

Przetwarzanie dźwięków i obrazów

Filtry cyfrowe

część 2:

**FILTRY IIR**

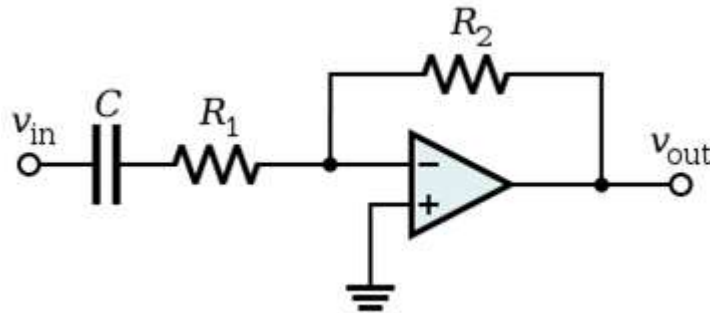
Opracowanie: Grzegorz Szwoch

Politechnika Gdańska, Katedra Systemów Multimedialnych

[greg@multimed.org](mailto:greg@multimed.org)

## Filtry w technice analogowej:

- działają tak jak filtry FIR: filtrują wybrane zakresy częstotliwości,
- są budowane z rezystorów i kondensatorów (RC), cewek (RLC), czasami z dodatkiem wzmacniaczy operacyjnych (filtry aktywne).

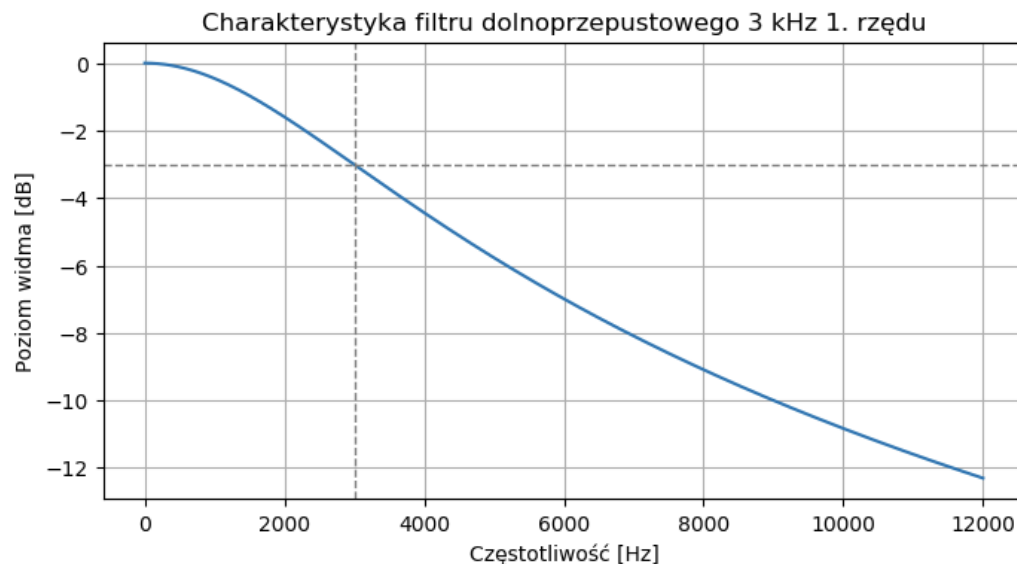
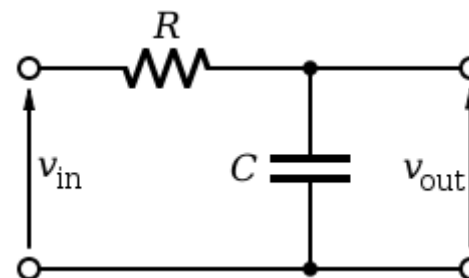


Transmitancja filtrów analogowych jest opisana w dziedzinie ciągłej zmiennej zespolonej  $s$ .

Przykład: dolnoprzepustowy filtr **pierwszego rzędu**

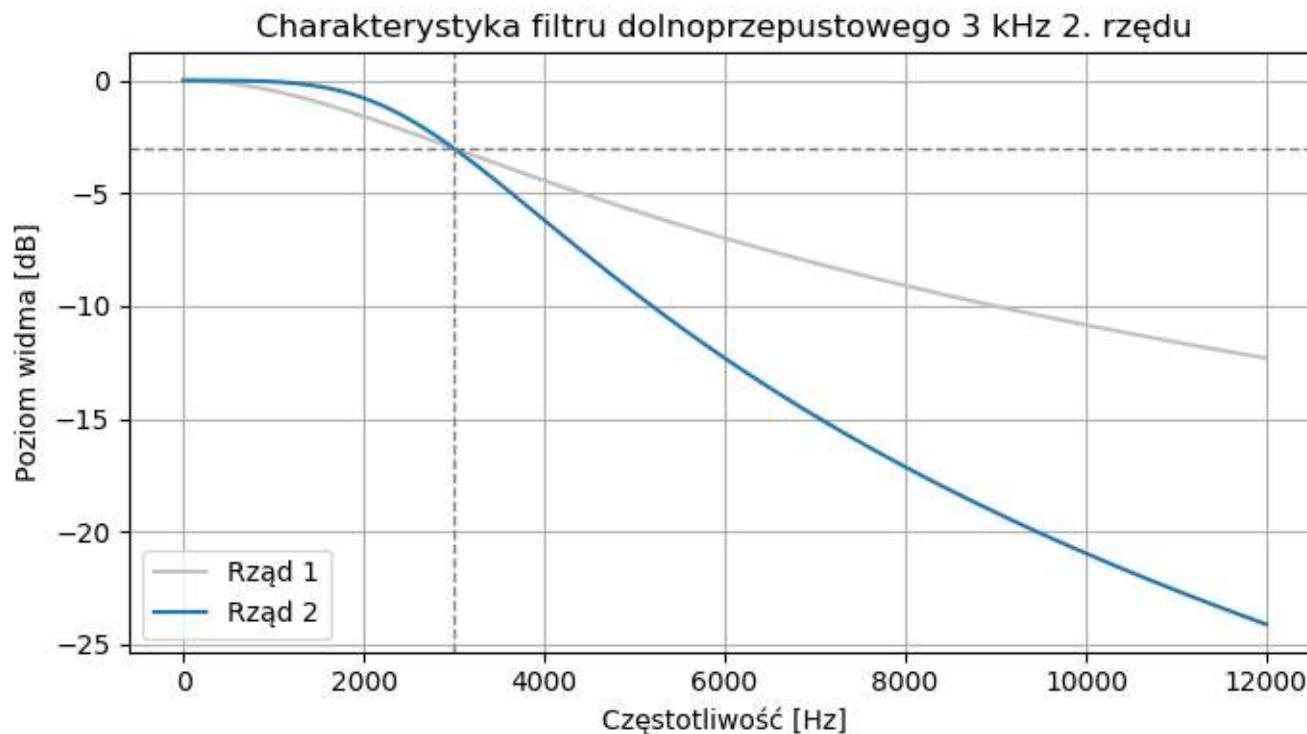
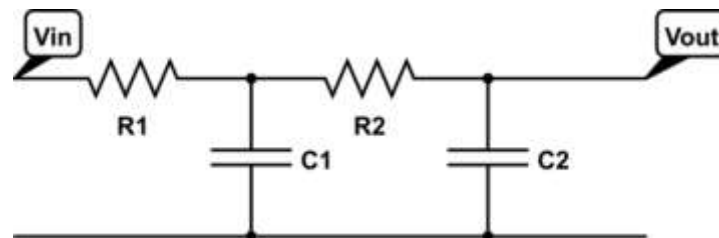
$$H(s) = \frac{\omega_c}{s + \omega_c}$$

$$f_c = \frac{\omega_c}{2\pi} = \frac{1}{2\pi RC}$$



Filtr drugiego rzędu – szeregowe połączenie dwóch filtrów pierwszego rzędu.

$$H(s) = \frac{\omega_c^2}{s^2 + a\omega_c s + \omega_c^2}$$



No dobrze, ale to miał być wykład o filtrach cyfrowych?

Transmitancję w dziedzinie „analogowej” zmiennej  $s$  można przekształcić na transmitancję w dziedzinie „cyfrowej” zmiennej  $z$  za pomocą **przekształcenia dwuliniowego** (*bilinear transform*):

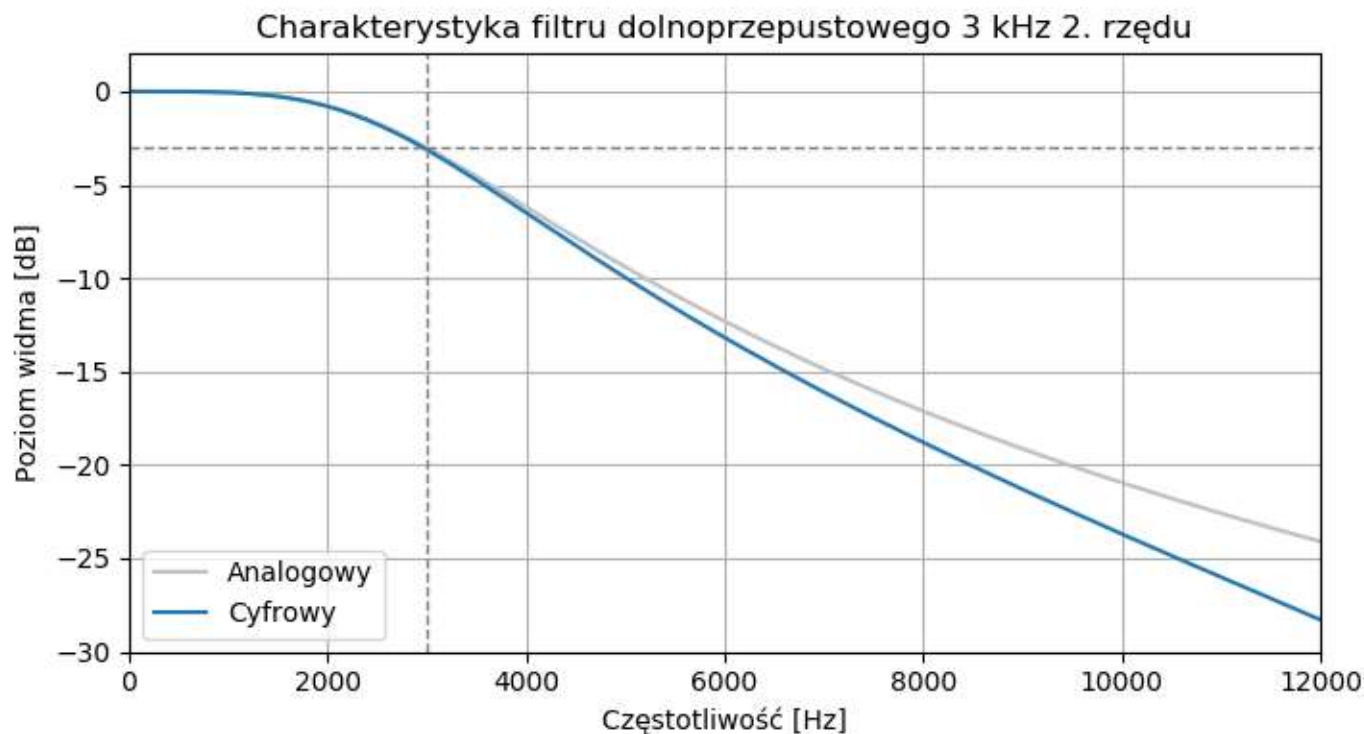
$$s \rightarrow \frac{2}{T} \frac{z-1}{z+1} = \frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}}$$

Transmitancja filtru drugiego rzędu przyjmuje postać:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Przykład dla filtru DP 2. rzędu, cz. graniczna 3 kHz:

$$H(z) = \frac{0,02929 + 0,05858z^{-1} + 0,0929z^{-2}}{1 - 1,46091z^{-1} + 0,57807z^{-2}}$$



## Metoda projektowania filtra cyfrowego:

- projektujemy filtr analogowy o zadanych parametrach,
- wykonujemy przekształcenie dwuliniowe,
- otrzymujemy transmitancję filtra cyfrowego.
- W odróżnieniu od filtrów FIR, te filtry mają współczynniki w liczniku i w mianowniku  $H(z)$ .
- Filtry tego typu nazywamy filtrami cyfrowymi o nieskończonej odpowiedzi impulsowej (IIR, *infinite impulse filter*).

## Transmitancja filtru IIR:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}}$$

- Wartość  $N$  oznacza **rzęd filtru** (*filter order*).
- W filtrach IIR nie posługujemy się już pojęciem długości filtru.
- Filtr rzędu  $N$  ma  $N+1$  współczynników  $b$  w liczniku i  $N$  współczynników  $a$  w mianowniku.
- Unormowana transmitancja ma zawsze  $a_0 = 1$ .
- W algorytmach czasami podaje się  $a_0$ , czasami się go pomija.



Transmitancja filtru IIR 2. rzędu:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Rozpiszmy ją w dziedzinie czasu:

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + \\ - a_1 y[n-1] - a_2 y[n-2]$$

Jest to **równanie różnicowe** filtru IIR 2. rzędu.

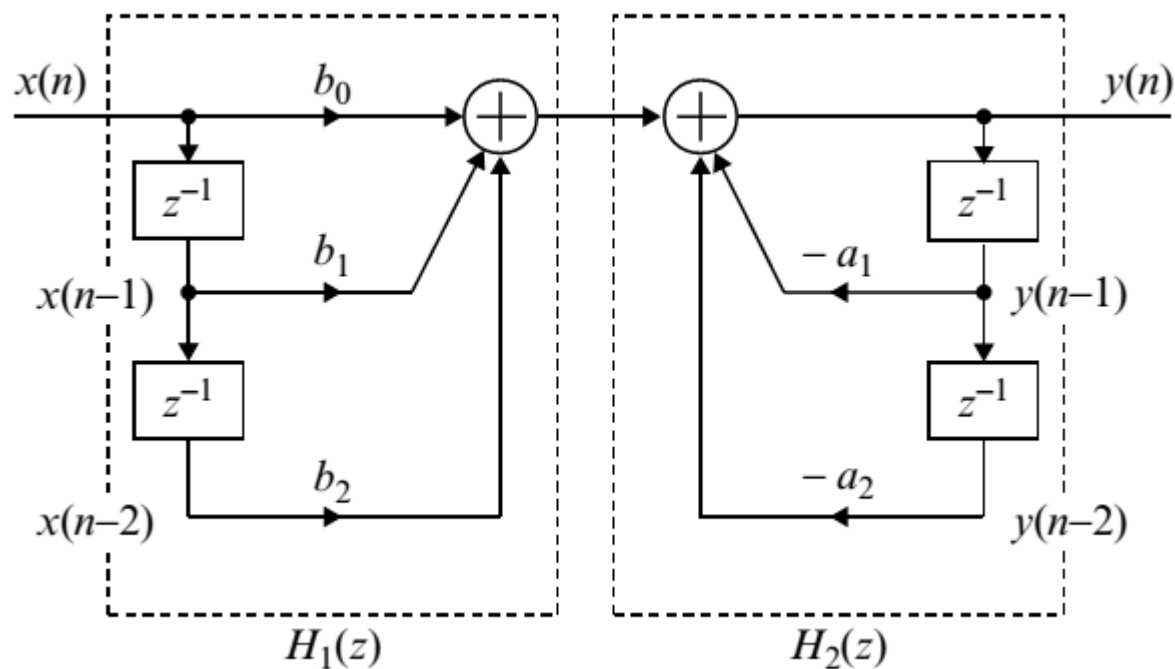
$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + \\ - a_1y[n-1] - a_2y[n-2]$$

- A zatem wynik filtracji zależy nie tylko od wartości próbek sygnału wejściowego, ale również od poprzednich wyników filtracji!
- Filtry IIR są **rekursywne**. Posiadają pętlę sprzężenia zwrotnego między wyjściem a wejściem.
- Wynik filtracji bieżącej próbki ma wpływ na wyniki dla kolejnych próbek sygnału.
- Jest to efekt mianownika w transmitancji filtru. Filtry FIR mają 1 w mianowniku – nie są rekursywne.

- Aby dokonać filtracji, filtr musi zapamiętać poprzednie próbki wejściowe i poprzednie wyniki filtracji.
- Jest to **pamięć** albo **stan** filtru (*filter memory/state*).
- Stan dla filtru rzędu  $N$  musi zawierać:
  - $N$  poprzednich próbek wejściowych,
  - $N$  poprzednich wyników filtracji.
- Stan filtru musi być aktualizowany po każdej przetworzonej próbce.

# Schemat filtracji IIR dla filtru rzędu 2

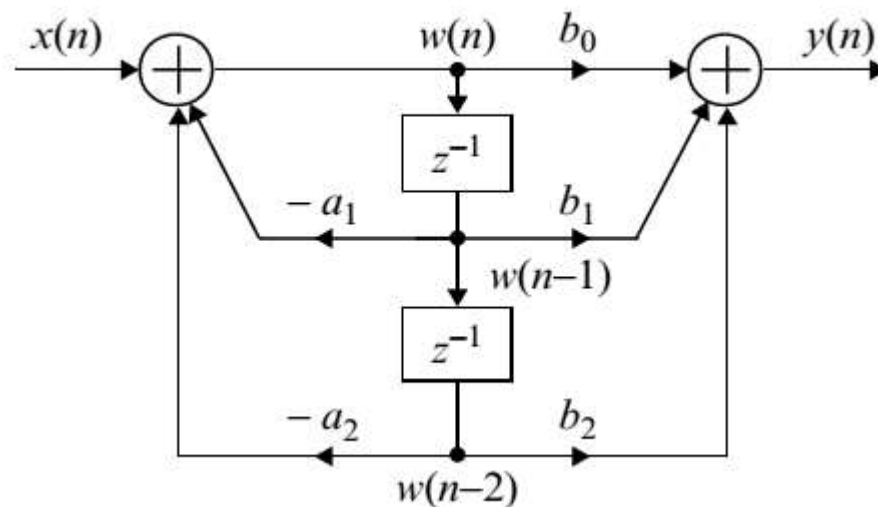
- forma bezpośrednia pierwsza (*direct form I, DF1*)



## Schemat filtracji IIR dla filtru rzędu 2

- forma bezpośrednia druga (*direct form II, DF2*)

- wymaga mniej operacji (+, \*),
- bardziej podatna na błędy numeryczne,
- częściej stosowana w praktyce.



$$w[n] = x[n] - a_1 w[n-1] - a_2 w[n-2]$$

$$y[n] = b_0 w[n] + b_1 w[n-1] + b_2 w[n-2]$$

Transmitancję filtru IIR:

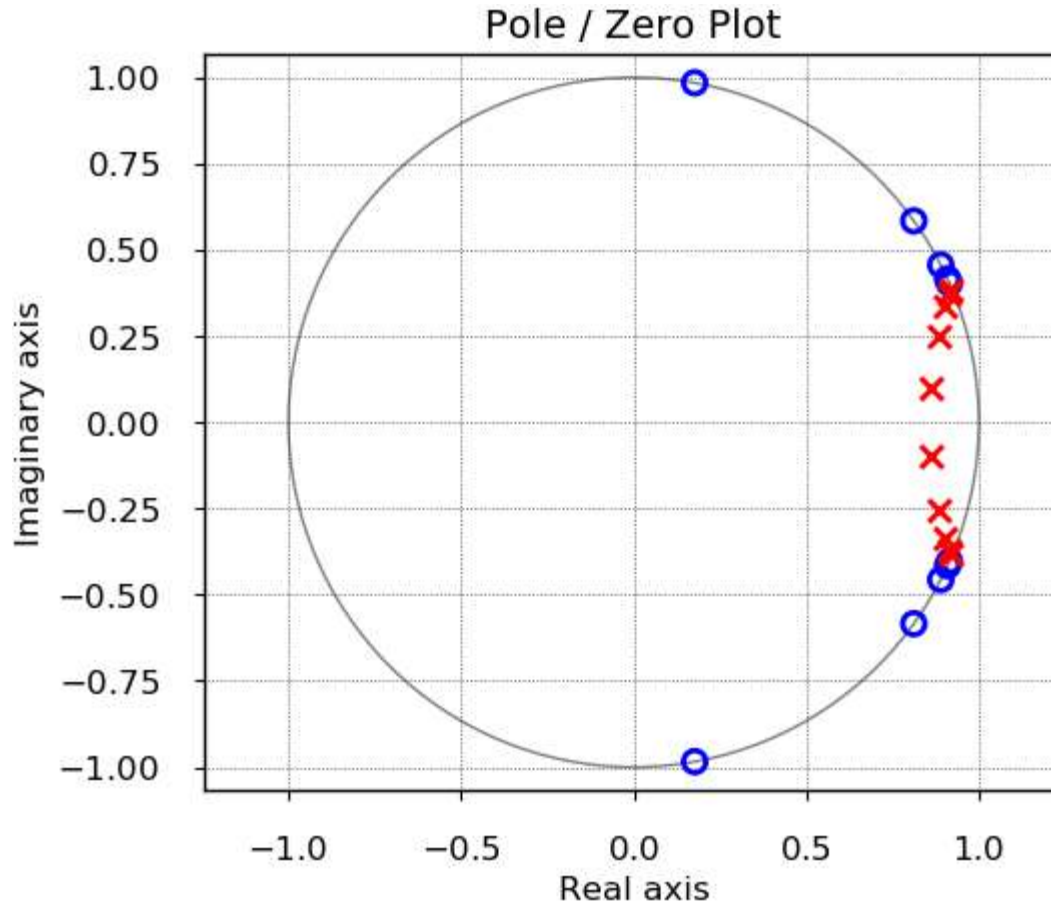
$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}}$$

można zapisać w równoważnej formie ZPK:

$$H(z) = k \frac{(z - r_1)(z - r_2) \dots (z - r_N)}{(z - p_1)(z - p_2) \dots (z - p_N)}$$

- $r$  – zera transmitancji (miejsca zerowe licznika),
- $p$  – bieguny (*pole*) (miejsca zerowe mianownika),
- $k$  – stałe wzmocnienie (*gain*)

Zera (o) i bieguny (x) na płaszczyźnie zespolonej:



Zera oraz bieguny tworzą pary zespolone sprzężone.

