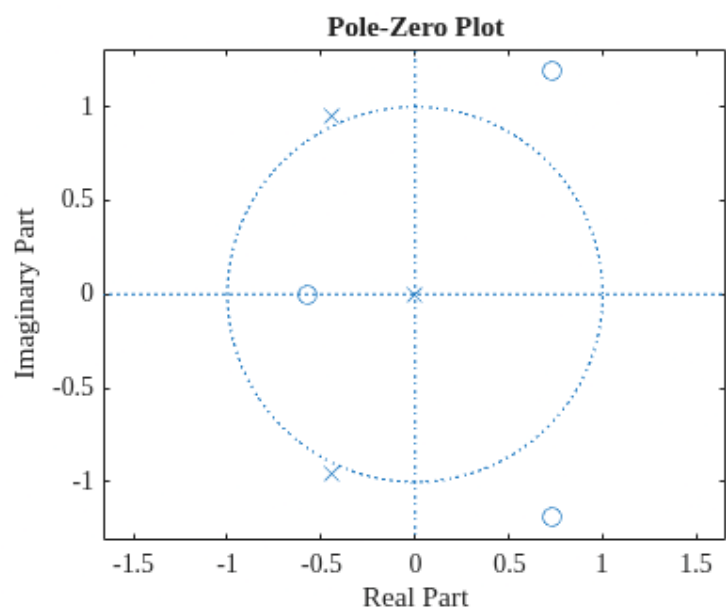


9)

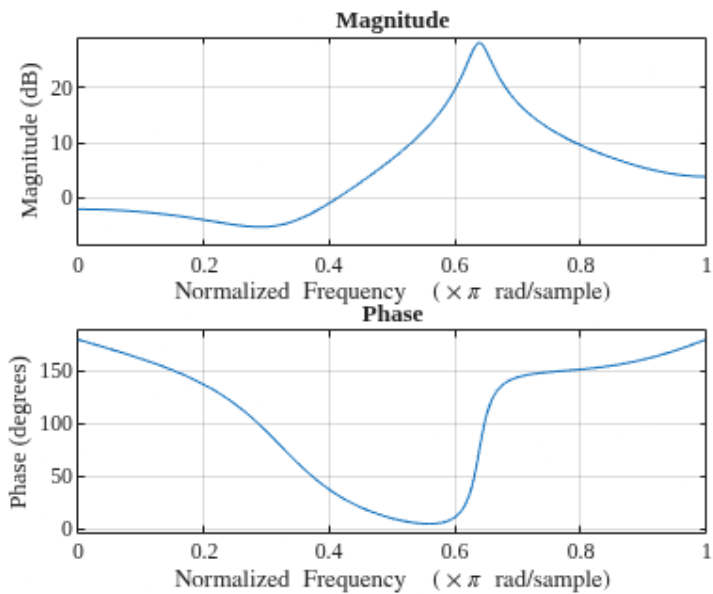
```
b = [0.9 -0.8 1 1];
a = [-0.9 -0.8 -1];
flag = isstable(b, a)
```

```
flag = logical
      0
```

