

Zastosowania Procesorów Sygnałowych

dr inż. Grzegorz Szwoch

greg@multimed.org

p. 732 - Katedra Systemów Multimedialnych

Zaawansowane algorytmy DSP

Wstęp

Cztery algorytmy wybrane spośród bardziej zaawansowanych metod przetwarzania sygnałów, przedstawione w praktycznych zastosowaniach:

- zmiana częstotliwości próbkowania, decymacja i interpolacja sygnału
- filtry adaptacyjne
- autokorelacja
- sygnał analityczny i transformator Hilberta

Zmiana częstotliwości próbkowania

PROBLEM #1: częstotliwość próbkowania 48 kHz jest za mała dla naszych potrzeb:

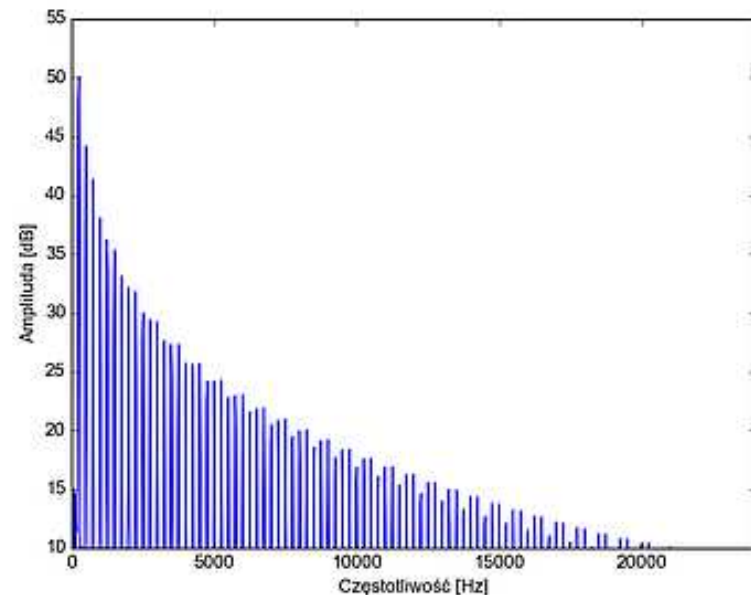
- występuje aliasing (np. przy generowaniu sygnałów harmonicznym),
- rozdzielczość czasowa jest za mała:
 - filtracja – zbyt szerokie pasmo przejściowe,
 - generowanie sygnałów z tablicy – za mało próbek sygnału,
 - konieczność stosowania interpolacji.

Nadpróbkowanie

- **Nadpróbkowanie** (*oversampling*)
 - przetwarzanie sygnału z większą częstotliwością próbkowania niż docelowa.
- W technice audio często stosuje się 4-krotne nadpróbkowanie, czyli $f_s = 192 \text{ kHz}$.
- Wada: więcej obliczeń, przy 4-krotnym nadpróbkowaniu DSP musi wykonać 4 razy więcej operacji w tym samym czasie.