- Ze względu na pętlę sprzężenia zwrotnego, odpowiedź filtru IIR na pobudzenie impulsem **może być** nieskończona (stąd jego nazwa).
- To wcale nie oznacza, że **powinna być** nieskończona!
- Aby filtr mógł działać w praktyce, odpowiedź na skończone pobudzenie **musi** wygasnąć do zera w skończonym czasie.
- Taki filtr nazywamy **stabilnym**.
- Jeżeli ten warunek nie jest spełniony, filtr jest **niestabilny** i nie nadaje się do użytku.



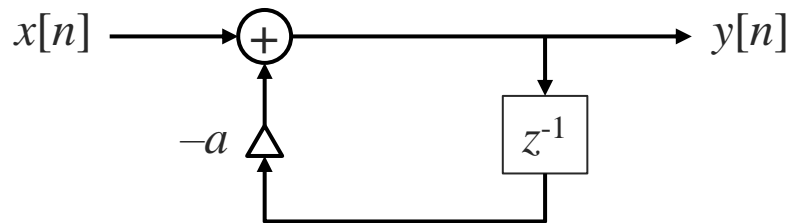
Co się dzieje, jeżeli układ nie jest stabilny?

Przykład: katastrofa mostu Tacoma Bridge (1940)



Stabilność filtra najprościej jest wytłumaczyć na przykładzie filtra 1. rzędu (jednobiegunowego).

$$y[n] = x[n] + ay[n - 1]$$



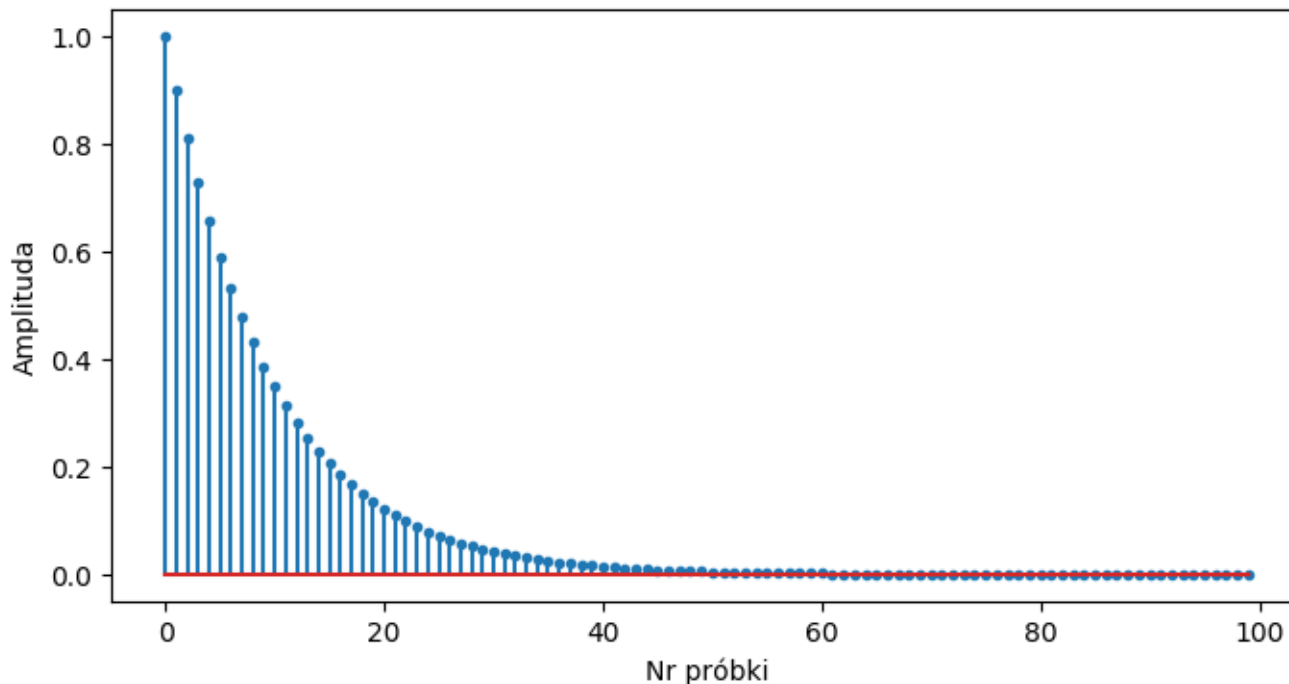
Pobudzamy układ impulsem jednostkowym:  
[1, 0, 0, 0, 0, ...]

Co się stanie dla różnych wartości  $a$ ?

## Przypadek 1: $|a| < 1$

$a = -0,9$ :  $y = [1.0000, 0.9000, 0.8100, 0.7290, 0.6561, 0.5905, 0.5314, 0.4783, 0.4305, 0.3874, \dots]$

Amplituda impulsu zmniejsza się po każdym przejściu przez pętlę i wygasa do zera – filtr jest **stabilny**.

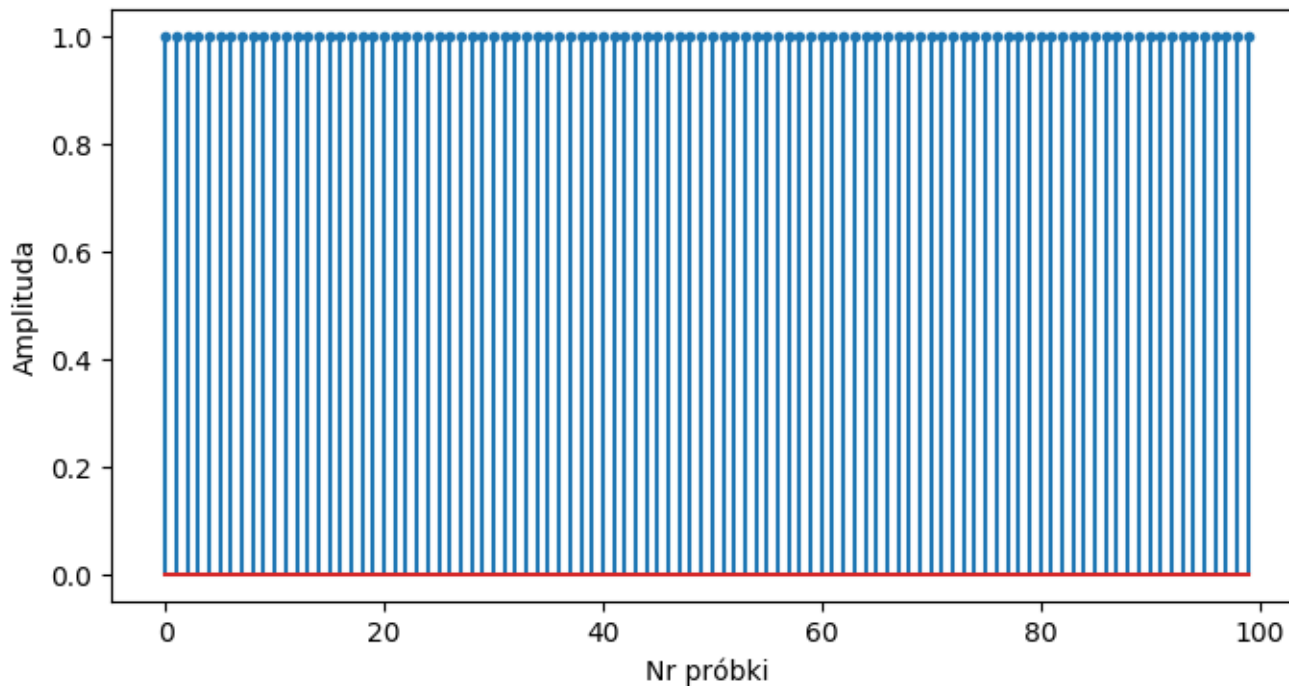


## Przypadek 2: $|a| = 1$

$a = -1$ :  $y = [1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, \dots]$

Impuls będzie krążył w nieskończoność w pętli,  
z niezmienną amplitudą.

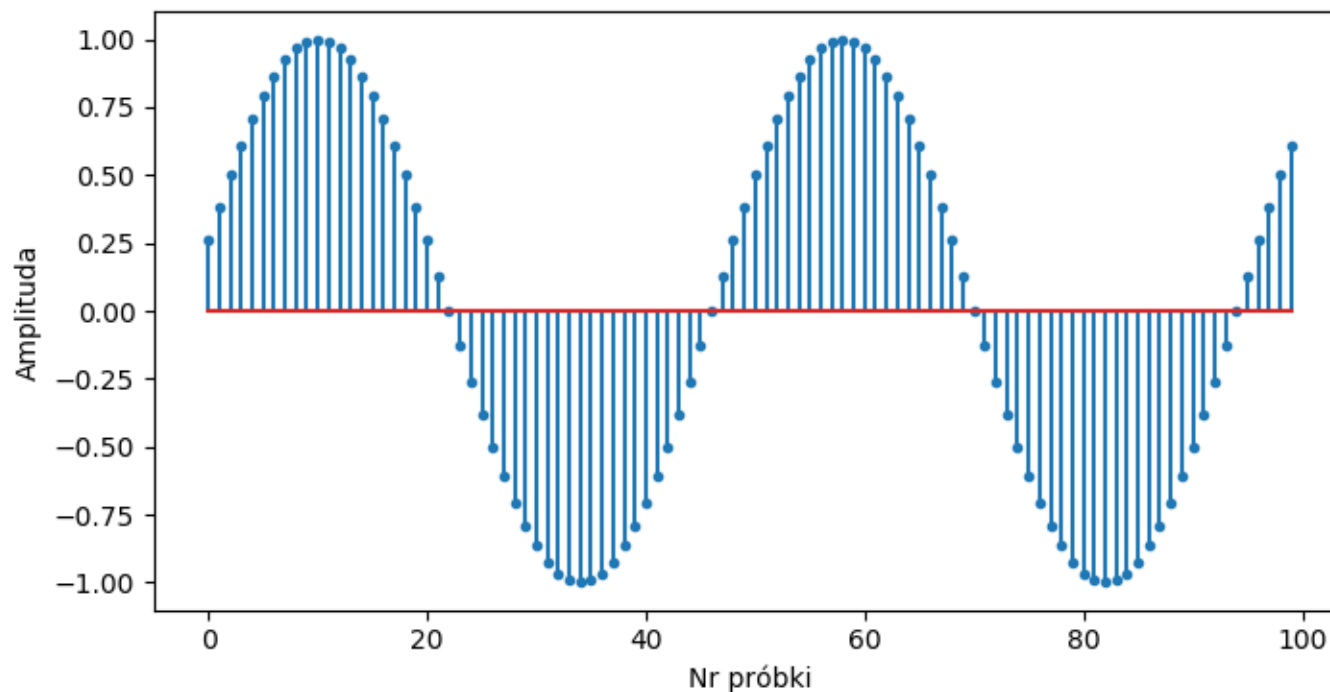
Filtr jest **na granicy stabilności**.



Układ na granicy stabilności może mieć pewne zastosowania, choć nie jako filtr.

Może np. generować sinusa.

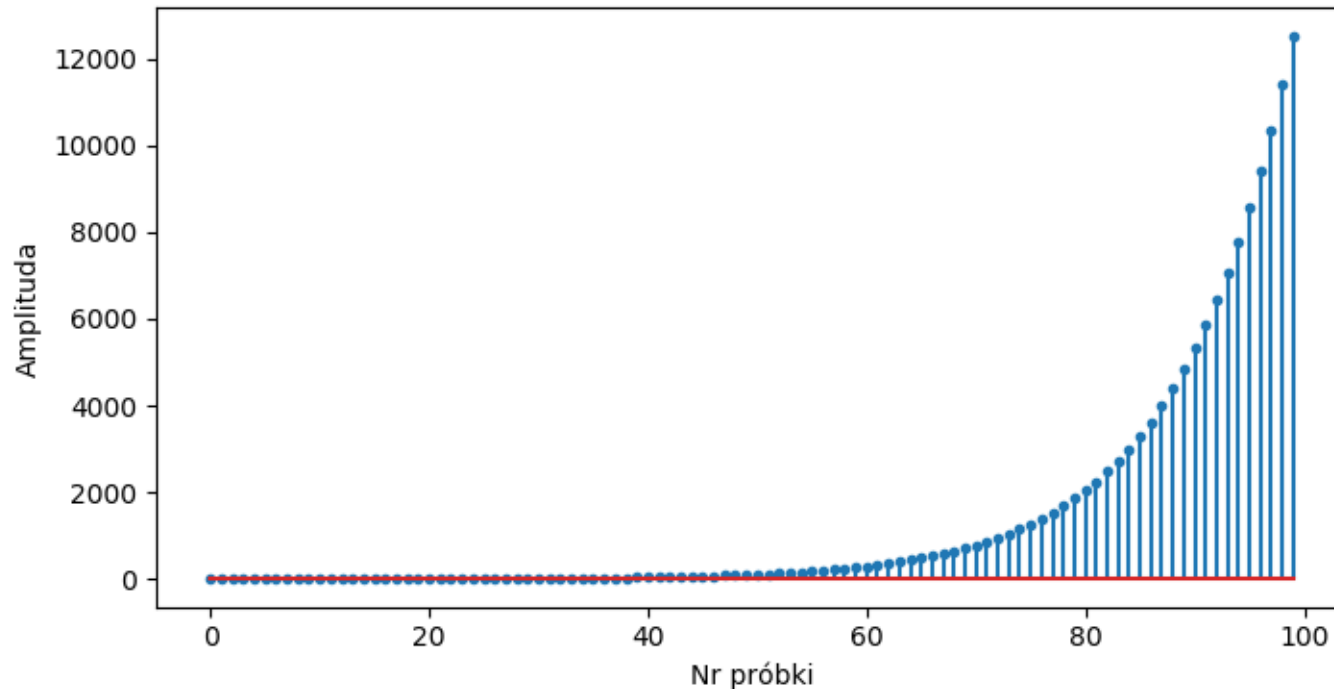
Odpowiedź impulsowa takiego układu:



## Przypadek 3: $|a| > 1$

$a = -1,1$ :  $y = [1.0000, 1.1000, 1.2100, 1.3310, 1.4641, 1.6105, 1.7716, 1.9487, 2.1436, 2.3579, \dots]$

Po każdym przejściu przez pętlę, amplituda impulsu zwiększa się i narasta do  $\infty$ ! Filtr jest **niestabilny**.



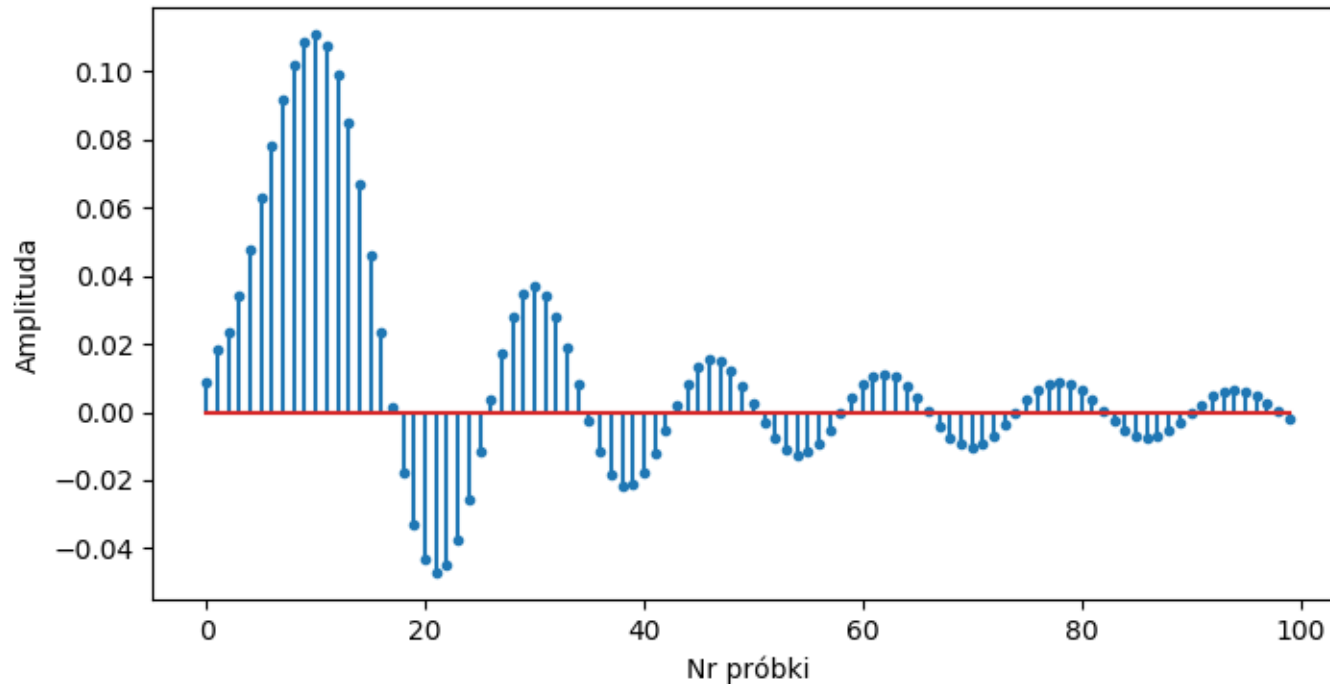
- Dla filtru pierwszego rzędu, warunek stabilności to  $|a_1| < 1$ .
- Analogiczny warunek można sformułować dla filtrów wyższych rzędów. **Filtr IIR jest stabilny, jeżeli dla wszystkich biegunów:**

$$|p_i| < 1$$

- Jeżeli choć jeden biegun nie spełnia tego warunku, wzmocnienie w pętli sprzężenia zwrotnego staje się dodatnie i filtr jest niestabilny.
- Projektowane filtry IIR muszą być stabilne.

Przykład odpowiedzi impulsowej stabilnego filtra IIR  
(filtr dolnoprzepustowy 3 kHz, 4. rzędu).

Odpowiedź impulsowa wygasa do zera.





## Sprawdzanie stabilności filtru IIR:

- Transmitancję zapisaną jako współczynniki ( $b, a$ ) musimy zamienić na formę ZPK (funkcja `tf2zpk` w Matlabie i w SciPy).
- Obliczamy wartość bezwzględną wszystkich biegunów.
- Obliczamy maksimum tych wartości.
- Jeżeli jest mniejsza od 1, filtr jest stabilny.

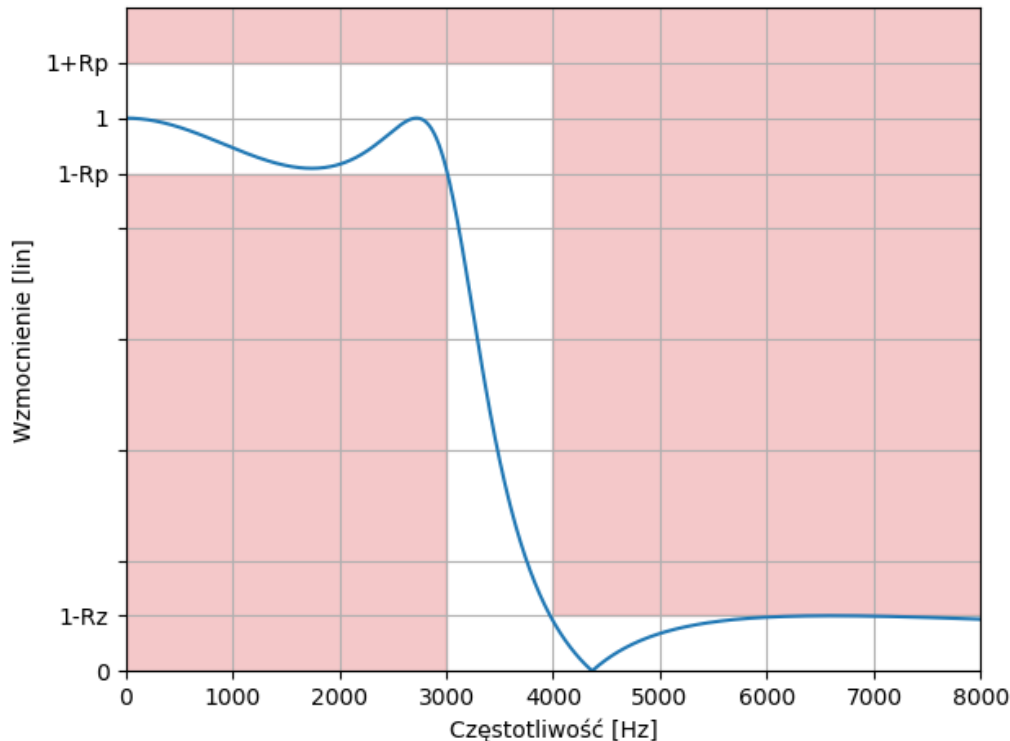
Filtry FIR mają  $p_i = 0$ , dlatego są zawsze stabilne.

W filtrach IIR, bieguny zależą od współczynników  $a$ .