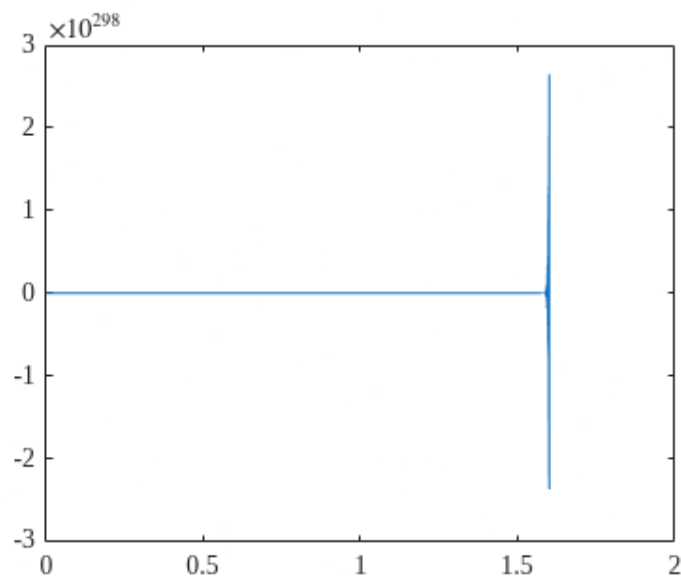
```
zplane(b, a) % niestabilny
```



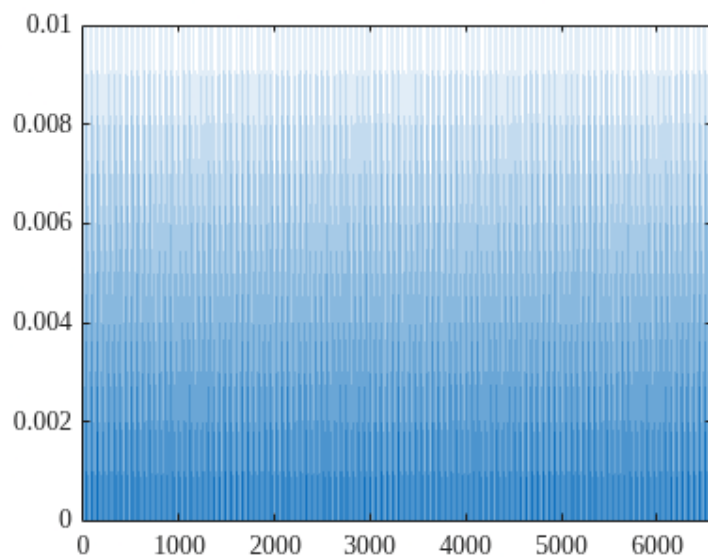
```
freqz(b, a)
```



```
outsignal = filter(b,a,y);
plot(t, outsignal) % sygnał przefiltrowany - dziedzina czasu
```

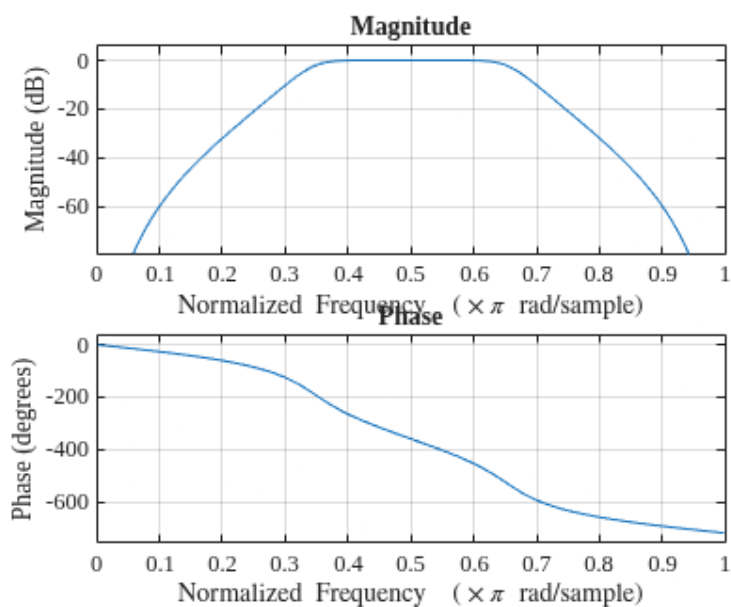


```
xfft=abs(fft(outsignal));
xfft=xfft/13129;
x1=1:1:6564;
bar(x1(1:6564), xfft(1:6564)); % sygnał przelitrowany - dziedzina
częstotliwości
axis([0,6564, 0,0.01]);
```



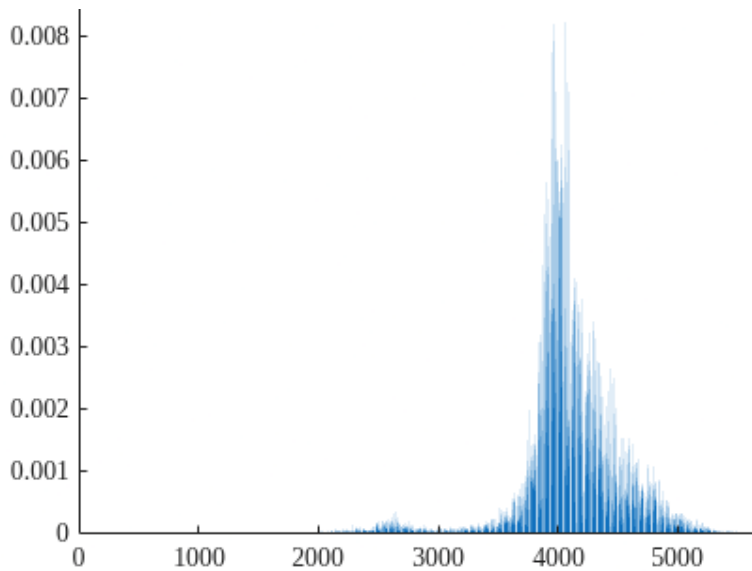
10)

```
[n,Wn] = buttord([2000 3000]/5000,[500 4500]/5000,1,60);
[b,a] = butter(n,Wn);
freqz(b, a)
```