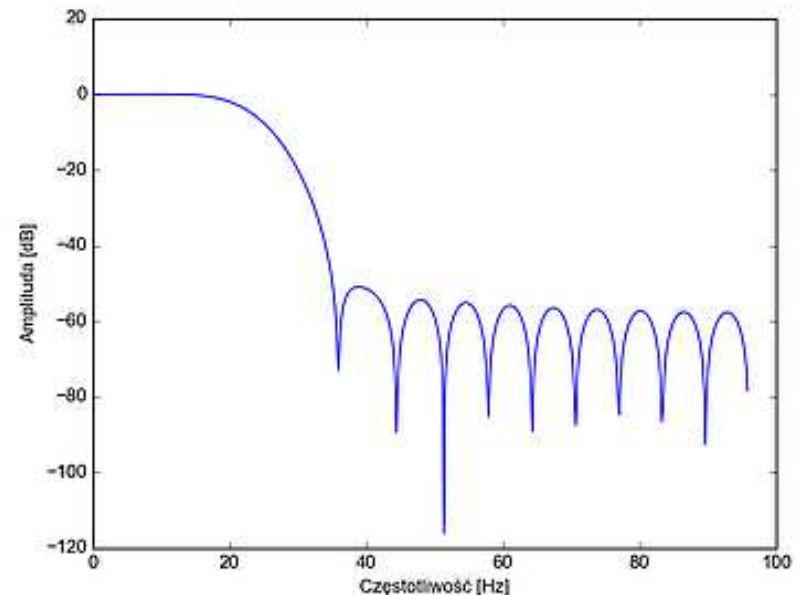
Decymacja

- Na wyjściu potrzebujemy sygnał z $f_s = 48$ kHz.
- Założmy $f_s = 192$ kHz. Mamy 4 razy za dużo próbek.
- Weźmy tylko co czwartą próbkę. Mamy tyle próbek, ile potrzebujemy.
- Widmo: nie o to chodziło...
Zakres (0 – 96 kHz)
został „ściśnięty”
do (0 – 24 kHz).



Decymacja

- Zanim pominiemy próbki, musimy zastosować filtr dolnoprzepustowy o częstotliwości granicznej równej docelowej cz. Nyquista.
- W naszym przypadku: $f_g = 24$ kHz.
- **Filtr decymacyjny** obcina pasmo do docelowej częstotliwości Nyquista.

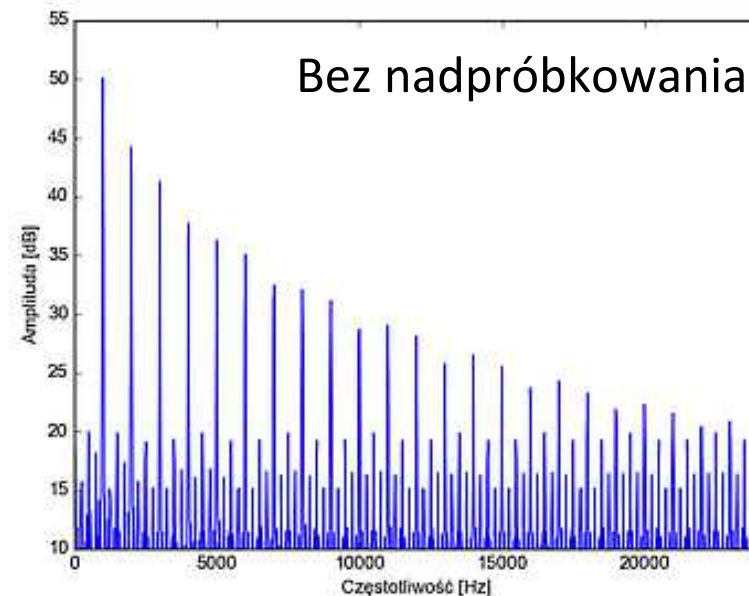
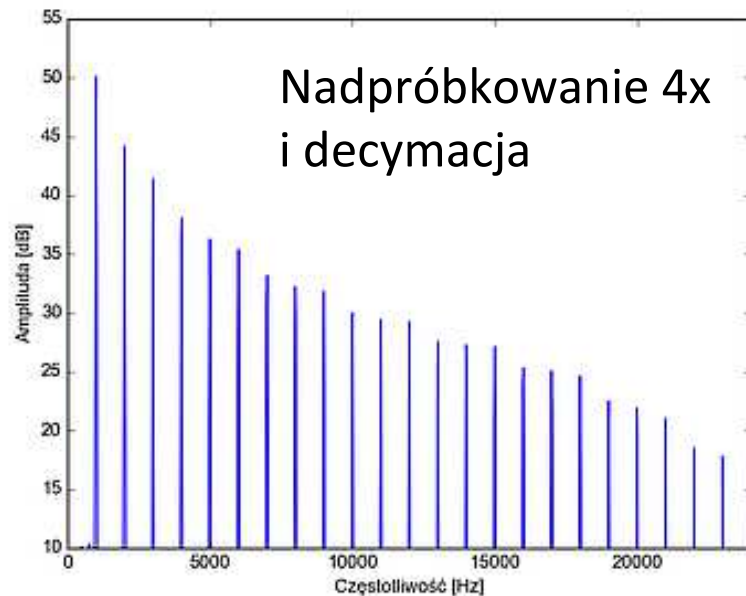


Decymacja

Decymacja z $fs1$ do $fs2 = fs1 / D$:

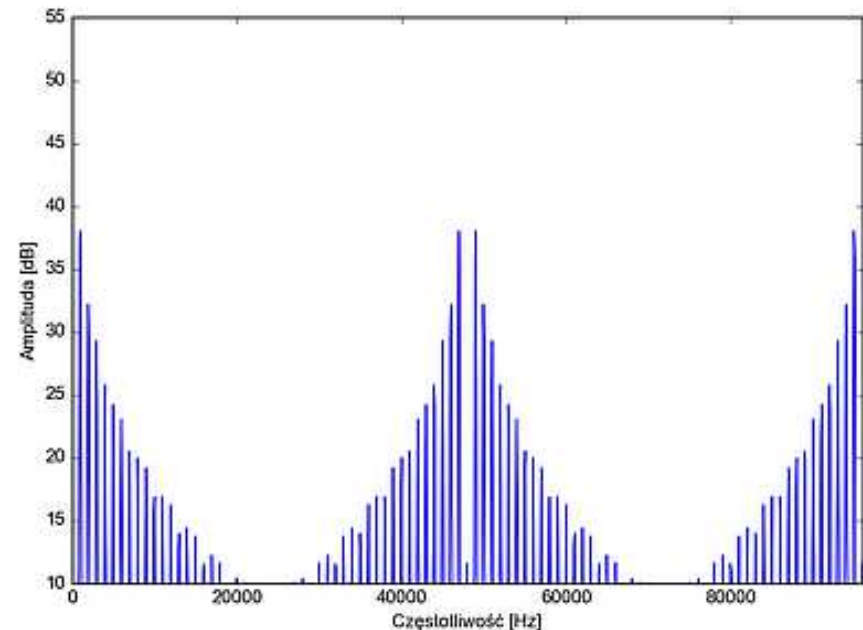
- najpierw filtr dolnoprzepustowy o $f_g = fs2 / 2$,
- następnie wzięcie co D próbki.

Przykład: sygnał piłokształtny $f = 1$ kHz:



Interpolacja

- Mamy sygnał z $f_s = 48$ kHz. Jak go nadpróbekować do 192 kHz?
- Wstawmy 3 zera pomiędzy każdą parę próbek.
- Mamy tyle próbek, ile potrzeba.
- Widmo: prawie dobrze...

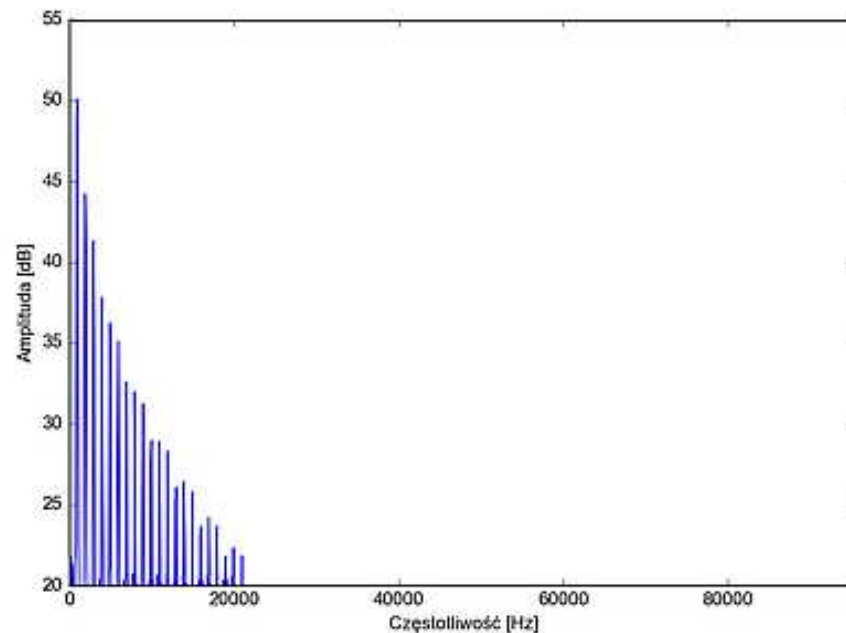


Interpolacja

Interpolacja z $fs1$ do $fs2 = fs1 * L$:

- najpierw wstawienie $(L-1)$ próbek „pomiędzy”,
- następnie filtr dolnoprzepustowy o $fg = fs1 / 2$.