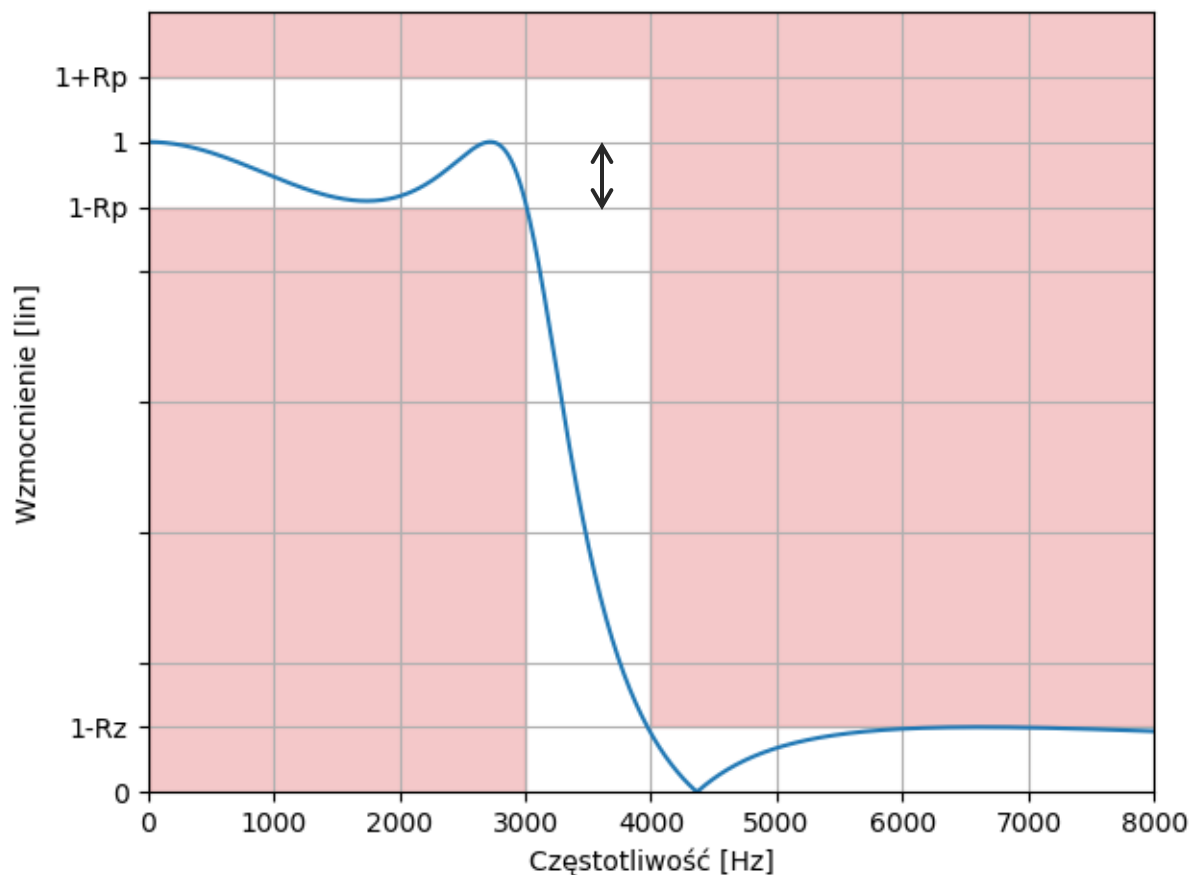
## Projektowanie filtru IIR

Projekt wymaga zadania parametrów, opisujących dopuszczalne odchyłki względem idealnego filtru.

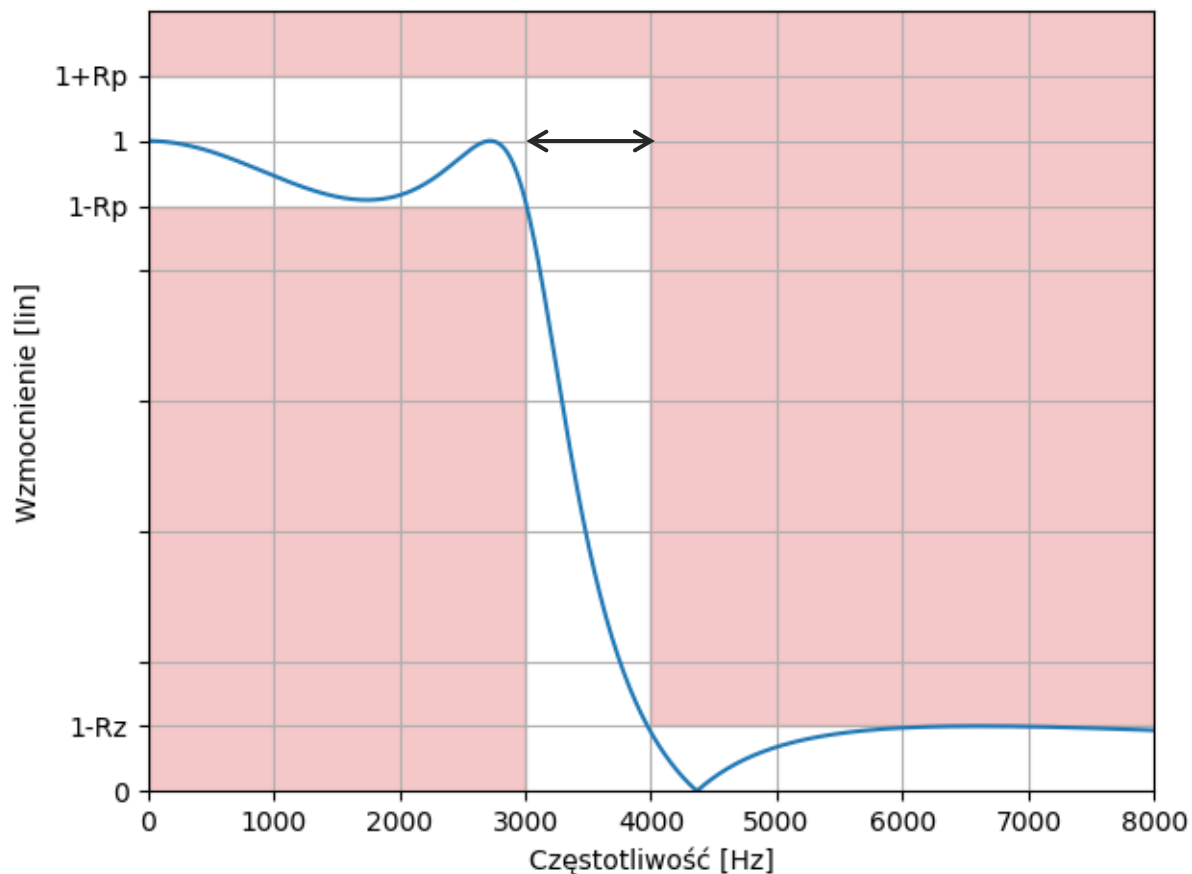
Zaprojektowana charakterystyka nie może „wchodzić” na czerwone pola:



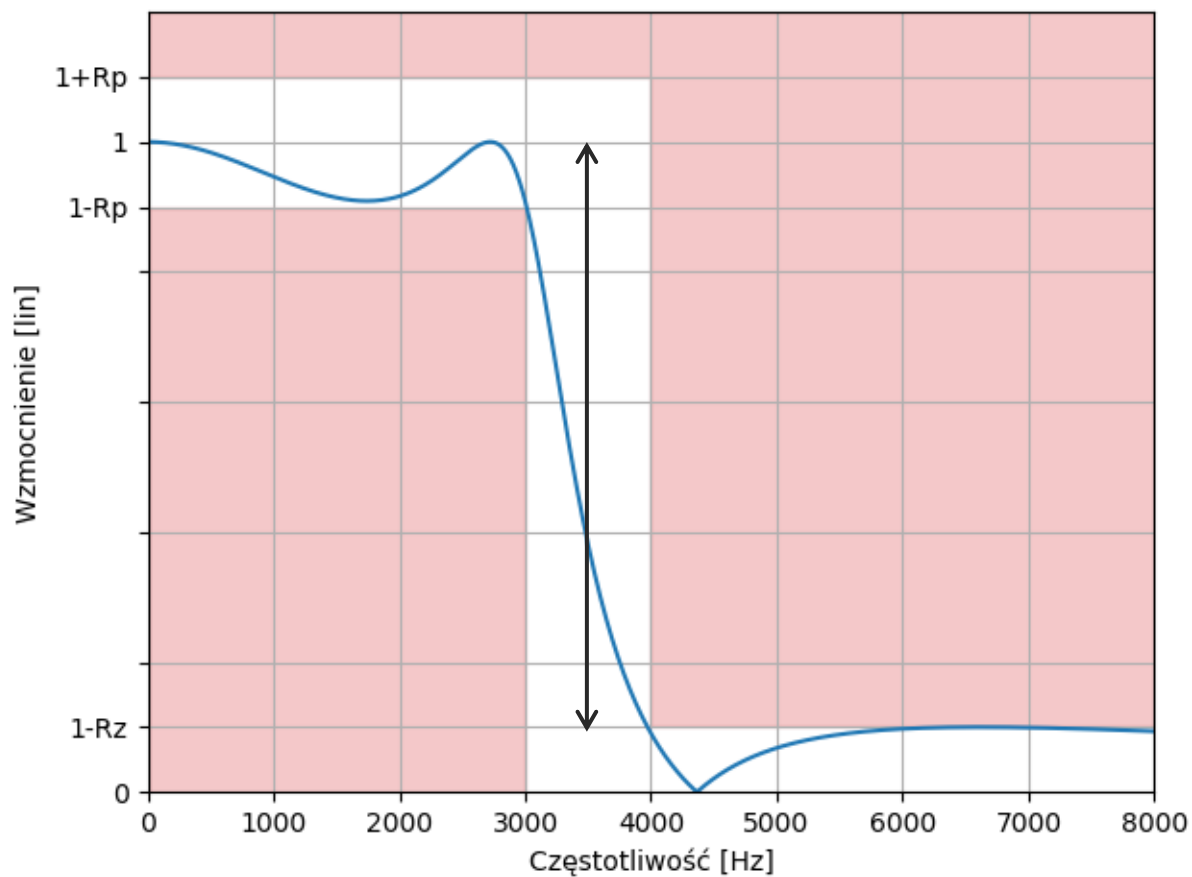
1.  $R_p$  – maksymalne zafalowanie w paśmie przepustowym (maksymalna utrata wzmacnienia w tym paśmie). Najczęściej podawane w decybelach.



2.  $(f_p, f_z)$  – szerokość pasma przejściowego, czyli częstotliwości graniczne pasma przepustowego i zaporowego. Alternatywnie:  $(f_p, N)$  – rząd filtru.

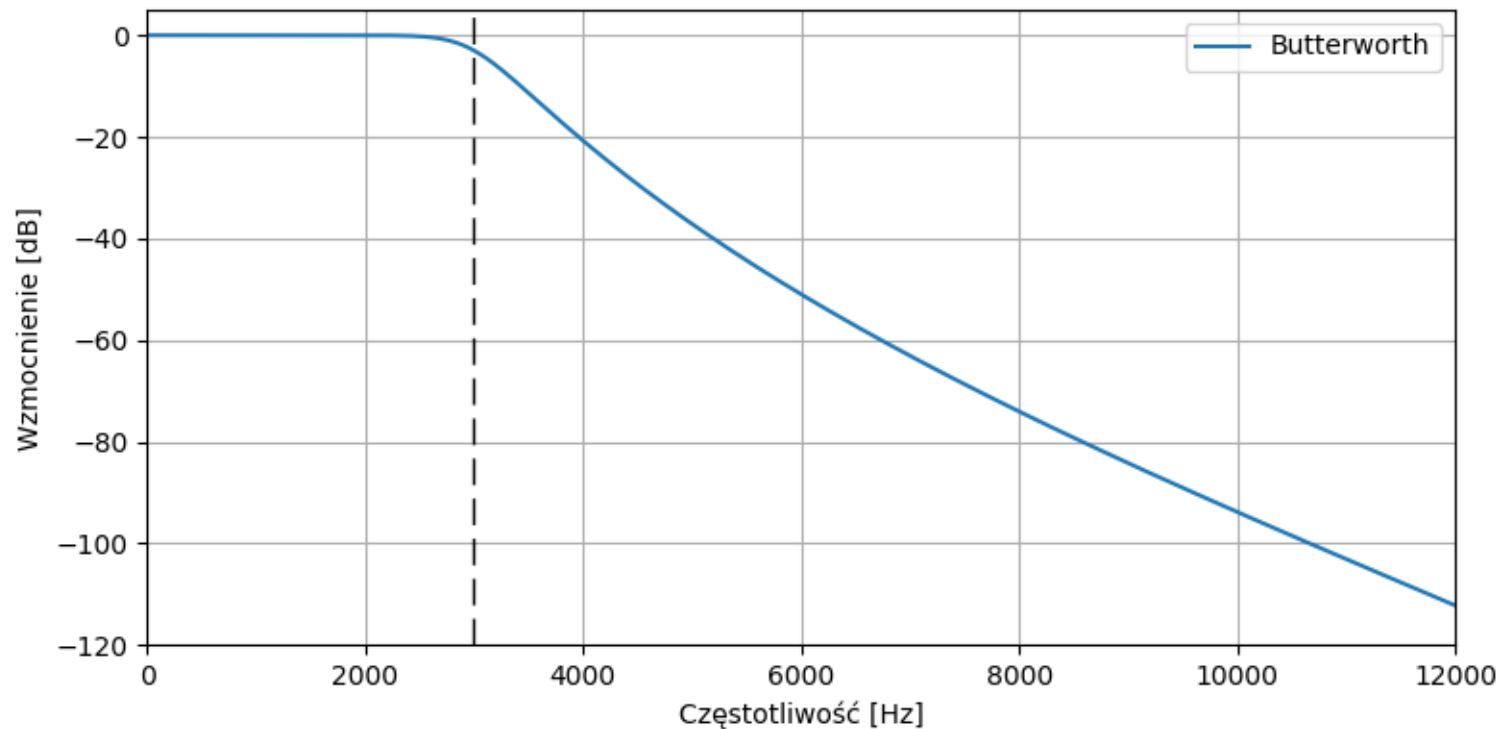


3.  $R_z$  – minimalne tłumienie w paśmie zaporowym, względem pasma przepustowego. Najczęściej w dB.



# Filtr Buttwerwortha

- parametry:  $(f_p, f_z)$  lub  $(f_p, N)$
- brak zafalowań w p. przepustowym i zaporowym
- szerokie pasmo przejściowe (słabe tłumienie)

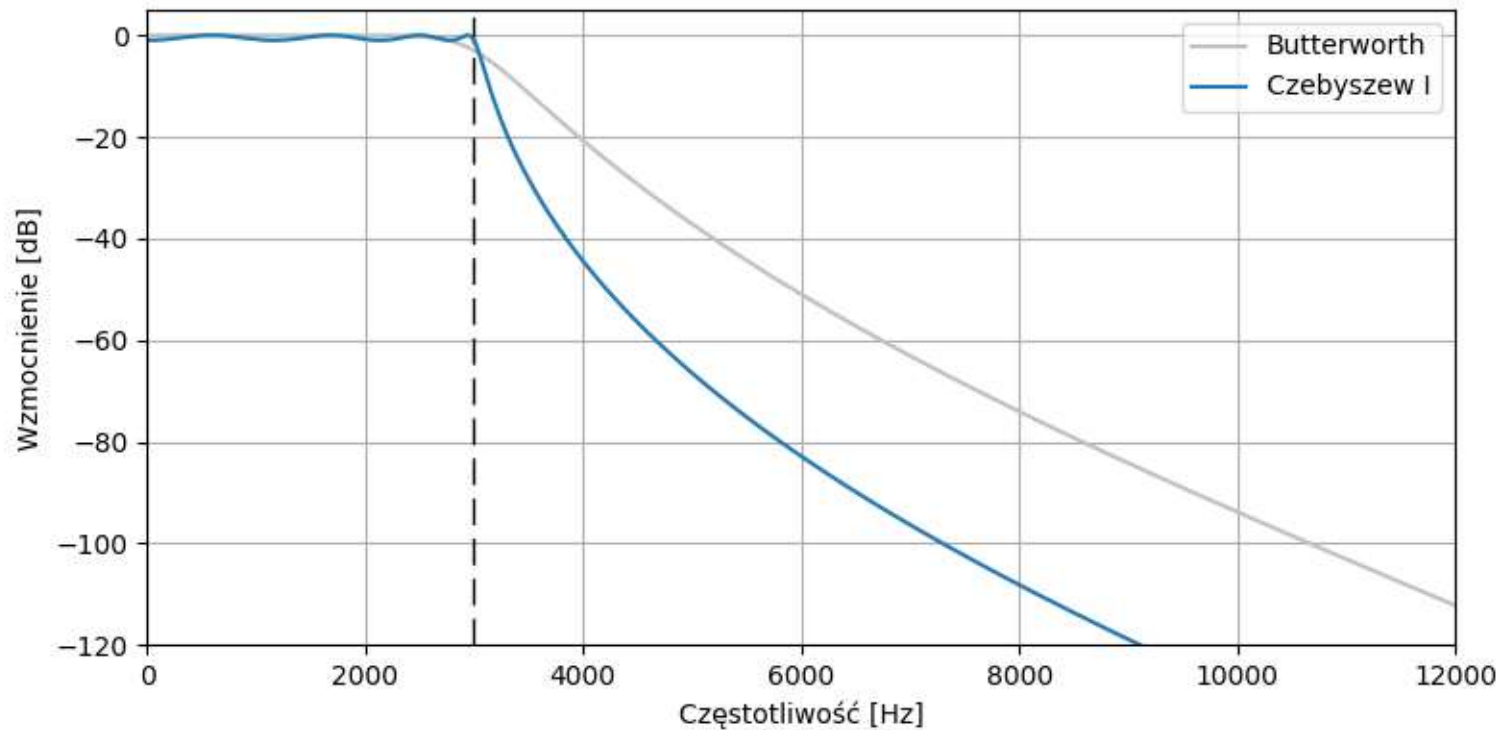


## Cechy filtru Butterwortha:

- Wzmocnienie na częstotliwości granicznej jest o 3 dB mniejsze niż w paśmie przepustowym (0,707 razy liniowo).
- Szybkość opadania charakterystyki w paśmie przejściowym wynosi około  $N \cdot 6$  dB/oktawę, gdzie  $N$  jest rzędem filtru (oktawa = podwojenie częstotliwości, np. 4000 Hz  $\rightarrow$  8000 Hz).
- Pasma zaporowe w zasadzie nie występuje.
- Słabe tłumienie wymaga stosowania dużych  $N$ .