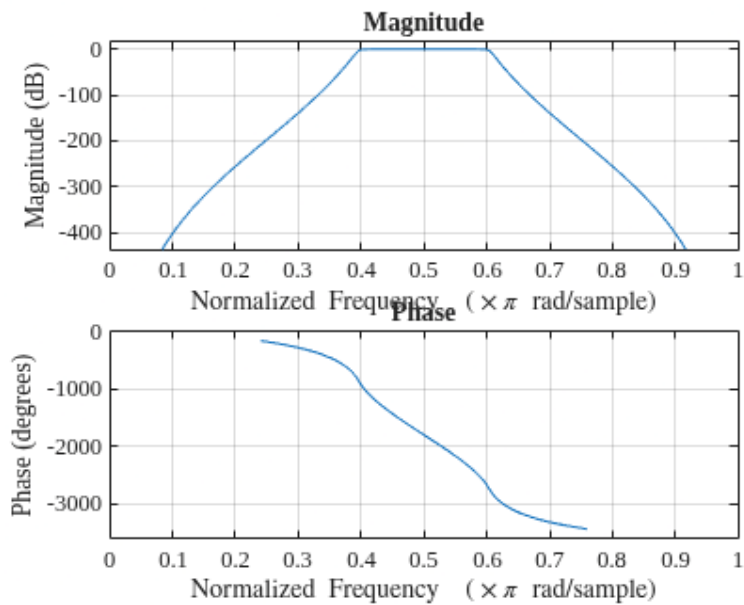
```
outsignal = filter(b,a,y);

xfft=abs(fft(outsignal));
xfft=xfft/13129;
x1=1:1:6564;
bar(x1(1:6564), xfft(1:6564)); % sygnał przelitrowany - dziedzina
częstotliwości
axis([0,6564, 0,0.01]);
```



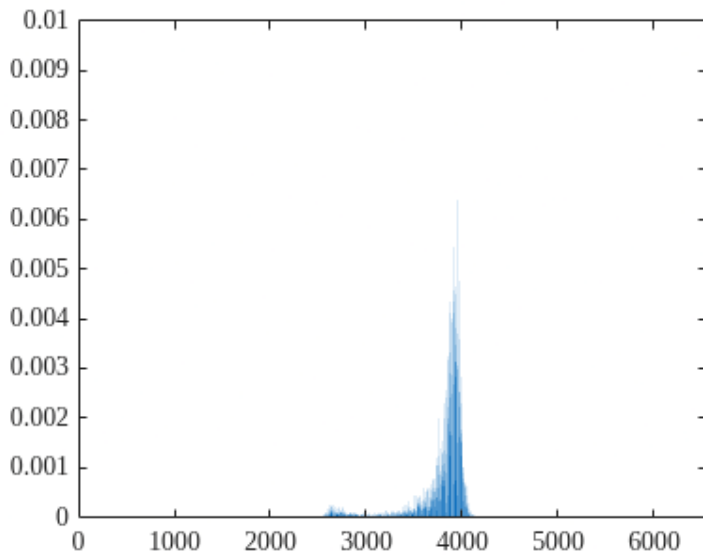
11)

```
[n,Wn] = buttord([2000 3000]/5000,[1800 3200]/5000,1,60);
[b,a] = butter(n,Wn);
freqz(b, a)
```