## Filtr Czebyszewa typu pierwszego

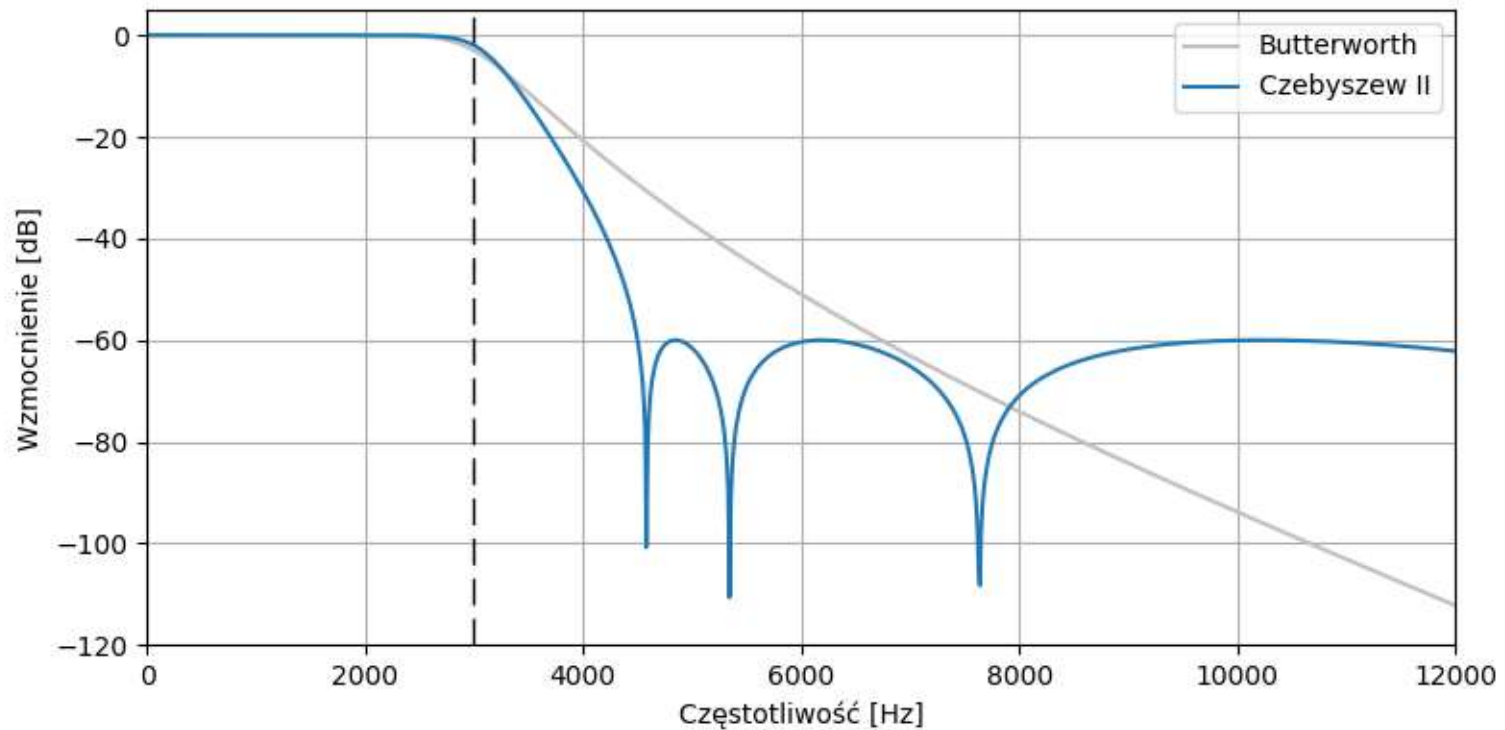
- parametry:  $(R_p, f_p, f_z)$  lub  $(R_p, f_p, N)$
- lepsze tłumienie niż f. Butterwortha,
- za to zafalowania w p. przepustowym





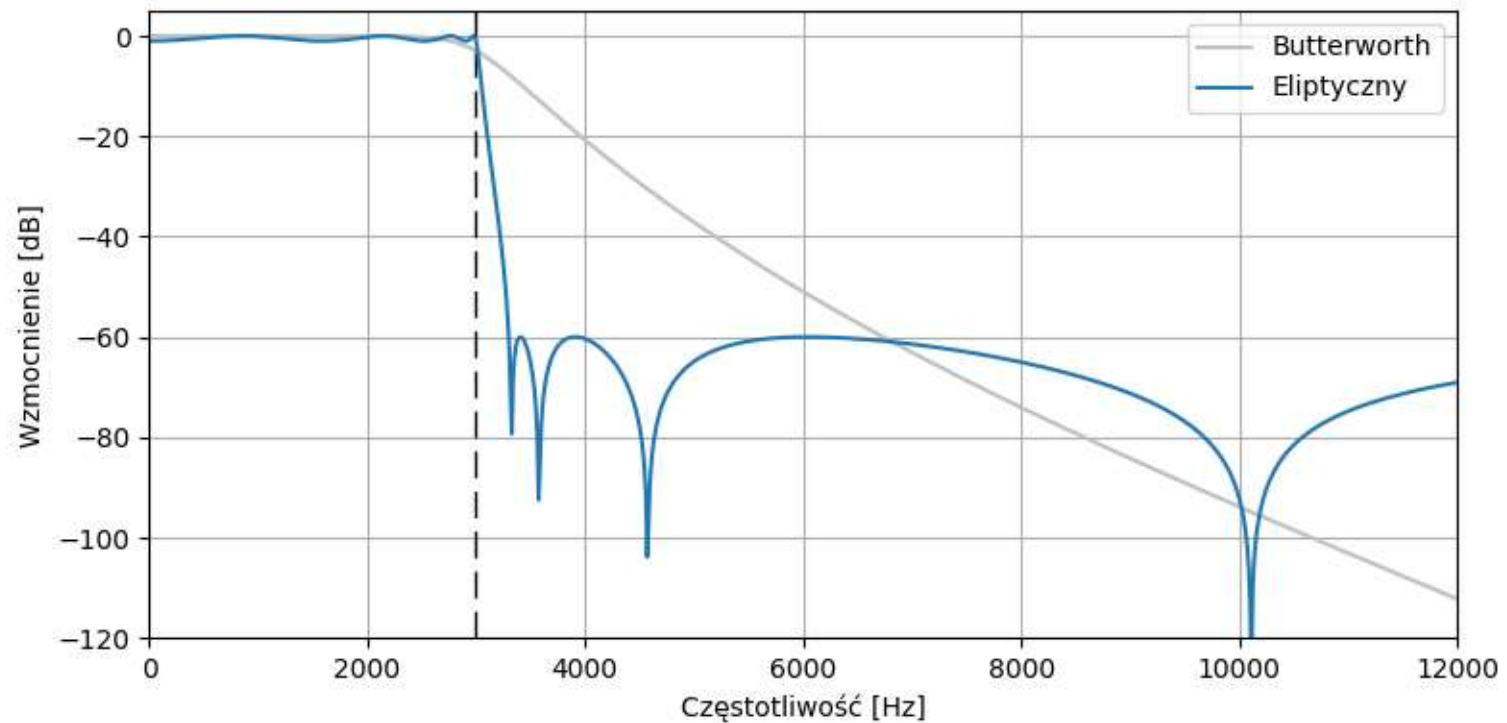
## Filtr Czebyszewa typu drugiego

- parametry:  $(f_p, f_z, Rz)$  lub  $(f_p, N, Rz)$
- lepsze tłumienie niż f. Butterwortha,
- za to zafalowania w p. zaporowym

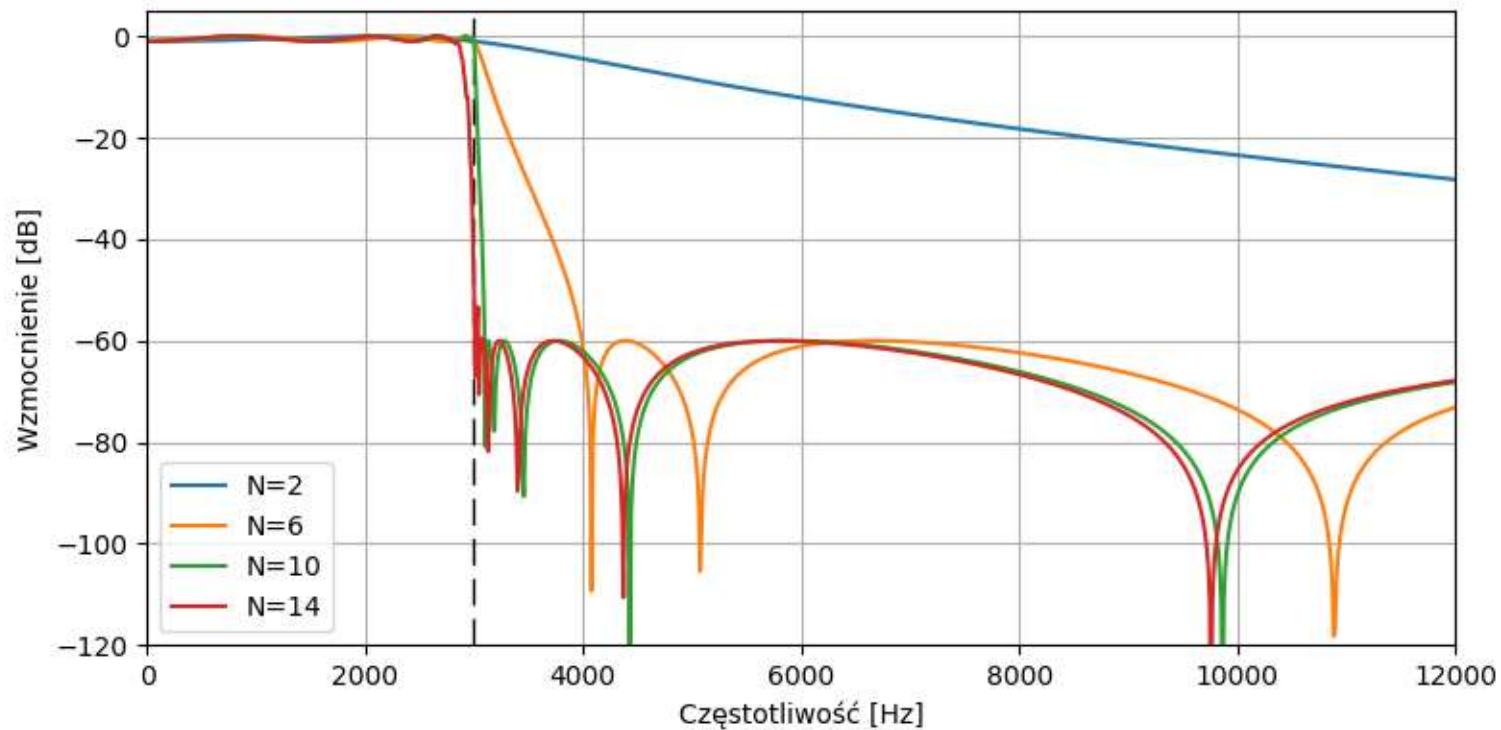


## Filtr eliptyczny (Cauera)

- parametry:  $(R_p, f_p, f_z, R_z)$  lub  $(R_p, f_p, N, R_z)$
- największe pasmo przejściowe ze wszystkich typów,
- za to zafalowania w obu pasmach!

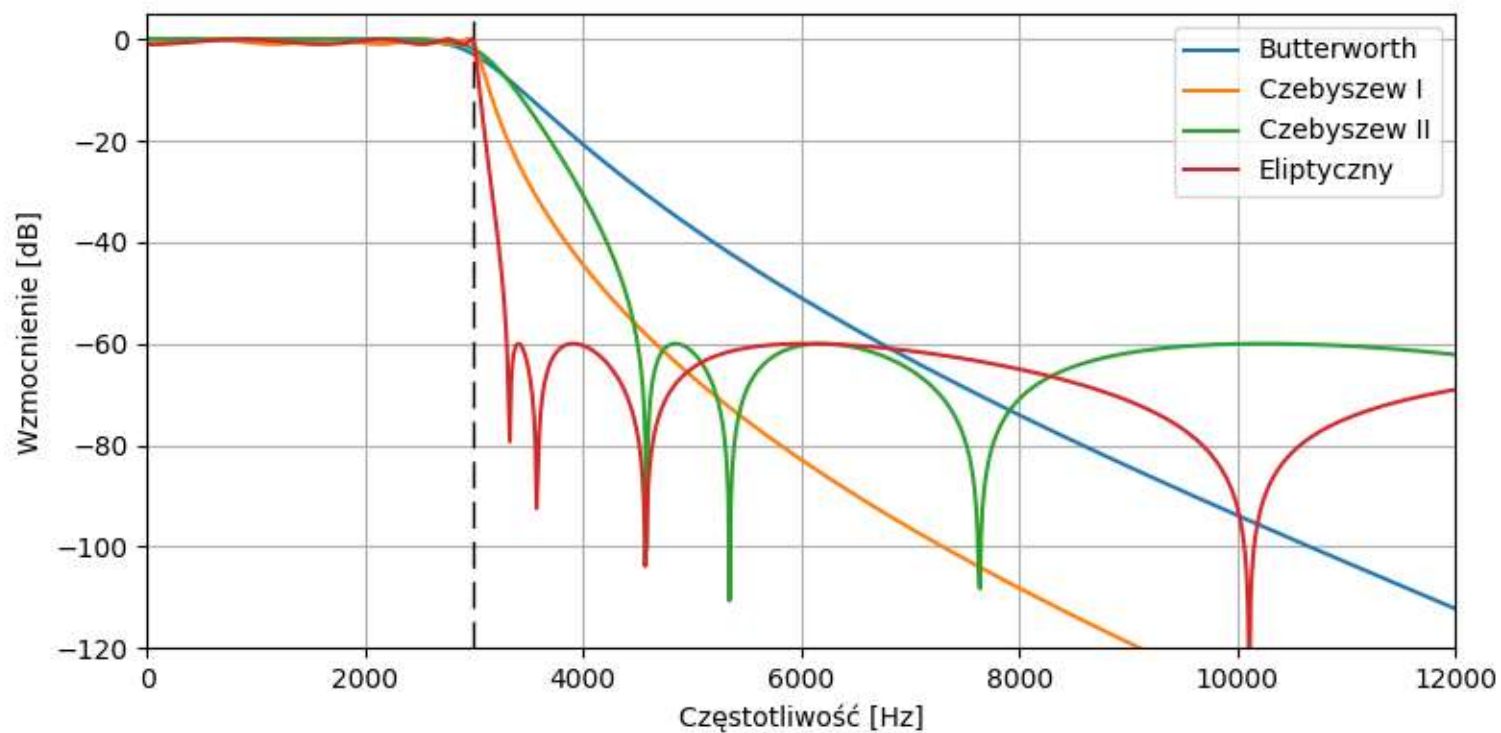


Wpływ rzędu filtru eliptycznego na charakterystykę:  
większy rząd  $N$  to węższe pasmo przejściowe  
(dotyczy wszystkich typów).



## Porównanie wszystkich typów

(filtr DP 3 kHz,  $N = 8$ ,  $R_p = 1$  dB,  $R_z = 60$  dB)



No więc który typ wybrać?

- **Eliptyczny**, jeżeli zależy nam na skutecznym tłumieniu, a dopuszczamy zafalowania w obu p.
- **Czebyszewa**, jeżeli zależy nam na braku zafalowań w jednym z pasm (zwłaszcza w p. przepustowym), a godzimy się na mniej skuteczne tłumienie.
- **Butterwortha**, jeżeli absolutnie nie zgadamy się na zafalowania, a skuteczność tłumienia nie jest tak bardzo ważna (używamy filtrów wyższego rzędu).

Filtr eliptyczny jest najczęściej stosowany.

- Projektuje się **analogowy prototyp** wybranego typu filtru, spełniający założenia projektowe.
- Jest to zawsze filtr dolnoprzepustowy, o znormalizowanej pulsacji granicznej  $\omega = 1$ .
- Następnie przekształca się prototyp do żądanej charakterystyki i częstotliwości granicznej.
- Zamiana w filtr dolno- lub górnoprzepustowy:

DP

$$s \rightarrow \frac{s}{\omega_c}$$

GP

$$s \rightarrow \frac{\omega_c}{s}$$

- Zamiana w filtr pasmowo-przepustowy / -zaporowy:

PP

$$s \rightarrow \frac{s^2 + \omega_d \omega_g}{(\omega_g - \omega_d)s}$$

PZ

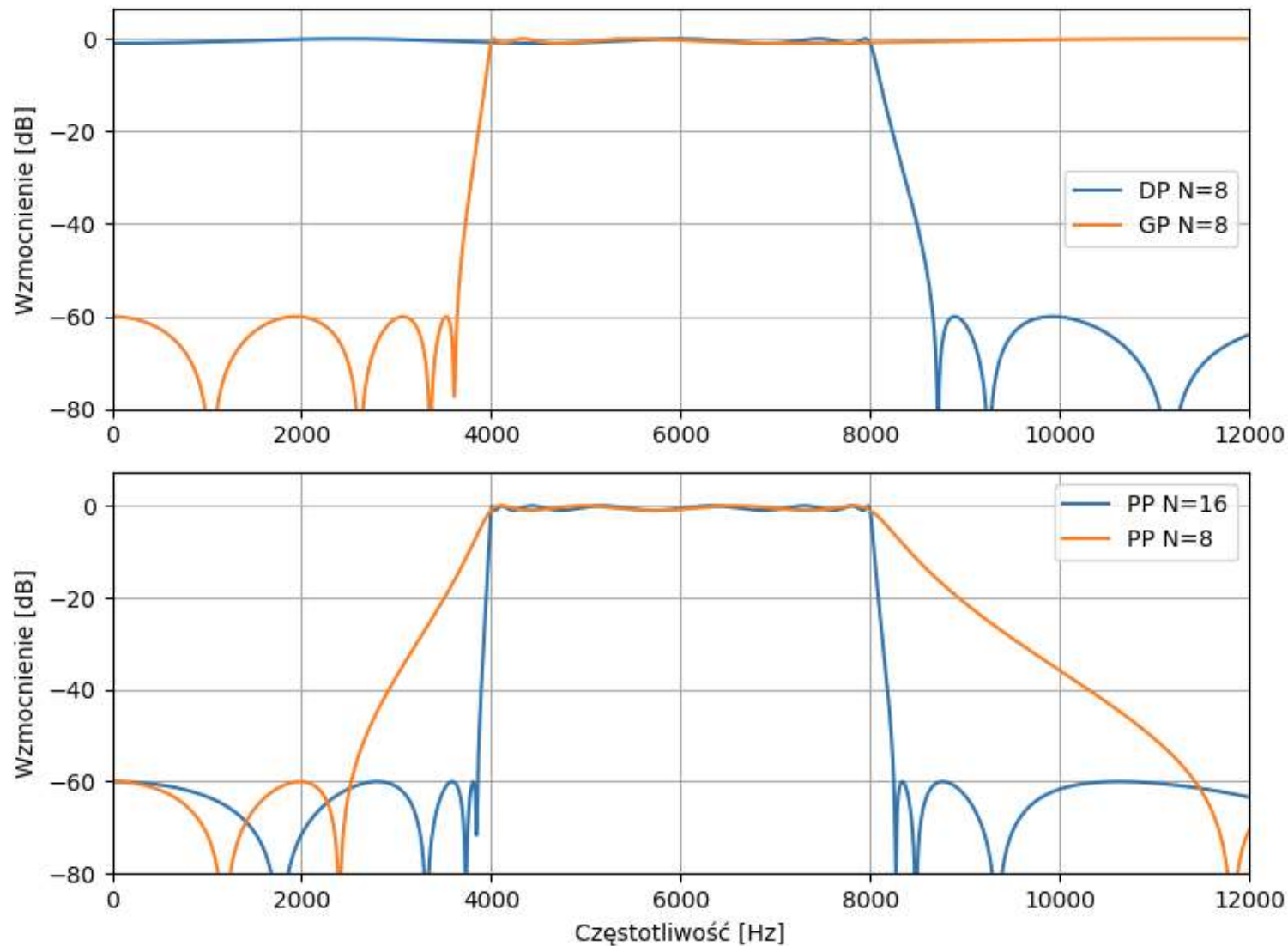
$$s \rightarrow \frac{(\omega_g - \omega_d)s}{s^2 + \omega_d \omega_g}$$

- Zauważmy, że zamiast  $s$  pojawia się  $s^2$ .
- Oznacza to, że rząd filtru PP/PZ **wzrasta dwukrotnie** w stosunku do prototypu DP.
- W programach do projektowania filtrów, zadawany rząd dotyczy czasem prototypu, a wynikowy filtr PP lub PZ ma **dwa razy większy rząd niż zadany!**

- Ta konwencja pozwala zachować zależność między rzędem a tłumieniem filtru.
- Projektując filtr DP lub GP zadanego rzędu, spodziewamy się konkretnego tłumienia (nachylenia charakterystyki w paśmie przejściowym).
- Spodziewamy się, że filtr PP lub PZ tego samego rzędu da takie same tłumienie.
- Nie jest to możliwe, potrzebny jest filtr rzędu około dwukrotnie większego.
- Filtr PP/PZ można traktować jako szeregowe połączenie filtrów DP i GP, wynikowy rząd jest dwukrotnie większy niż rząd składowych filtrów.



# Porównanie charakterystyk dla różnych rzędów:



## Normalizacja wzmacnienia filtrów IIR

- chcemy mieć wzmacnienie 1 w paśmie przepustowym.

Wzmacnienie  $g$  dla pulsacji  $\omega = 2\pi f$  można obliczyć podstawiając do wzoru na transmitancję:

$$z \rightarrow e^{j\omega} = \cos \omega + j \sin \omega$$

- Dla DP i PZ:  $\omega = 0$       Dla GP:  $\omega = \pi$
- Dla PZ:  $\omega$  w środku pasma przepustowego

$$g_{DP} = \frac{\sum_{i=0}^N b_i}{\sum_{i=0}^N a_i}$$

$$g_{GP} = \frac{\sum_{i=0}^N (-1)^i b_i}{\sum_{i=0}^N (-1)^i a_i}$$

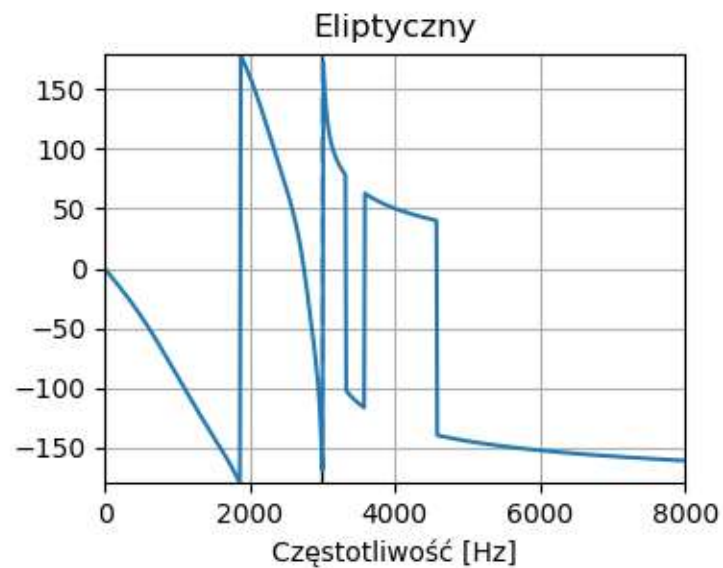
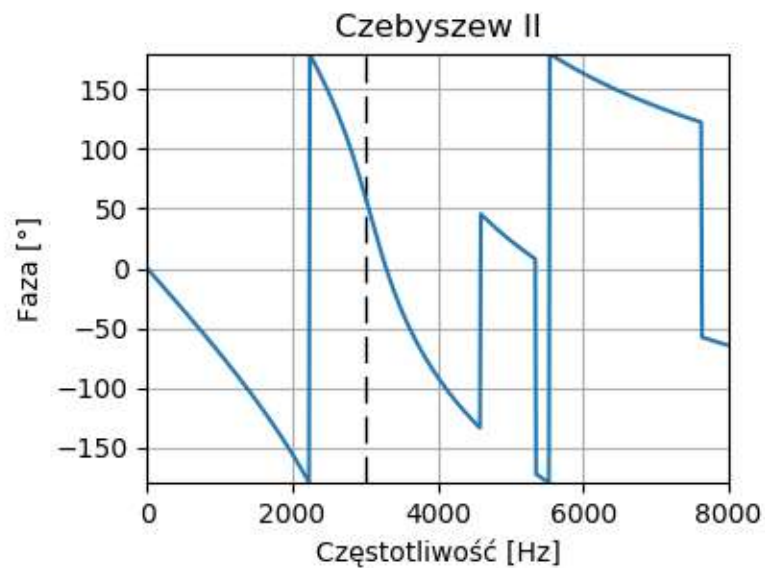
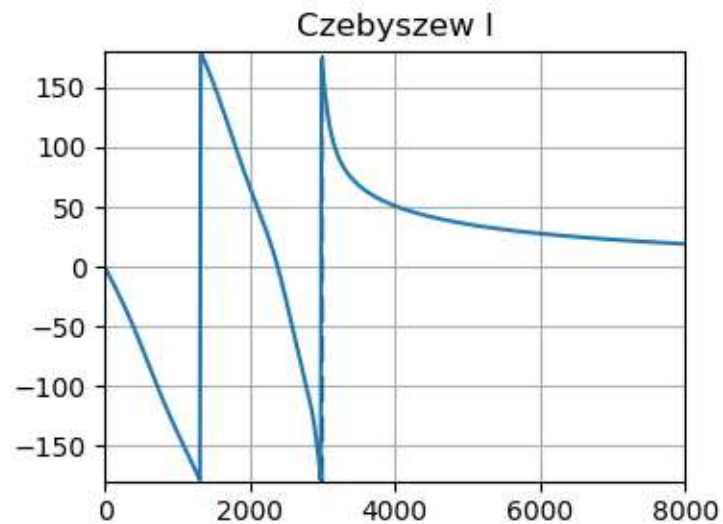
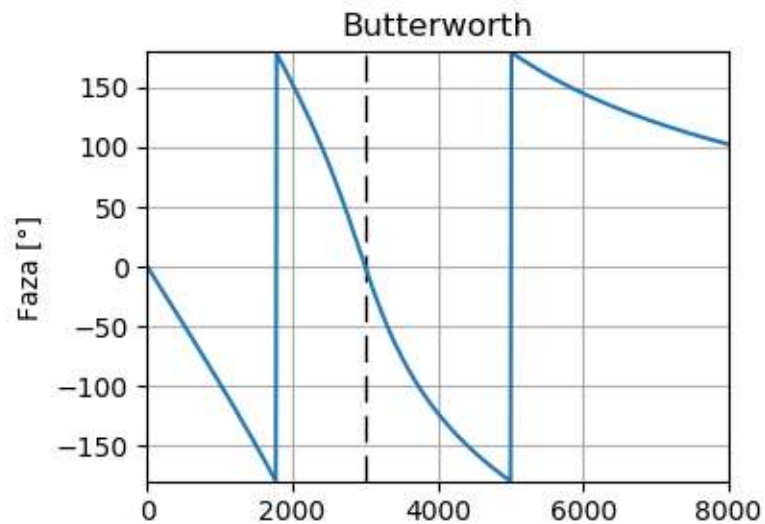
## Normalizacja wzmacnienia filtrów IIR

- Obliczamy wzmacnienie  $g$  z transmitancji
  - dla DP, GP, PZ – wzory uproszczone,
  - dla PP – trzeba podstawić do transmitancji.
- Dzielimy współczynniki  $b$  przez wzmacnienie  $g$ .
- Wynikowy filtr ma wzmacnienie 1 w paśmie przepustowym.