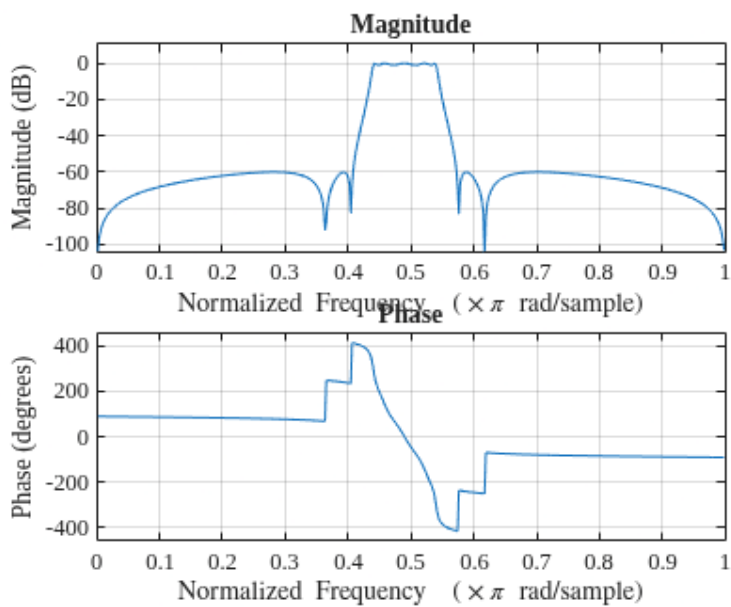
```
outsignal = filter(b,a,y);

xfft=abs(fft(outsignal));
xfft=xfft/13129;
x1=1:1:6564;
bar(x1(1:6564), xfft(1:6564)); % sygnał przelitrowany - dziedzina
częstotliwości
axis([0,6564, 0,0.01]);
```



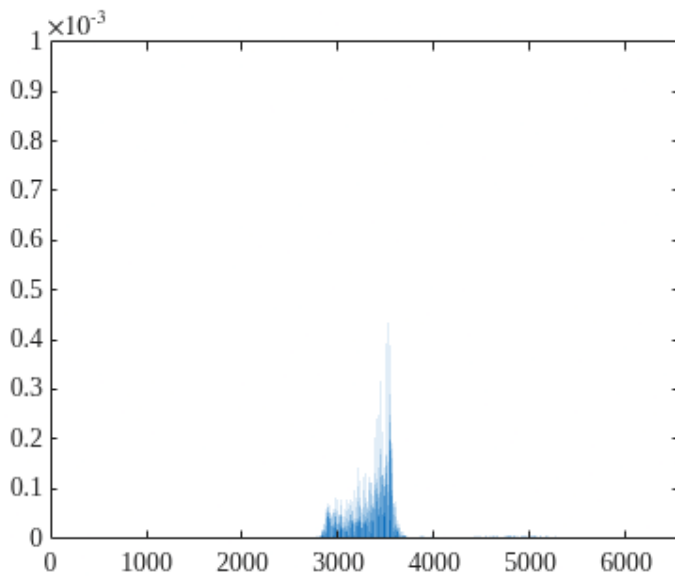
12)

```
[n,Wn] = ellipord([2200 2700]/5000,[2000 2900]/5000,1,60);
[b,a] = ellip(n, 1, 60, Wn);
freqz(b, a)
```



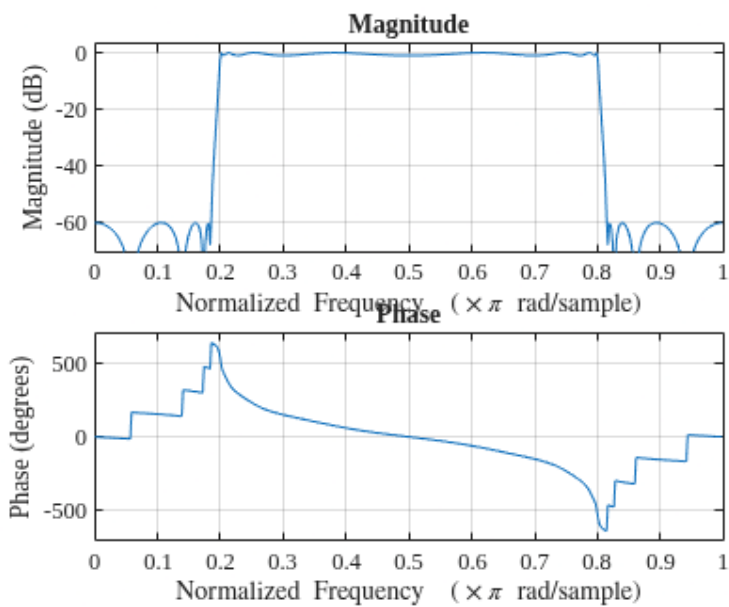
```
outsignal = filter(b,a,y);

xfft=abs(fft(outsignal));
xfft=xfft/13129;
x1=1:1:6564;
bar(x1(1:6564), xfft(1:6564)); % sygnał przelitrowany - dziedzina
częstotliwości
axis([0,6564, 0,0.001]);
```



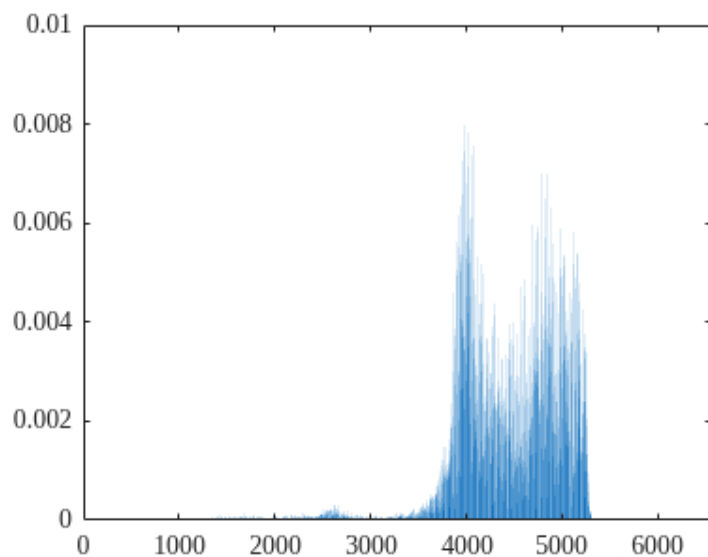
13)

```
[n,Wn] = ellipord([1000 4000]/5000,[900 4100]/5000,1,60);
[b,a] = ellip(n, 1, 60, Wn);
freqz(b, a)
```



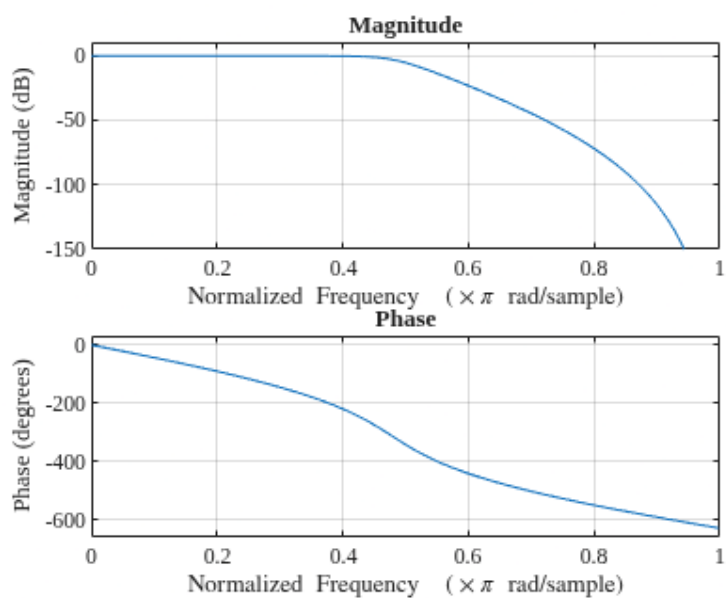
```
outsignal = filter(b,a,y);

xfft=abs(fft(outsignal));
xfft=xfft/13129;
x1=1:1:6564;
bar(x1(1:6564), xfft(1:6564)); % sygnał przelitrowany - dziedzina
częstotliwości
axis([0,6564, 0,0.01]);
```

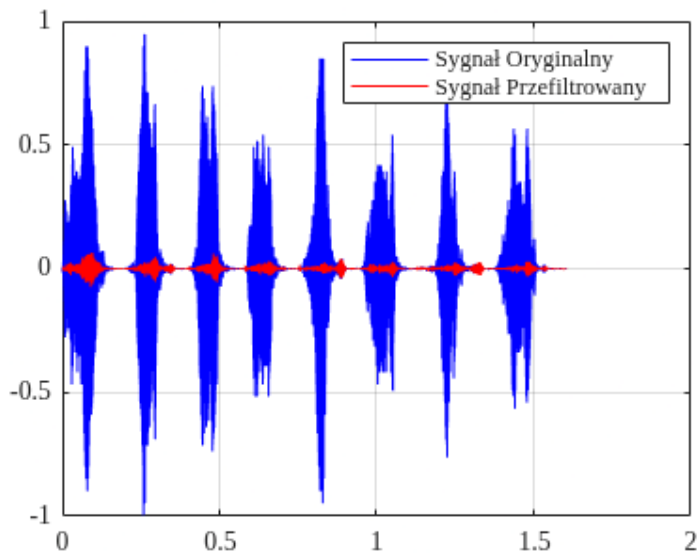
14)

```
[zhi,phi,khi] = butter(7,0.48,'low');
soshi = zp2sos(zhi,phi,khi);
freqz(soshi)
```



```
outsignal = sosfilt(soshi, y);

figure;
plot(t, y, 'b');
hold on;
plot(t, outsignal, 'r');
legend('Sygnał Oryginalny', 'Sygnał Przetworzony');
grid on;
```