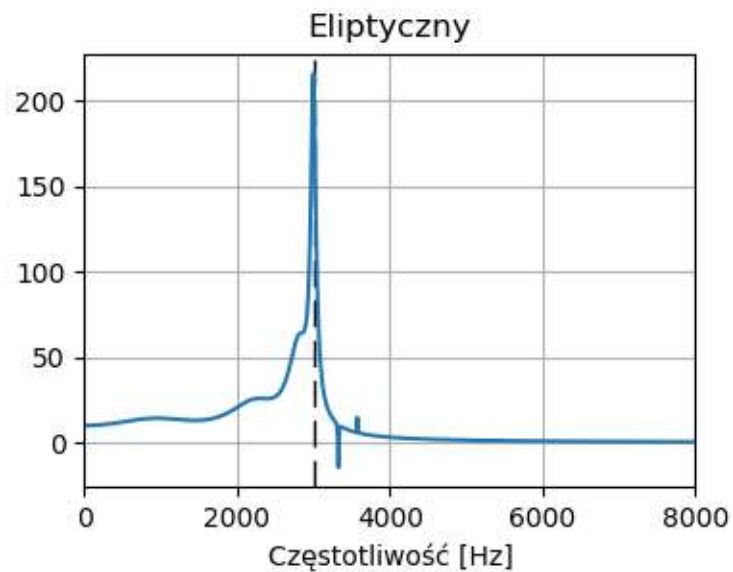
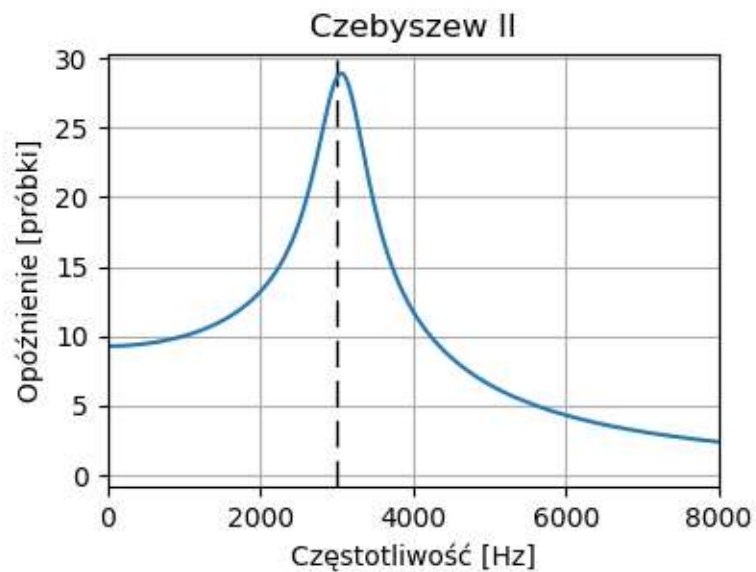
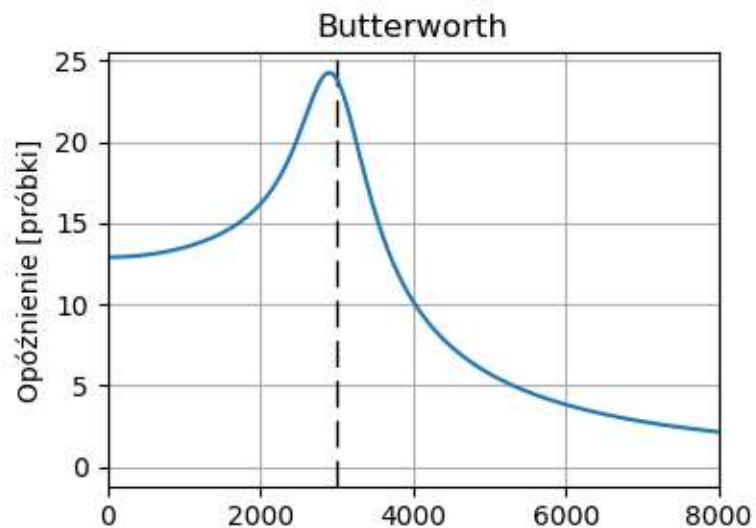
Uwaga: nie dzielić z rozpędu współczynników  $a$  (liczba  $\cdot$  ułamek = liczba  $\cdot$  licznik ułamka).

- Przypomnijmy cechy (typowych) filtrów FIR:
  - liniowa charakterystyka fazowa,
  - stałe opóźnienie grupowe,
  - brak zniekształceń fazowych.
- Niestety, filtry IIR nie zapewniają tych właściwości.
- Filtry IIR zniekształcają fazę sygnału. Opóźnienie zależy od częstotliwości.
- Dla sygnałów o złożonym widmie (mowa, muzyka), sygnał na wyjściu filtru składa się z komponentów widmowych, które pojawiły się na wejściu w różnym czasie!

# Charakterystyki fazowe filtrów IIR (DP 3 kHz, $N=8$ )

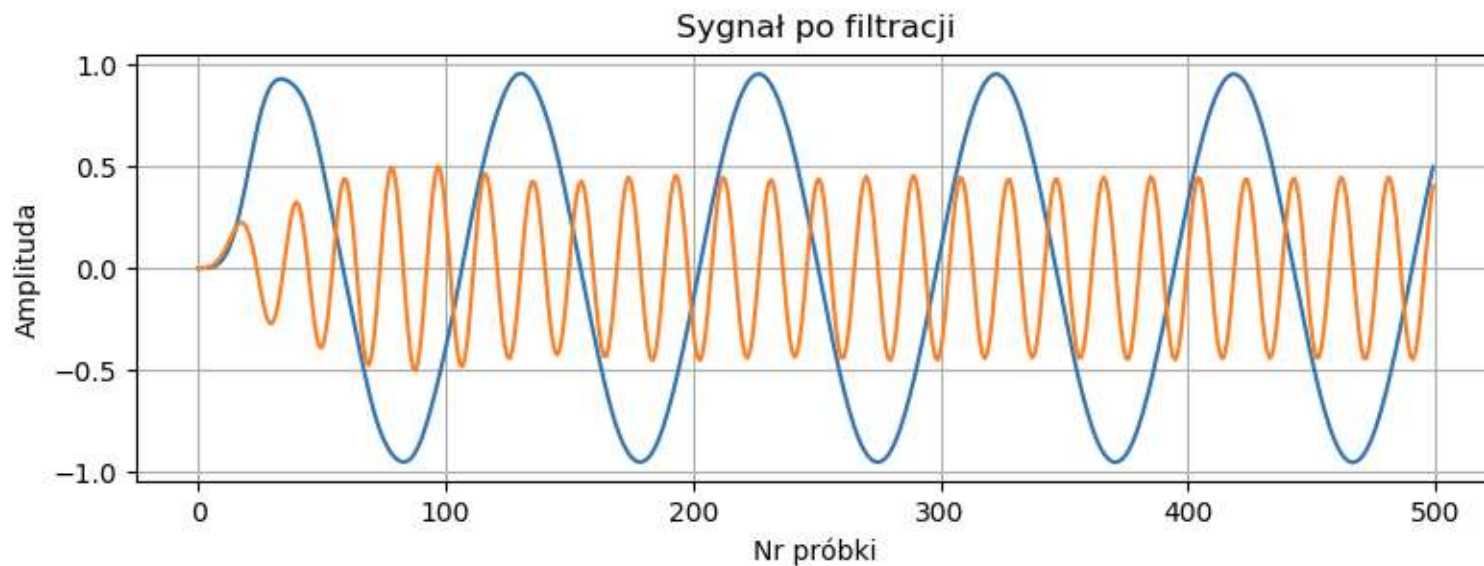
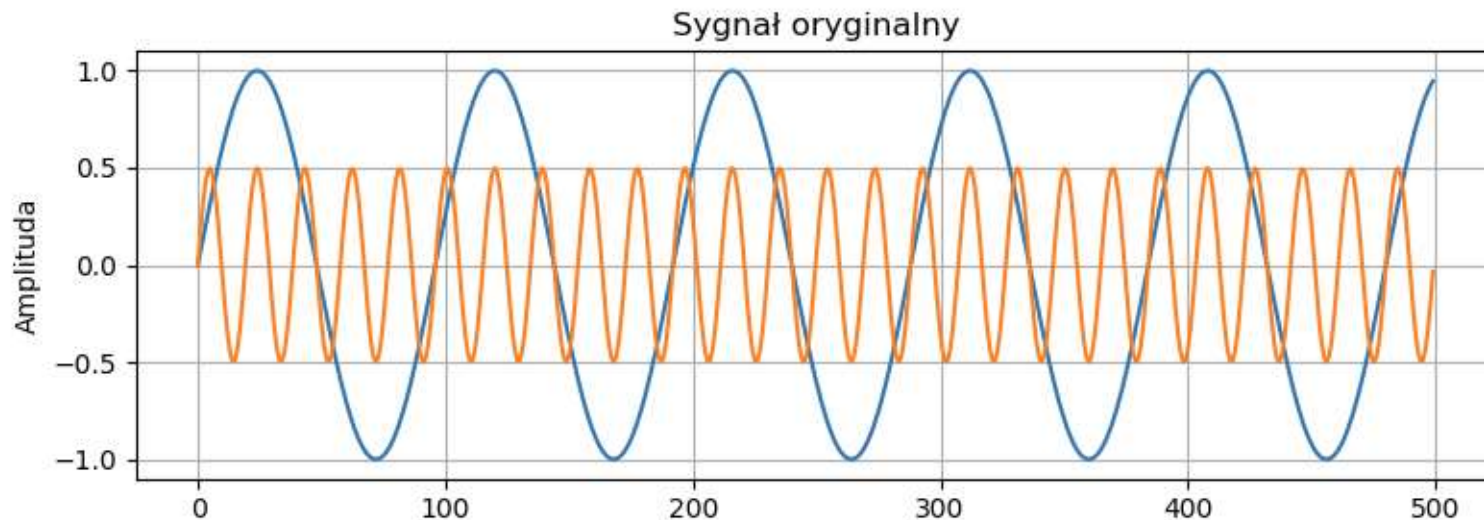


# Opóźnienie grupowe filtrów IIR (DP 3 kHz, $N=8$ )



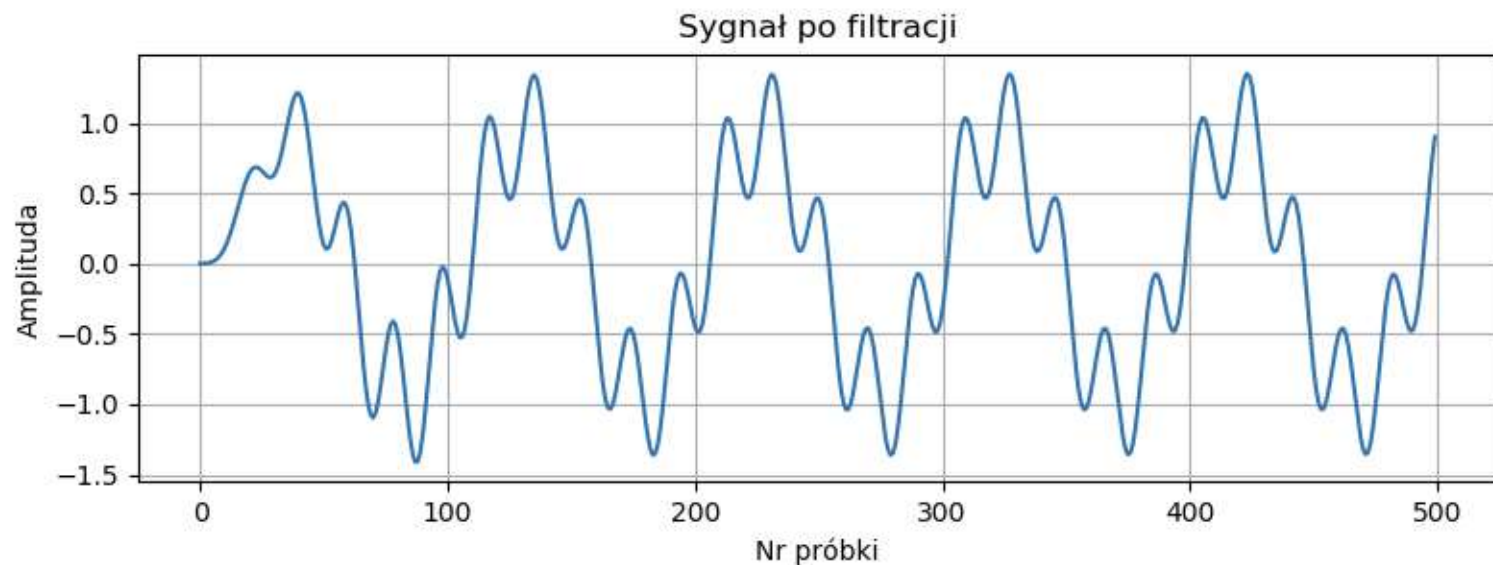
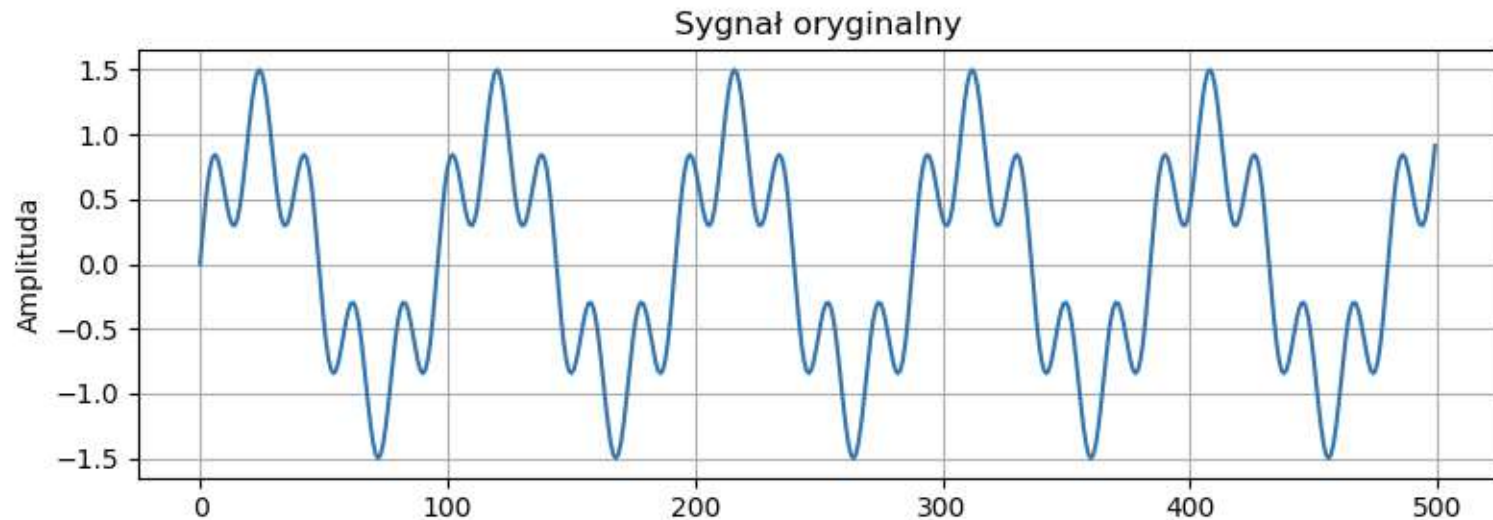
Ilustracja efektu opóźnienia:

$\sin(500 \text{ Hz}) + 0,5 \sin(2,5 \text{ kHz})$ ; filtr DP 3 kHz (eliptyczny)





Sygnal po filtracji, na skutek zniekształceń fazowych, zmienia swój kształt!





Jak naprawdę działają filtry IIR? W jaki sposób filtrują wybrany zakres częstotliwości?

Mamy dwa sygnały sinus.

- Identyczna częstotliwość.
- Identyczna amplituda.

Dodajemy oba sygnały do siebie.

Jaki będzie wynik tego dodawania?

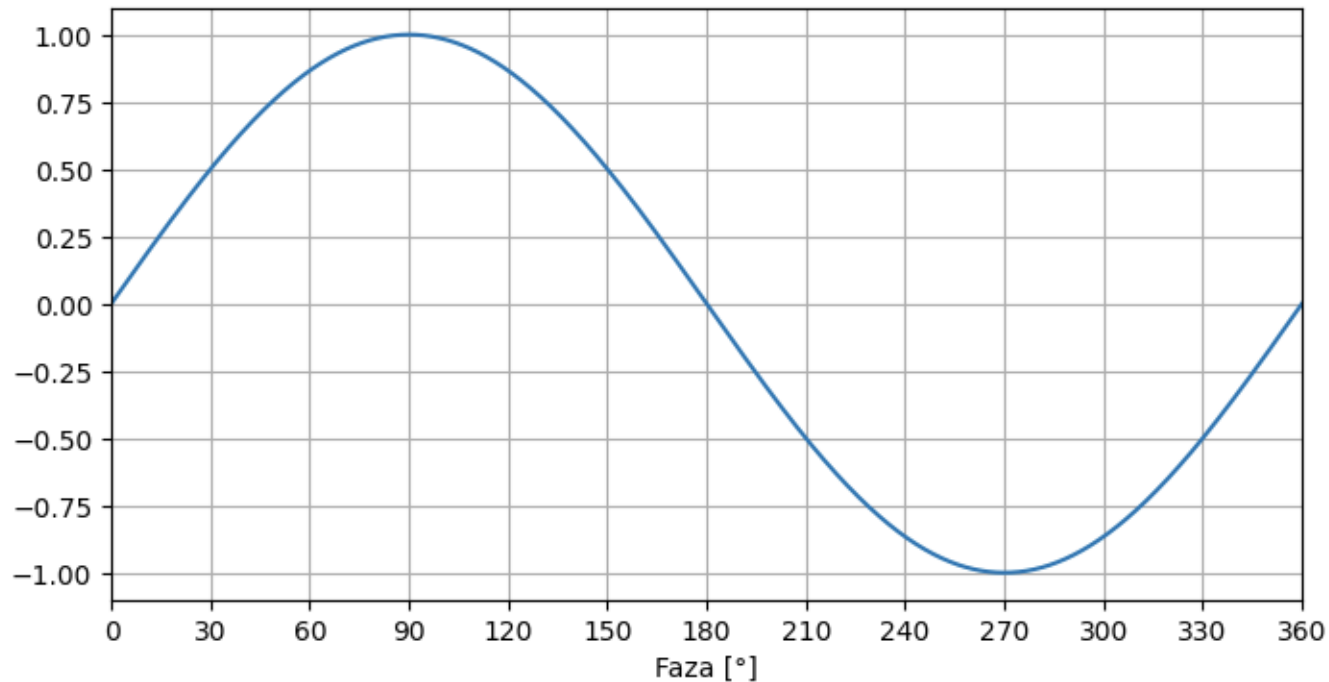
Odpowiedź na kolejnych slajdach.