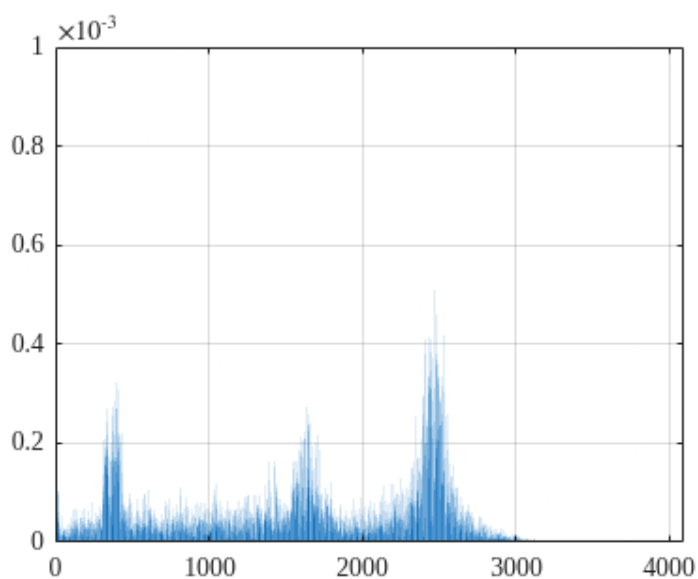
% Wizualizacja Sygnału Wynikowego w Dziedzinie Częstotliwości (FFT)

```
N_fft = length(y);
xfft = abs(fft(outsignal));

xfft_norm = xfft / N_fft;
f_Hz = (0:N_fft/2) * Fs / N_fft;
figure;
bar(f_Hz, xfft_norm(1:N_fft/2 + 1));
```

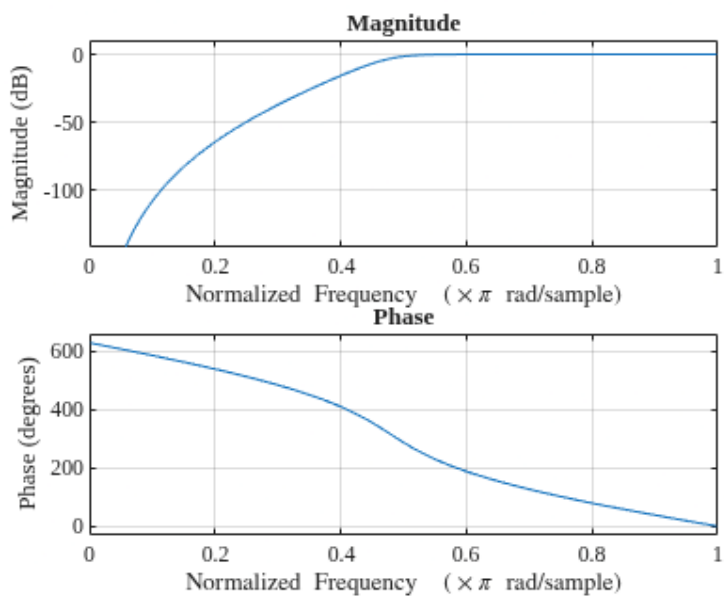
Warning: Integer operands are required for colon operator when used as index.

```
grid on;
axis([0, Fs/2, 0, 0.001]);
```

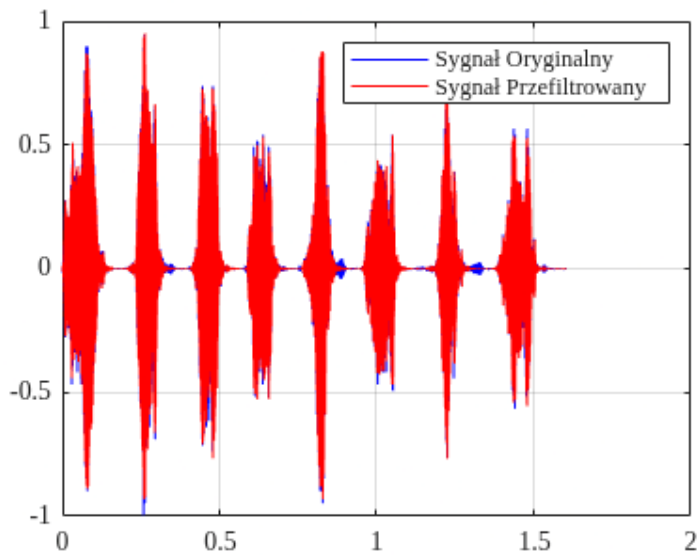


15)

```
[zhi,phi,khi] = butter(7,0.48,'high');  
soshi = zp2sos(zhi,phi,khi);  
freqz(soshi)
```



```
outsignal = sosfilt(soshi, y);  
  
figure;  
plot(t, y, 'b');  
hold on;  
plot(t, outsignal, 'r');  
legend('Sygnał Oryginalny', 'Sygnał Przetworzony');  
grid on;
```



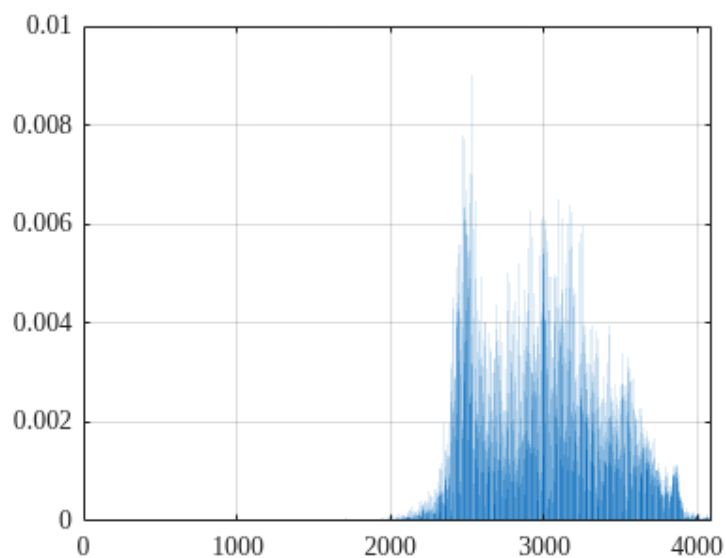
% Wizualizacja Sygnału Wynikowego w Dziedzinie Częstotliwości (FFT)

```
N_fft = length(y);
xfft = abs(fft(outsignal));

xfft_norm = xfft / N_fft;
f_Hz = (0:N_fft/2) * Fs / N_fft;
figure;
bar(f_Hz, xfft_norm(1:N_fft/2 + 1));
```

Warning: Integer operands are required for colon operator when used as index.

```
grid on;
axis([0, Fs/2, 0, 0.01]);
```



WNIOSKI

W przypadku filtru Yule-Walker ($N=10$), analiza z-płaszczyzny wykazała, że wszystkie bieguny leżą wewnątrz okręgu jednostkowego, potwierdzając stabilność systemu.

Specyfikacja rygorystycznego, wąskiego pasma przejściowego w filtrze Eliptycznym wymusiła znaczne podniesienie rzędu filtru N . Potwierdza to inżynierski kompromis: zwiększona stromość lub tłumienie zawsze prowadzi do wyższej złożoności obliczeniowej.

Przy filtrze IIR o wysokim rzędzie $N=7$ lub $N=10$ zaleca się użycie struktury kaskadowej SOS (`zp2sos`, `sosfilt`) zamiast bezpośredniej formy (`filter(b,a)`), aby złagodzić krytyczne problemy numeryczne, takie jak błędy zaokrąglania i ryzyko niestabilności wywołane skończoną precyzją współczynników.

Proces projektowania (`yulewalk`) skutecznie przetwarza abstrakcyjną specyfikację (f , m) na implementacyjne współczynniki (b , a), ale z widocznym błędem aproksymacji, charakteryzującym się pofalowanymi minimami w paśmie zaporowym, widzione są "górki", z dokładnymi zerami jedynie tam, gdzie zostało to określone w charakterystyce f, m .