(Tak, to jest podchwytliwe pytanie.)

Faza sygnału sinus określa położenie wewnątrz okresu.

Faza = argument  $x$  sinusa  $\sin(x)$ .

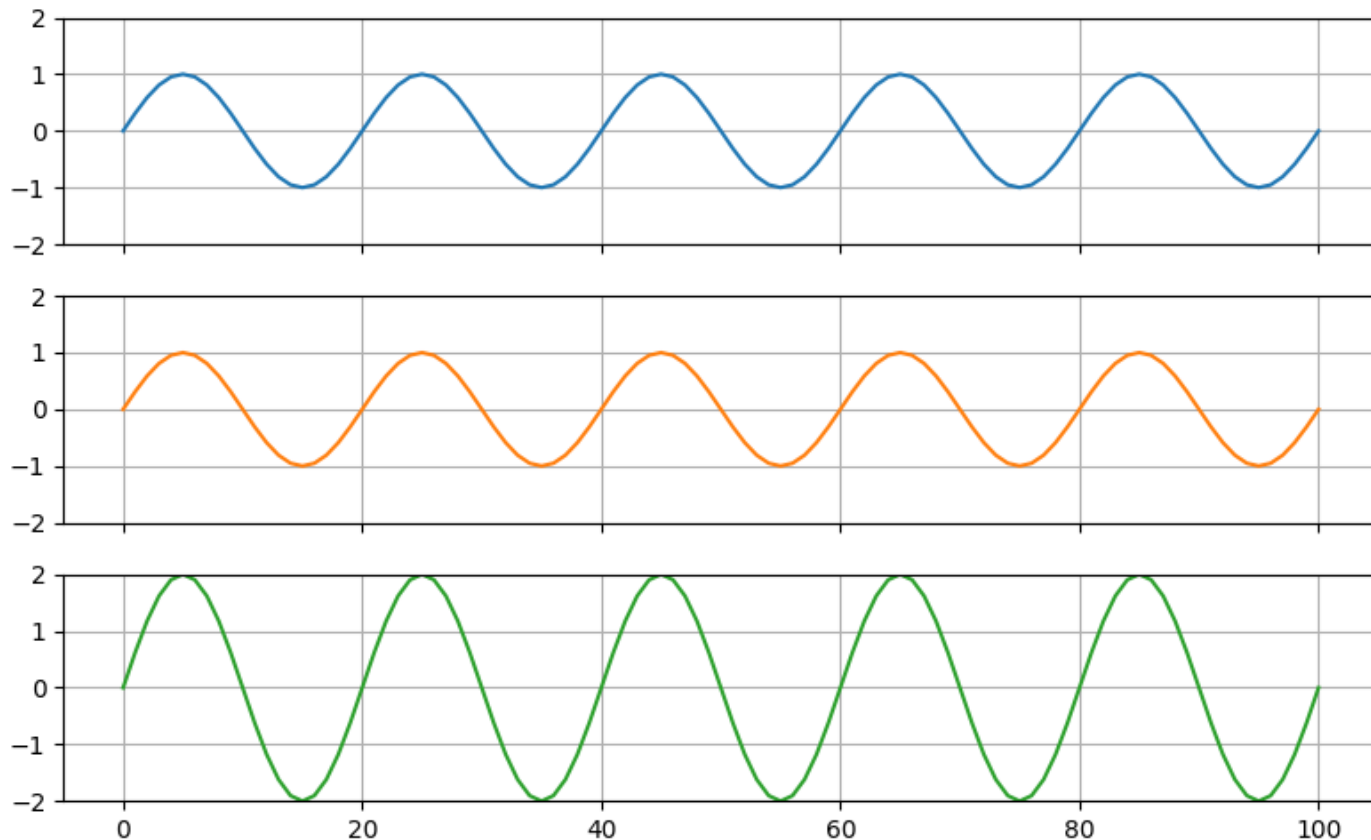
Kąt w stopniach (0-360) lub w radianach (0- $2\pi$ ).



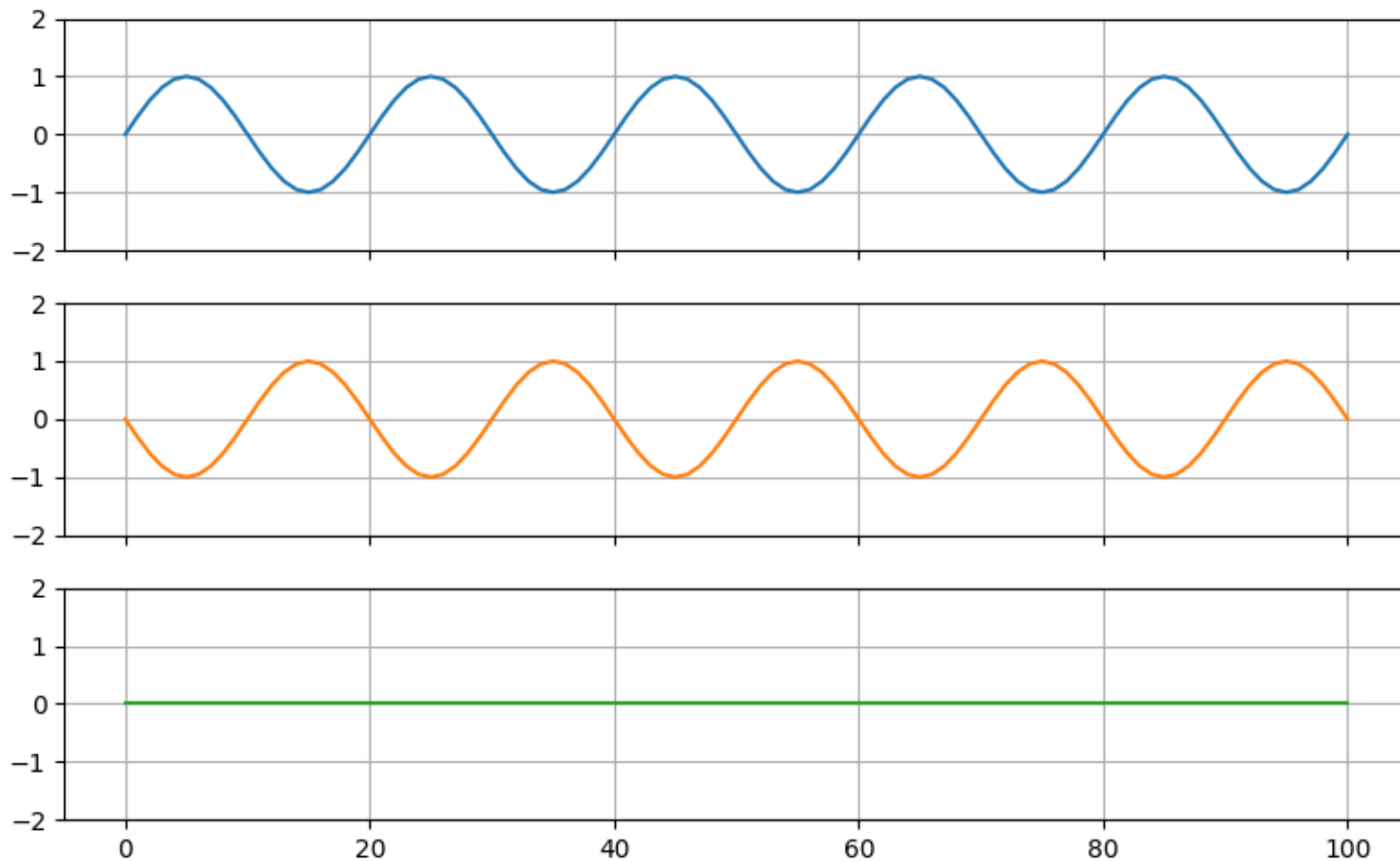
Faza pojedynczego sygnału jest bez znaczenia.

Dla dwóch sygnałów możemy mówić o **różnicy fazy** lub o **przesunięciu fazowym** (*phase shift*).

Suma dwóch sinusów o zgodnej fazie:

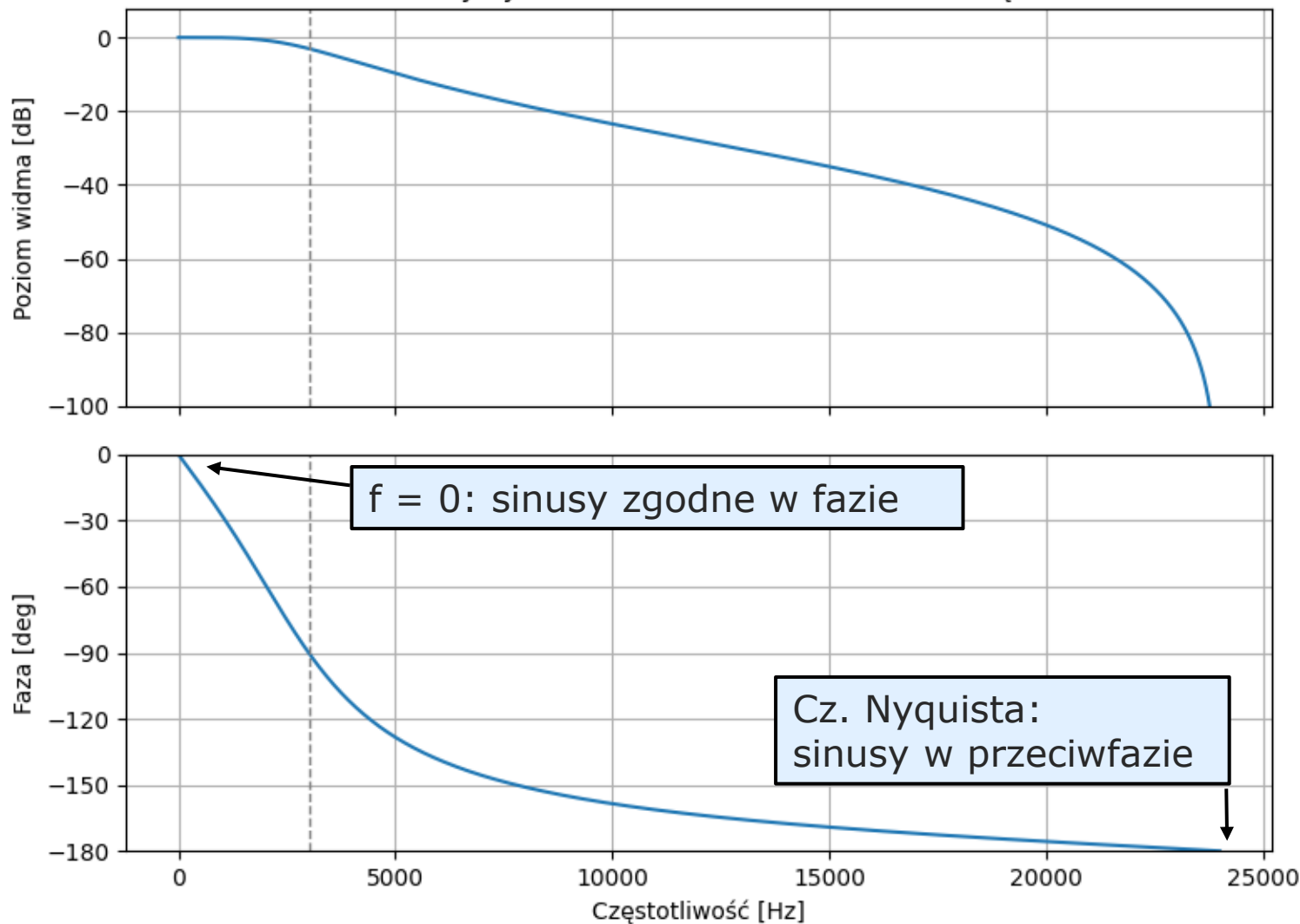


Jeżeli jeden z sinusów jest przesunięty w fazie względem drugiego o dokładnie **pół okresu** ( $180^\circ$ ,  $\pi$ ), sygnały są w **przeciwfazie** – ich suma jest zerowa.




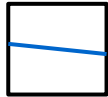

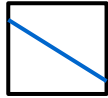

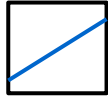

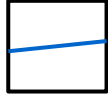

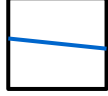
- Złożony sygnał można traktować jako sumę sinusów o różnych częstotliwościach i amplitudach.
- Sumujemy te sinusy z kopiami przesuniętymi fazowo.
- W paśmie **przepustowym** filtru – staramy się, aby różnica fazy była zerowa (**zgodna faza**).
- W paśmie **zaporowym** filtru – staramy się, aby różnica fazy była równa  $180^\circ$  (**przeciwfaza**), tak aby składowe zostały **wytlumione**.
- I tak właśnie działają filtry IIR – sumują sygnał z przetworzoną fazowo kopią, manipulując fazami składowych tak, aby niektóre wytłumić, a niektóre pozostawić. Oto cała magia.

Charakterystyka filtru Butterwortha DP 3 kHz, rzędu 2



Czy można dokonać filtracji sygnału filtrem IIR tak, aby nie powstały zniekształcenia fazowe?

- Tak, ale...
- ... ale pod warunkiem, że mamy cały sygnał do dyspozycji (przetwarzanie *offline*).
- Nie da się tego zrobić jeżeli wciąż napływają nowe próbki sygnału (przetwarzanie *online*).
- W takiej sytuacji, jeżeli nie dopuszczamy zniekształceń fazowych, zostają tylko filtry FIR.

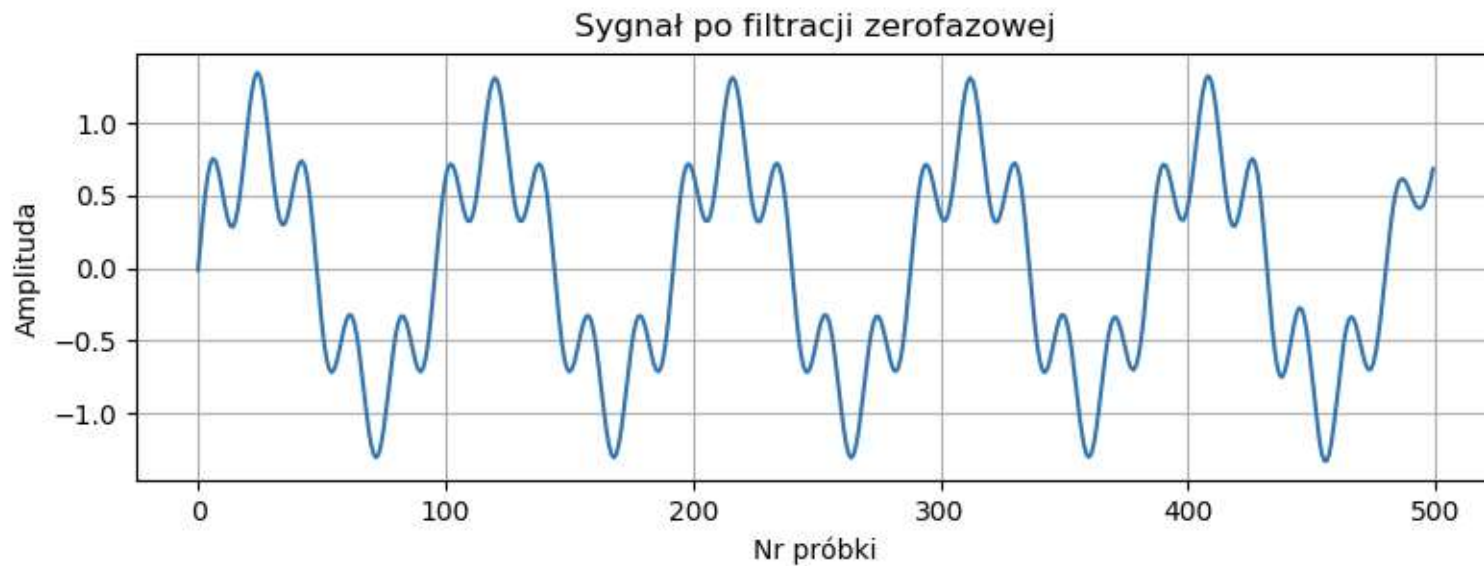
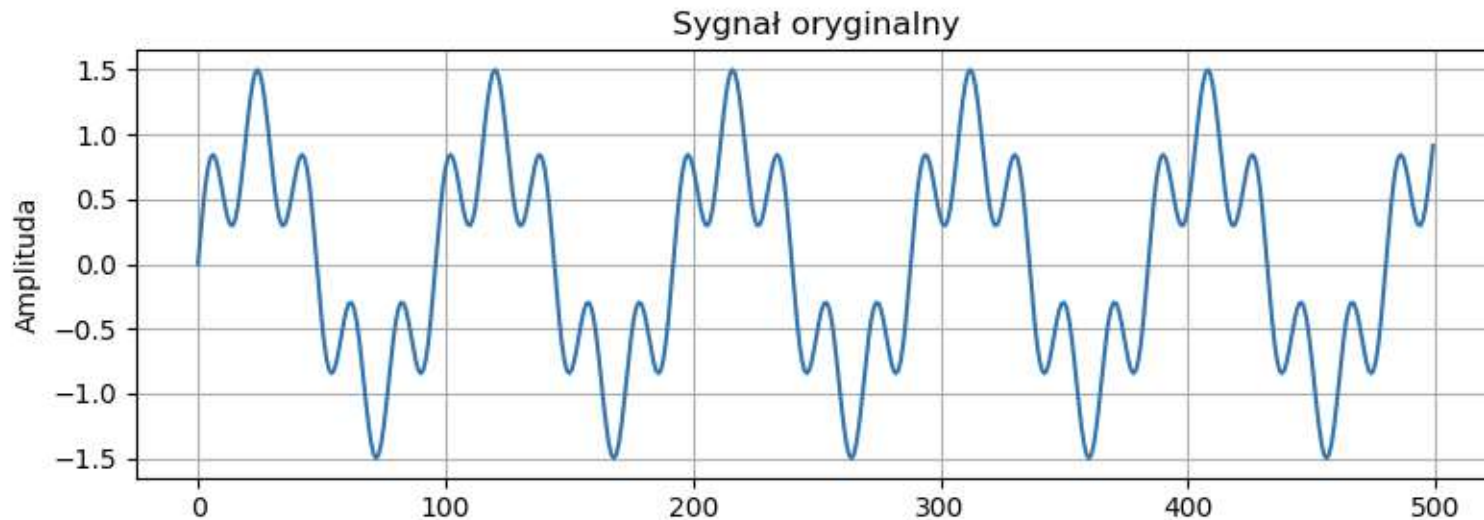
	Amp.	Faza
0. Sygnał oryginalny		
1. Filtracja sygnału.		
2. Odwrócenie kolejności próbek.		
3. Filtracja po raz drugi.		
4. Ponowne odwrócenie kolejności.		

Zniekształcenia fazowe nakładają się dwukrotnie, ale z przeciwnym znakiem, zatem kompensują się nawzajem.

Jest to **filtracja zerofazowa**.



# Wynik filtracji zerofazowej – brak zniekształceń fazy



## Efekty kwantyzacji współczynników

- Z projektu filtru wychodzą nam pewne wartości współczynników filtru ( $b$ ,  $a$ ).
- Zapisujemy te liczby ze skończoną precyzją. Np. liczby *double*: ok. 15 miejsc po przecinku.
- Zatem zapisane w programie wartości współczynników nieco różnią się od rzeczywistych.
- **Kwantyzacja** – „dosunięcie” wartości liczby do najbliższej liczby możliwej do zapisania.
- Różnica wyników między projektowanym a zaprogramowanym filtrem tworzy **szum kwantyzacji**.

Efekt szumu kwantyzacji  $e[n]$  jest też obecny w filtrach FIR, ale tam jest ograniczony do wyjścia filtru:

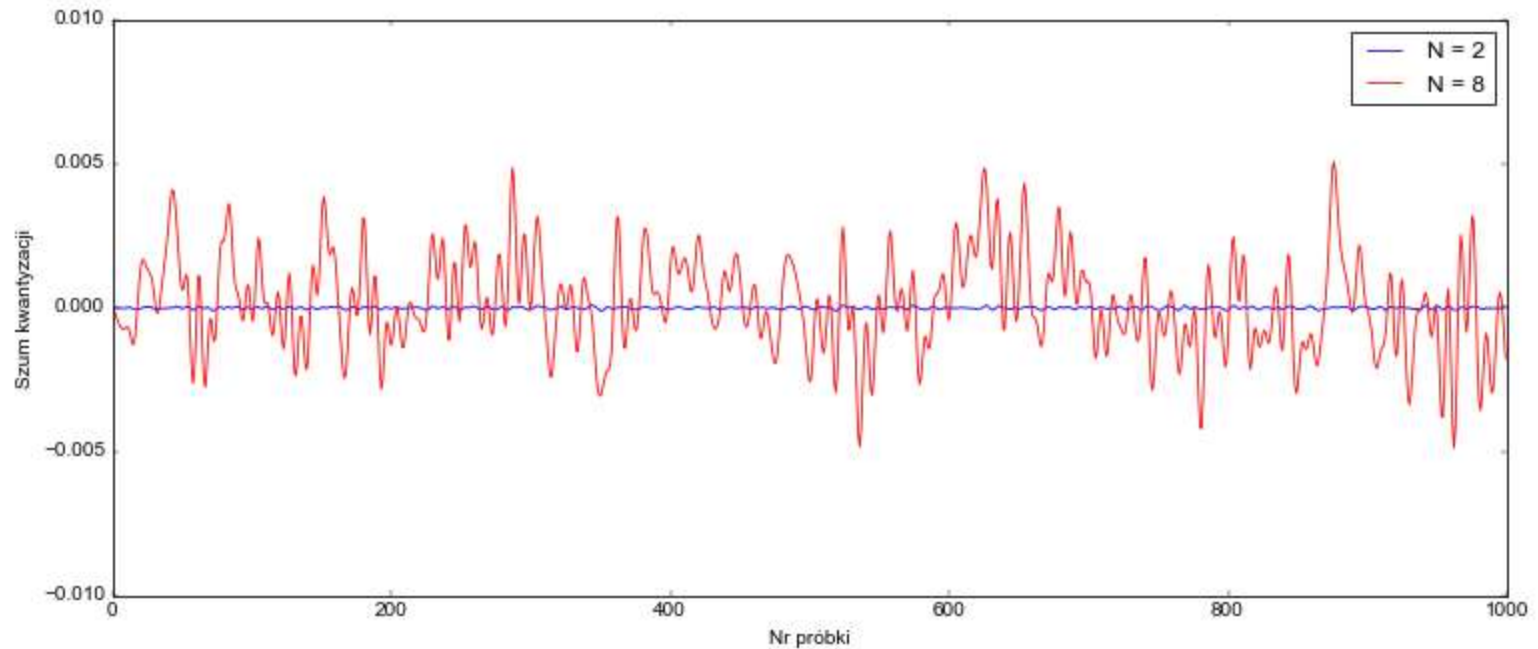
$$y[n] = \sum_{i=0}^{N-1} (h_i x[n-i]) + e[n]$$

W filtrze IIR, wynik obarczony błędem jest brany pod uwagę w obliczeniach dla następnych próbek!

$$y[n] = \sum_{i=0}^N (b_i x[n-i]) - \sum_{i=1}^N (a_i (y[n-i] + e[n-i])) + e[n]$$

Im większy rząd filtru, tym większy szum kwantyzacji (błąd dłużej „krąży” w filtrze).

Przykład: szum kwantyzacji filtra IIR w implementacji stałoprzecinkowej, dla rzędu 2 i 8.



Szum kwantyzacji wpływa na transmitancję filtra:

$$a_i y[n-i] \rightarrow a_i (y[n-i] + e[n-i]) = a_i y[n-i] + a_i e[n-i]$$

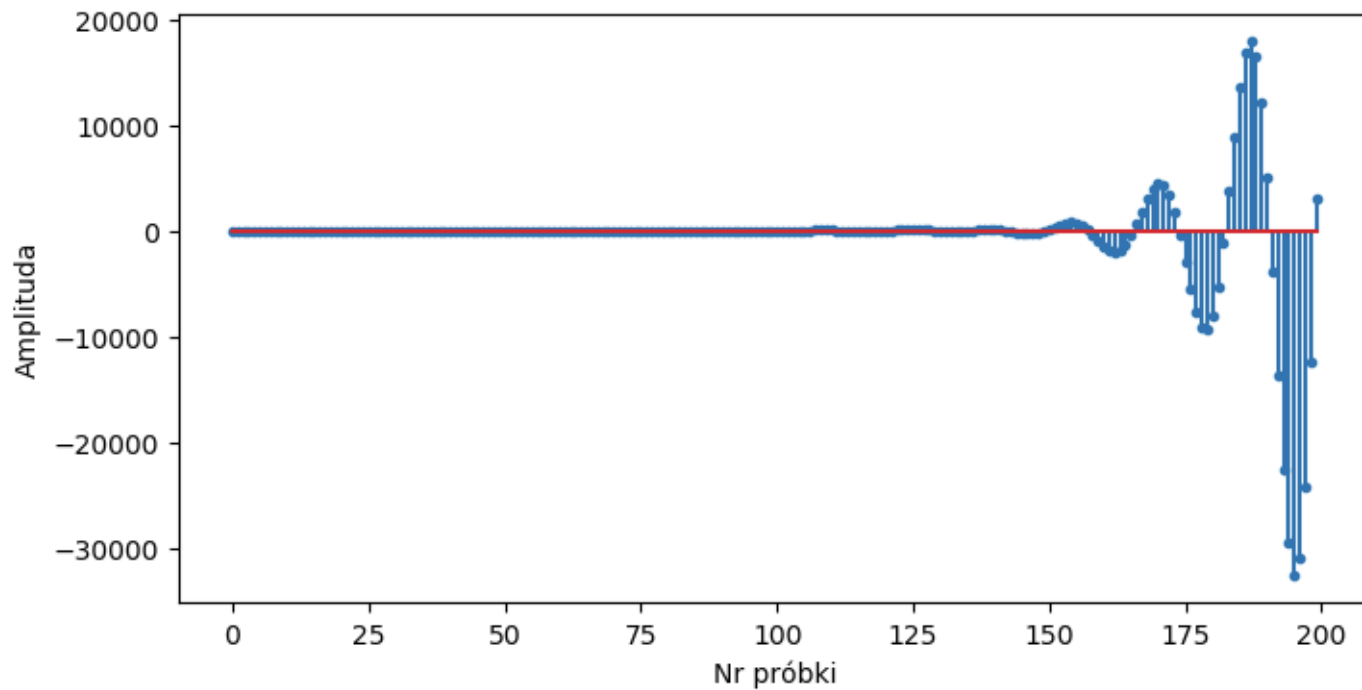
$$a_i[n] = \frac{a_i}{1 + \sum_{i=1}^N a_i e[n-i]}$$

$$b_i[n] = \frac{b_i}{1 + \sum_{i=1}^N a_i e[n-i]}$$

- Szum zmienia współczynniki filtra.
- Zmienia więc także zera i bieguny.
- Szum może zwiększyć moduł bieguna powyżej 1 - filtr staje się **niestabilny**, chociaż został zaprojektowany jako stabilny!

Filtr DP 3 kHz, eliptyczny, 20. rzędu.

Odpowiedź impulsowa:



(ups... 🙄 )

- Ryzyko utraty stabilności z powodu błędów kwantyzacji wzrasta z rzędem filtru.
- Filtry niskich rzędów ( $<8$ ) nie mają zwykle takiego problemu.
- Lepiej więc stosować filtry niskiego rzędu.
- Ale co zrobić, jeżeli charakterystyka filtru wymaga zastosowania wysokiego rzędu, np. 20?
- Jest na to sposób.

Transmitancja filtru IIR zapisana w formie ZPK:

$$H(z) = k \frac{(z - r_1)(z - r_2) \dots (z - r_N)}{(z - p_1)(z - p_2) \dots (z - p_N)}$$

Możemy sobie ten ułamek „pociąć” na kawałki:

$$\begin{aligned} H(z) &= \\ &= k \frac{(z - r_1)(z - r_2)}{(z - p_1)(z - p_2)} \cdot \frac{(z - r_3)(z - r_4)}{(z - p_3)(z - p_4)} \cdot \dots \cdot \frac{(z - r_{N-1})(z - r_N)}{(z - p_{N-1})(z - p_N)} \end{aligned}$$

Z matematycznego punktu widzenia, niczego to nie zmienia (rym niezamierzony).

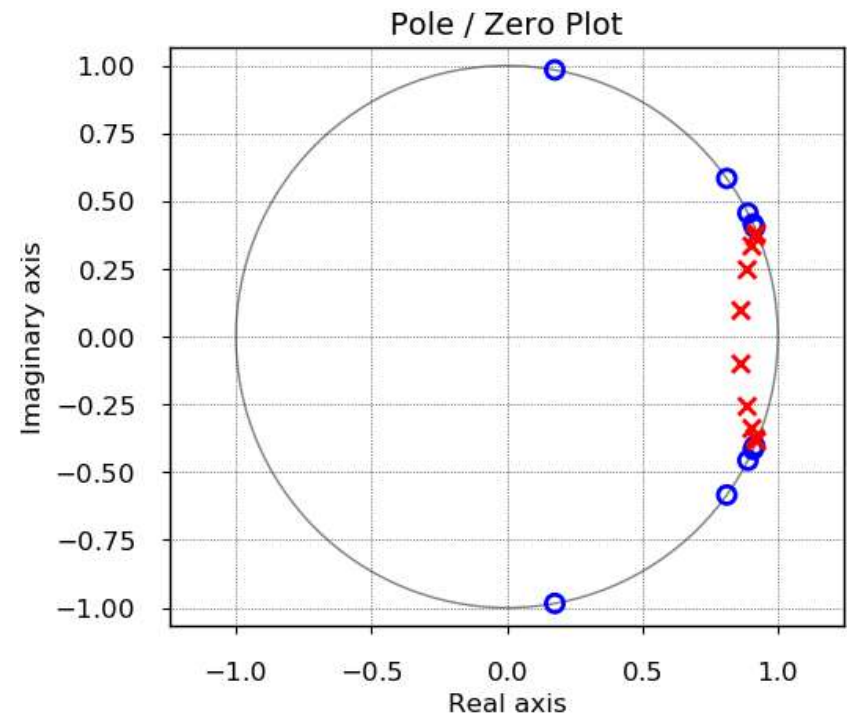


- Każdy „fragment” transmitancji reprezentuje filtr IIR drugiego rzędu:

$$H(z) = \frac{(z - r_1)(z - r_2)}{(z - p_1)(z - p_2)} = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}}$$

- Taki filtr składowy nazywa się **sekcją drugiego rzędu (SOS, second order section)**, inaczej filtrem **dwukwadratowym** lub **bikwadratowym (biquad)** (jeden „kwadrat” w liczniku, drugi w mianowniku).
- Filtr IIR rzędu  $N$  można podzielić na  $N/2$  sekcji drugiego rzędu (ostatnia sekcja może być 1. rzędu jeżeli  $N$  jest nieparzyste).

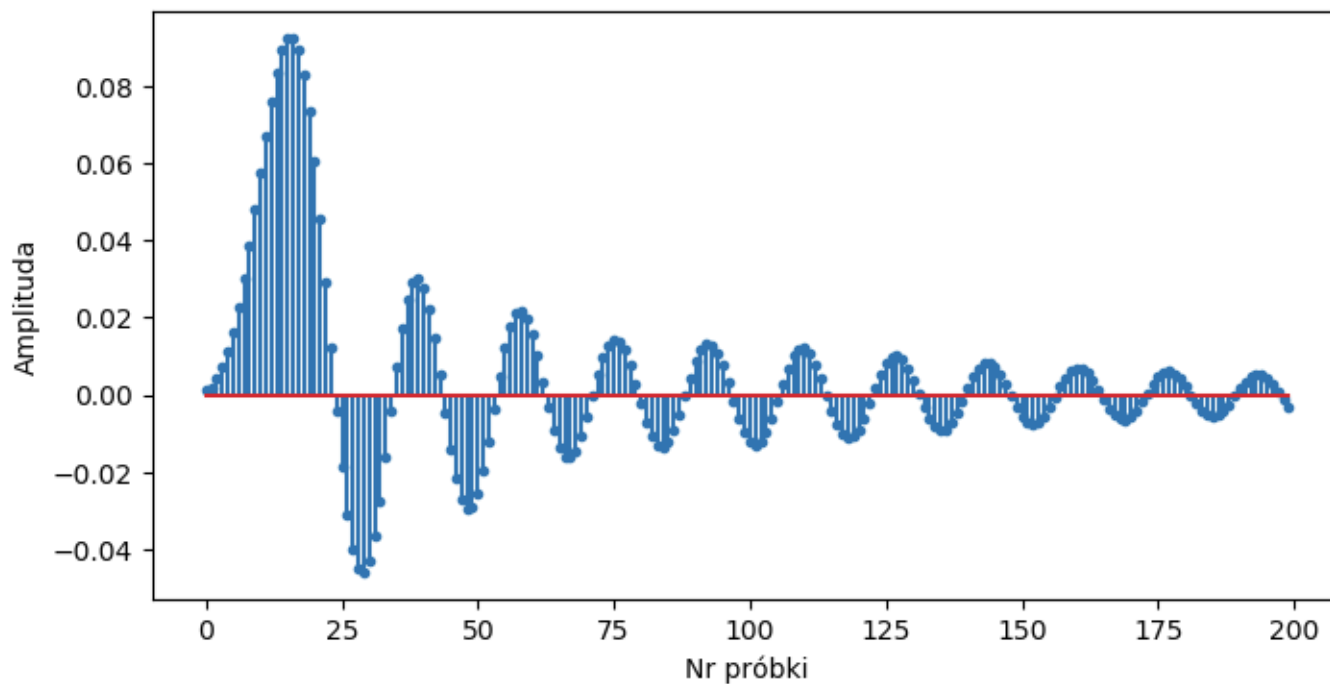
- Sposób podziału na sekcje jest w zasadzie dowolny.
- Wykres zer i biegunów jest symetryczny względem osi poziomej.
- Zera tworzą pary zespolone sprzężone.
- Bieguny tworzą pary zespolone sprzężone.
- W jednej sekcji zazwyczaj umieszcza się:  
parę sprzężonych zer  
i odpowiadającą im parę  
sprzężonych biegunów.



Co nam daje taka modyfikacja?

- Każda sekcja ma własną pętlę sprzężenia zwrotnego.
- Każda pętla obejmuje dwa opóźnienia zamiast  $N$ .
- Wpływ szumu kwantyzacji na wyniki filtracji jest zredukowany do pętli pojedynczych sekcji.
- W praktyce, **zabezpieczamy się przed utratą stabilności** filtru.
- Struktura SOS, inaczej nazywana **strukturą kaskadową**, jest zalecana przy projektowaniu filtrów IIR wyższych rzędów ( $N > 8$ , choć nie jest to sztywne zalecenie).

Ten sam filtr co poprzednio ( $N=20$ ), ale zaprojektowany jako sekcje drugiego rzędu. Odpowiedź impulsowa:



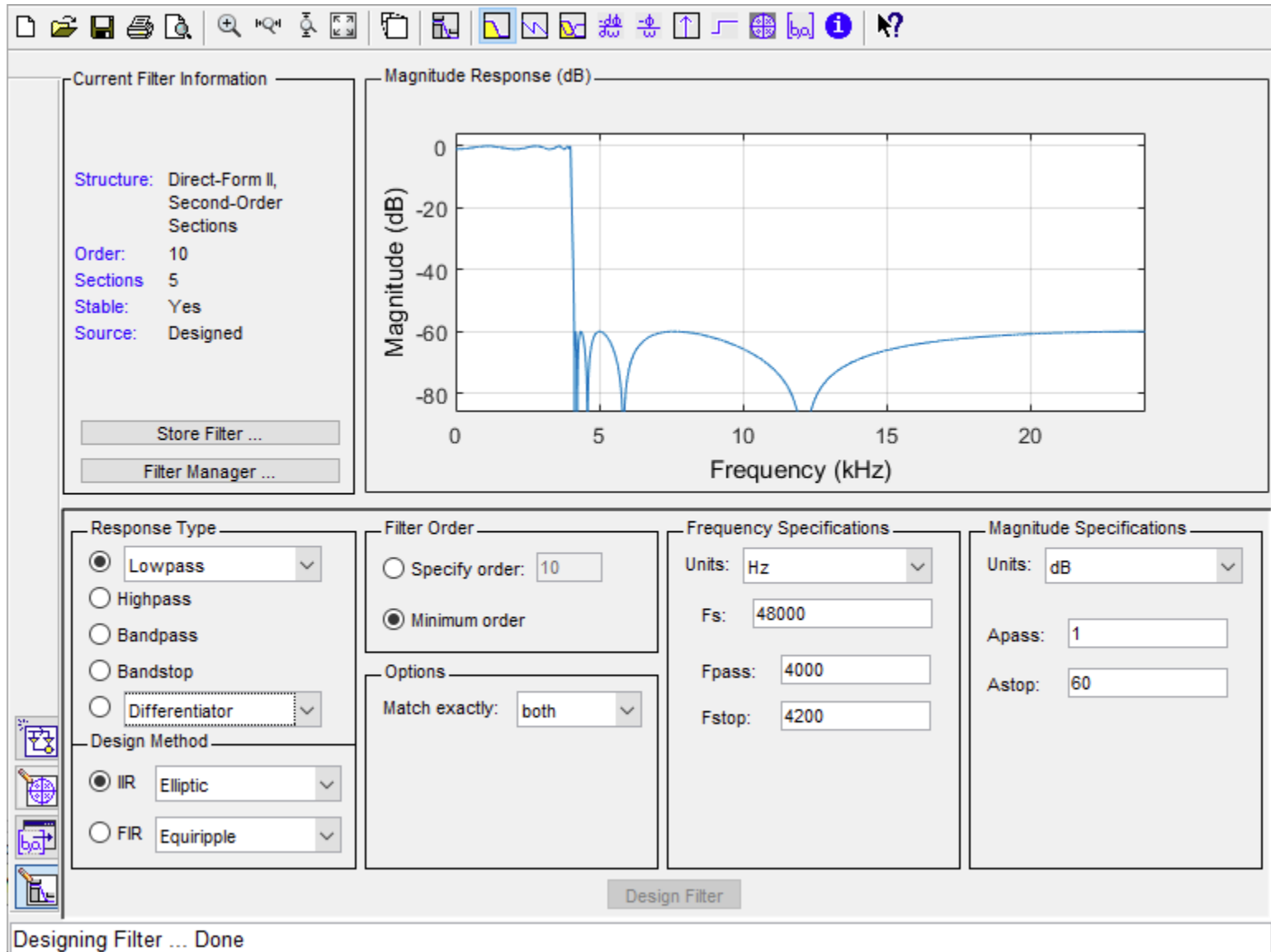
i wszystko jest jak trzeba 👍 .

Trzy możliwe opisy filtru IIR:

- $(b, a)$  (TF) – współczynniki licznika i mianownika,
- ZPK – zera, bieguny i stałe wzmocnienie,
- SOS – sekcje drugiego rzędu
  - $N/2$  sekcji,
  - dla każdej sekcji:  $[k, b_0, b_1, b_2, 1, a_1, a_2]$   
(jedynekę często się pomija),
  - wzmocnienie  $k$  jest czasem „wciągane” do licznika

Każdy z tych trzech opisów można przeliczyć na pozostałe, ale nie powinno się przekształcać  $(b, a)$  w SOS, zwłaszcza dla filtrów wyższego rzędu.

# Graficzne projektowanie filtrów: Matlab, *filterDesigner*



## Przykład projektowania filtrów IIR

Projektowanie za pomocą języka *Python* i modułów *SciPy* + *NumPy*.

Dokumentacja:

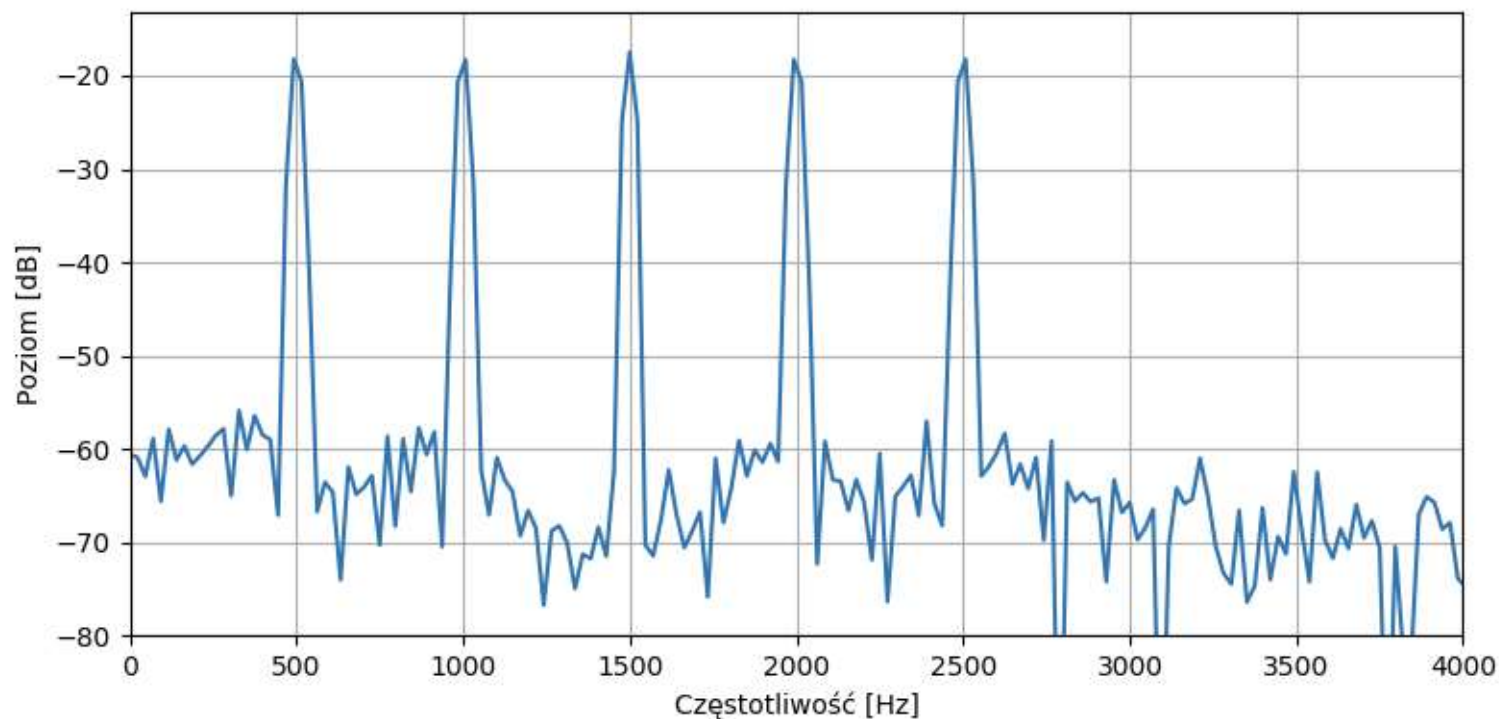
<https://docs.scipy.org/doc/scipy/reference/signal.html>

Wszystkie przykłady zakładają, że wcześniej wykonano instrukcje:

```
import numpy as np
import scipy.signal as sig
```

Częstotliwość próbkowania = 48 kHz

Sygnal testowy – suma pięciu „sinusów” + szum,  
częstotliwości: 500, 1000, 1500, 2000, 2500 Hz





## Projekt 1 – usuwamy dwa ostatnie sinusy.

- Filtr dolnoprzepustowy.
- Częstotliwość graniczna: 1700 Hz.
- Rząd filtru – ustalmy sobie  $N = 8$ .
- Typ – wybieramy eliptyczny.
- Zafalowania – zakładamy sobie  $R_p = 1$  dB, z wykresu: stosunek sygnału do szumu ok. 50 dB, więc  $R_z = 60$  dB powinno wystarczyć.
- Wynik w formie  $(b, a)$  – zakładamy, że filtr nie straci stabilności.

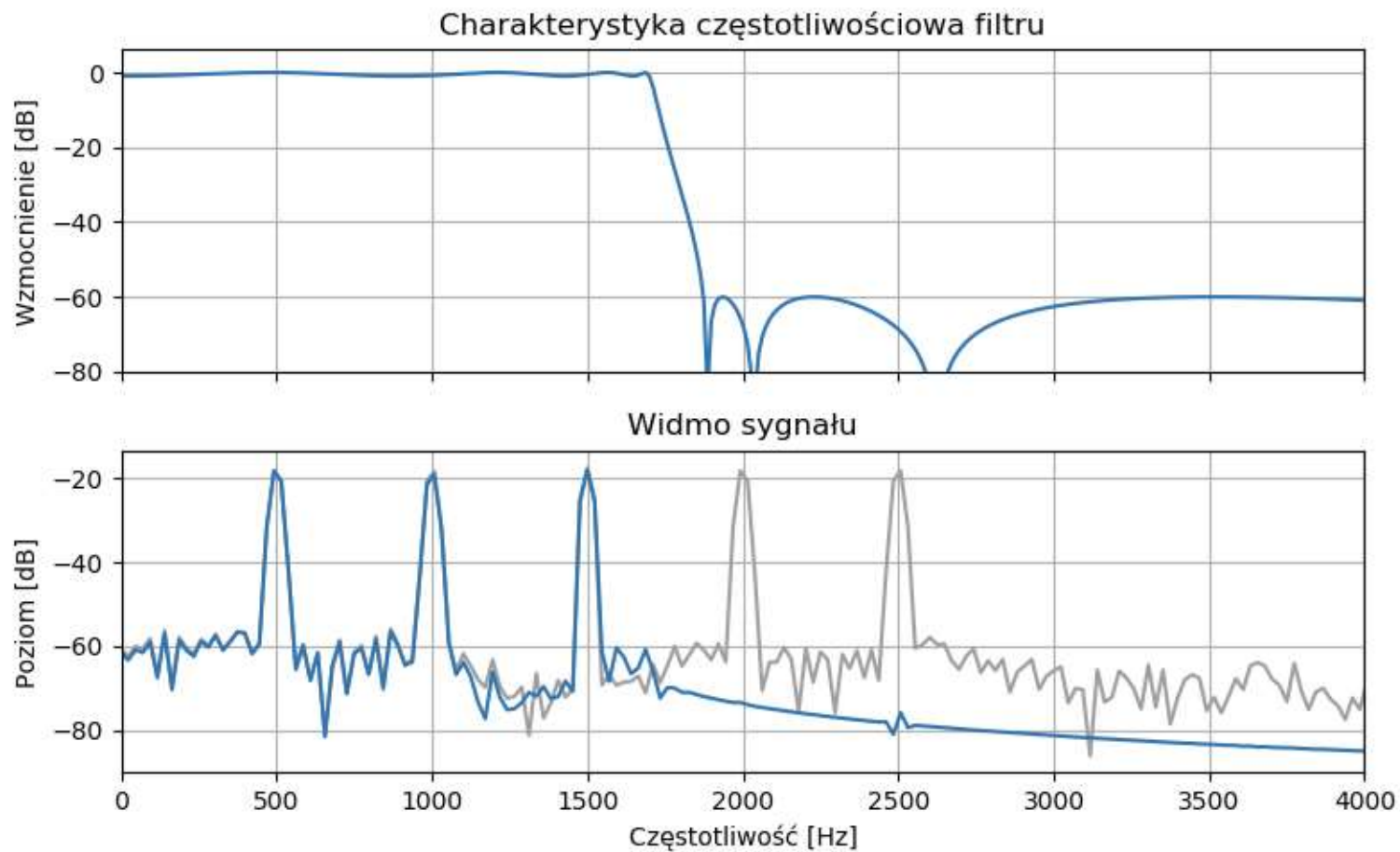
## Obliczenie współczynników:

```
b, a = sig.iirfilter(8, 1700, btype='lowpass', ftype='ellip',  
                    rp=1, rs=60, fs=48000)
```

- rząd filtru (8),
- częstotliwość graniczna (1700 Hz),
- charakterystyka (*lowpass, highpass, bandpass, bandstop*)
- typ (*butter, cheby1, cheby2, ellip*)
- zafalowania w paśmie przepustowym (rp = 1 dB)  
i w paśmie zaporowym (rs = 60 dB)
- częstotliwość próbkowania (fs = 48000 Hz)

# Projekt 1 – wynik filtracji sygnału x:

```
y = sig.lfilter(b, a, x)
```



## Projekt 2 – usuwamy pierwszy i ostatni prążek.

- Filtr pasmowo-przepustowy.
- Częstotliwości graniczne: 750 Hz, 2250 Hz.
- Tym razem zakładamy szerokość pasma przejściowego równą 250 Hz.
- Typ – wybieramy eliptyczny.
- Zafalowania:  $R_p = 1$  dB,  $R_z = 60$  dB.
- Wynik w formie SOS. Nie mamy pewności jaki będzie wynikowy rząd filtru, zabezpieczamy się przed utratą stabilności.

## Obliczenie współczynników:

```
sos2 = sig.iirdesign(wp=(750, 2250), ws=(500, 2500), gpass=1,  
                  gstop=60, ftype='ellip', output='sos', fs=48000)
```

- cz. graniczne pasma przepustowego (750, 2250 Hz),
- cz. graniczne pasma zaporowego (500, 2500 Hz),
- zafalowania w p. przepustowym (gpass = 1 dB)  
i w p. zaporowym (gstop = 60 dB),
- typ – jak poprzednio,
- wynik w formacie SOS,
- częstotliwość próbkowania (fs = 48000 Hz)

## Wynik obliczeń w formie SOS:

```
[ [ 0.0006 -0.0011 0.0006 1. -1.9186 0.9468 ]
  [ 1. 0. -1. 1. -1.8949 0.9488 ]
  [ 1. -1.9965 1. 1. -1.9579 0.9728 ]
  [ 1. -1.8783 1. 1. -1.8984 0.9743 ]
  [ 1. -1.9932 1. 1. -1.9794 0.9902 ]
  [ 1. -1.8956 1. 1. -1.9074 0.9931 ]
  [ 1. -1.9921 1. 1. -1.988 0.9977 ] ]
```

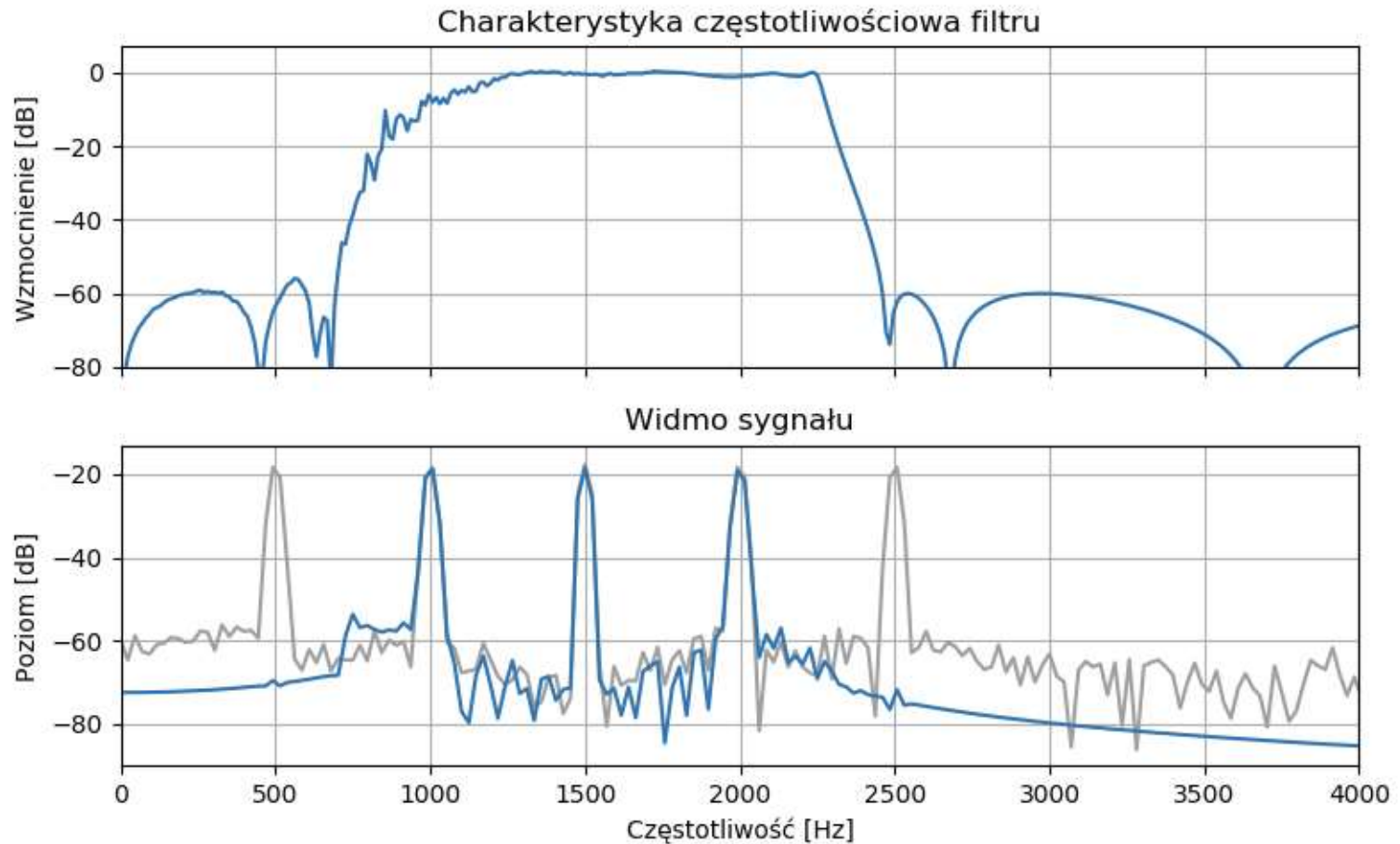
Otrzymaliśmy 7 sekcji drugiego rzędu,  
a więc filtr ma rząd 14.

Wzmocnienie jest zawarte w liczniku pierwszej sekcji  
( $k = 0,0006$ ).

Program uwzględnia  $a_0 = 1$  (czwarta kolumna).

# Projekt 2 – wynik filtracji sygnału x:

```
y = sig.sosfilt(sos, x)
```



Inne metody projektowania filtrów IIR:

## Metoda niezmienności odpowiedzi impulsowej

- projektujemy charakterystykę widmową filtru w dziedzinie zmiennej ciągłej  $s$ ,
- obliczamy ciągłą odpowiedź impulsową za pomocą odwrotnego przekształcenia Laplace'a,
- próbkujemy ciągłą odpowiedź impulsową,
- obliczamy zera i bieguny, stąd obliczamy współczynniki filtru.



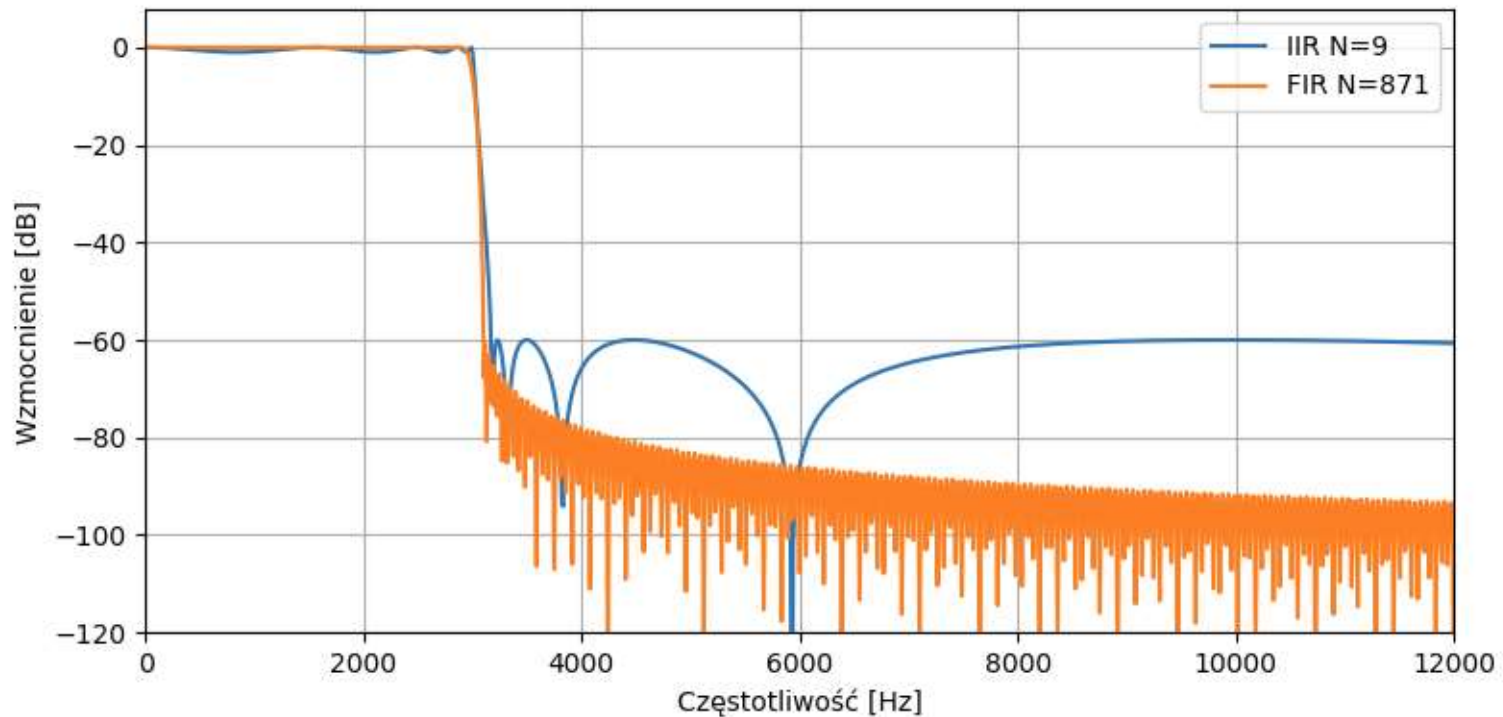
- Metoda niezmienności odpowiedzi impulsowej opiera się na założeniu, że odpowiedź impulsowa filtru analogowego i cyfrowego w punktach próbkowania jest identyczna.
- Metoda czasami stosowana zamiast przekształcenia dwuliniowego.
- Można obliczyć dowolne charakterystyki widmowe.
- Bardziej skomplikowane obliczenia.
- Metoda podatna na błędy związane z aliasingiem.

## Metoda Yule-Walkera

- Stosowana do systemów autoregresyjnych (AR), do których należą też filtry IIR.
- Współczynniki filtru są obliczane za pomocą algebry liniowej (operacji na macierzach).
- Dopasowanie metodą najmniejszych kwadratów.
- Umożliwia zaprojektowanie filtru o dowolnym kształcie charakterystyki.
- Skomplikowane obliczenia, dość rzadko stosowana do projektowania filtrów (Matlab: funkcja *yulewalk*).

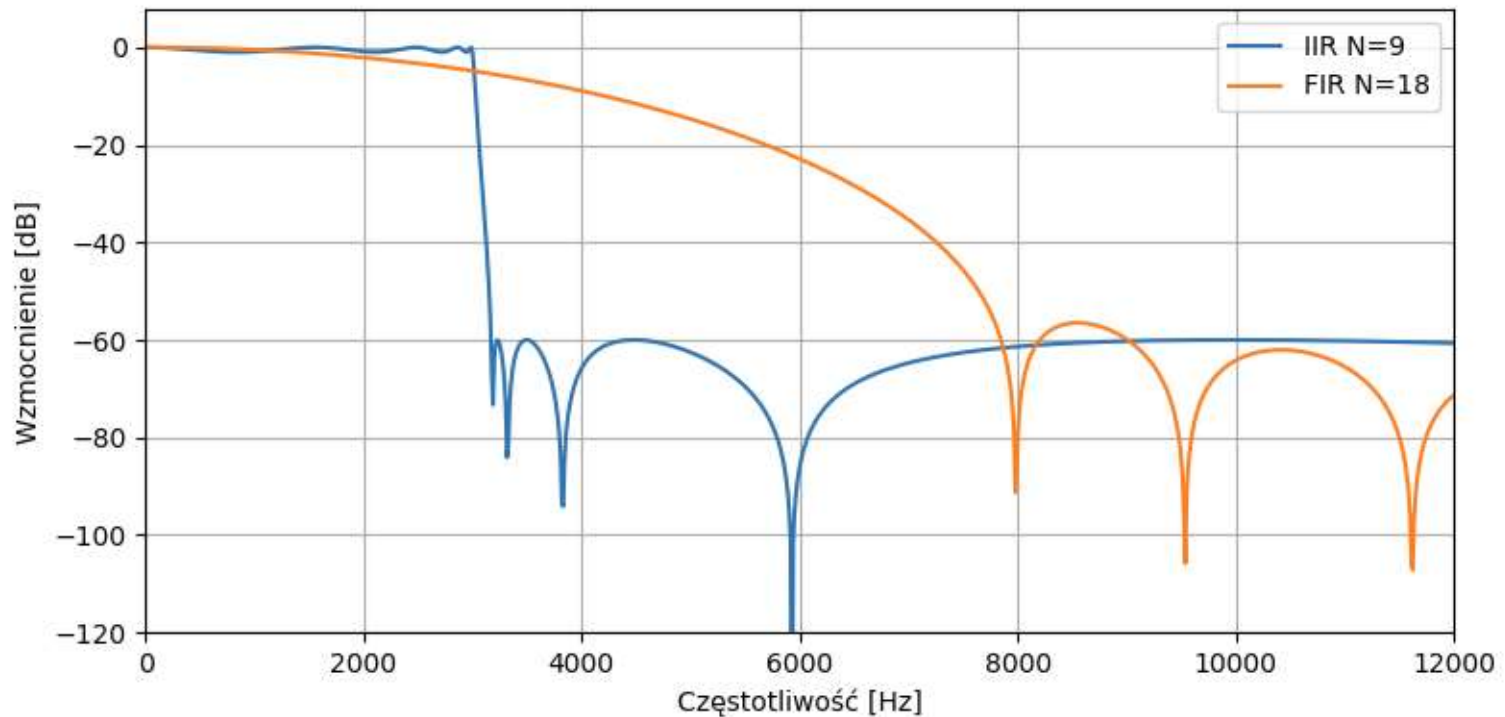
## Porównanie filtrów FIR i IIR – te same założenia projektowe (DP 3-3,2 kHz, Rz = 60 dB).

- FIR: rząd 871 (872 mnożenia)
- IIR: rząd 9 (19 mnożeń)



# Porównanie filtrów FIR i IIR – taka sama liczba mnożeń

FIR: rząd 18, IIR: rząd 9



- Filtry IIR są w stanie uzyskać taki sam efekt filtracji wykorzystując znacznie mniej operacji matematycznych niż filtry FIR (czyli są szybsze).
- Intuicyjne uzasadnienie:
  - Dla każdej nowej próbki sygnału, filtr FIR wykonuje filtrację „od zera”. Potrzebuje wielu próbek wejściowych, aby obliczyć poprawny wynik filtracji.
  - Filtr IIR potrzebuje tylko kilku próbek wejściowych i kilka poprzednich wyników filtracji. Następnie „poprawia” poprzednie wyniki, uwzględniając nową próbkę wejściową.

## Podsumowanie – zalety filtrów IIR

- Pozwalają uzyskać żądany efekt niskim nakładem obliczeniowym (filtry niskiego rzędu).
- Nie wymagają dużo miejsca w pamięci.
- Nie wprowadzają dużych opóźnień związanych z przetwarzaniem.
- Są proste w implementacji.
- Są cyfrowymi odpowiednikami filtrów analogowych. Można wykorzystać znane metody projektowania i analizy filtrów.

## Podsumowanie – wady filtrów IIR

- Mają nieliniową charakterystykę fazową. Opóźnienie zależy od częstotliwości. Filtry wprowadzają zniekształcenia fazowe – różne składowe widma mają różne opóźnienia, faza się „rozjeżdża”.
- Są podatne na błędy numeryczne związane z kwantyzacją współczynników. Mogą one spowodować utratę stabilności filtru. Problem można zmniejszyć stosując strukturę SOS.
- Projektowanie jest trudniejsze niż dla FIR.