Przykład: sygnał piłokształtny $f = 1$ kHz:



Przepróbkowanie

- Problem #1A: mamy sygnał o $fs1 = 44\ 100$ Hz, a potrzebujemy $fs2 = 48\ 000$ Hz.
- **Przepróbkowanie** (*resampling*)
 - zmiana częstotliwości próbkowania, np.:
 - wstawienie $(L-1)$ zer między próbki,
 - filtr dolnoprzepustowy – tylko raz!
 - wzięcie co D próbkę.
- $L / D = 48000 / 44100 = 160 / 147$
a więc interpolacja $L = 160$ i decymacja $D = 147$

Funkcje z DSPLIB

- Filtr decymacyjny:

firdec

Decimating FIR Filter

Function

ushort oflag = firdec (DATA *x, DATA *h, DATA *r, DATA *dbuffer , ushort nh, ushort nx, ushort D)

- Filtr interpolacyjny:

firinterp

Interpolating FIR Filter

Function

ushort oflag = firinterp (DATA *x, DATA *h, DATA *r, DATA *dbuffer , ushort nh, ushort nx, ushort I)

Usuwanie zakłóceń

PROBLEM #2

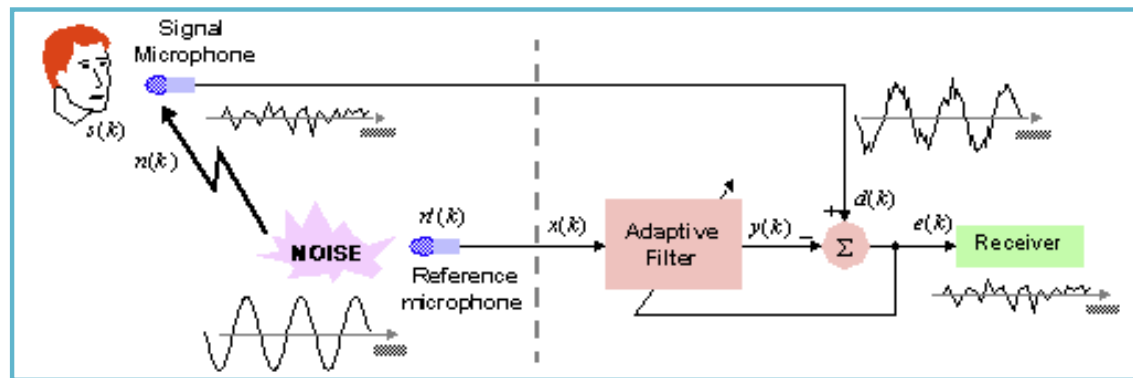
- Zestaw głośnomówiący w samochodzie.
- Mikrofon zbiera mowę oraz hałas z kabiny.
- Gdyby zakłócenia były stacjonarne, moglibyśmy zaprojektować filtr do usuwania zakłóceń.
- Zakłócenia są zmienne, więc to nie zadziała.
- Potrzebny jest filtr, który będzie **dostosowywał** swoją charakterystykę do zmiennego charakteru zakłóceń.

Filtry adaptacyjne

- **Filtry adaptacyjne** to takie filtry, których współczynniki są modyfikowane przez algorytm.
- **Sygnał referencyjny**: z głównego mikrofonu.
- **Sygnał filtrowany**: z drugiego mikrofonu, który zbiera tylko hałas, nie mowę.
- **Sygnał błędu**: różnica między wynikiem filtracji a sygnałem referencyjnym.
- **Adaptacja współczynników** filtru w taki sposób, aby zminimalizować błąd filtracji.

Filtry adaptacyjne

- Ilustracja działania:



<http://www.iro.umontreal.ca/~mignotte/IFT3205/APPLETS/AdaptiveSignalProcessing/AdaptiveSignalProcessing.html>

- Tutaj chcemy uzyskać taki sygnał błędu, aby jak najbardziej przypominał on mowę bez szumu.
- W typowych przypadkach: sygnał błędu powinien być bliski zeru.

Algorytm LMS

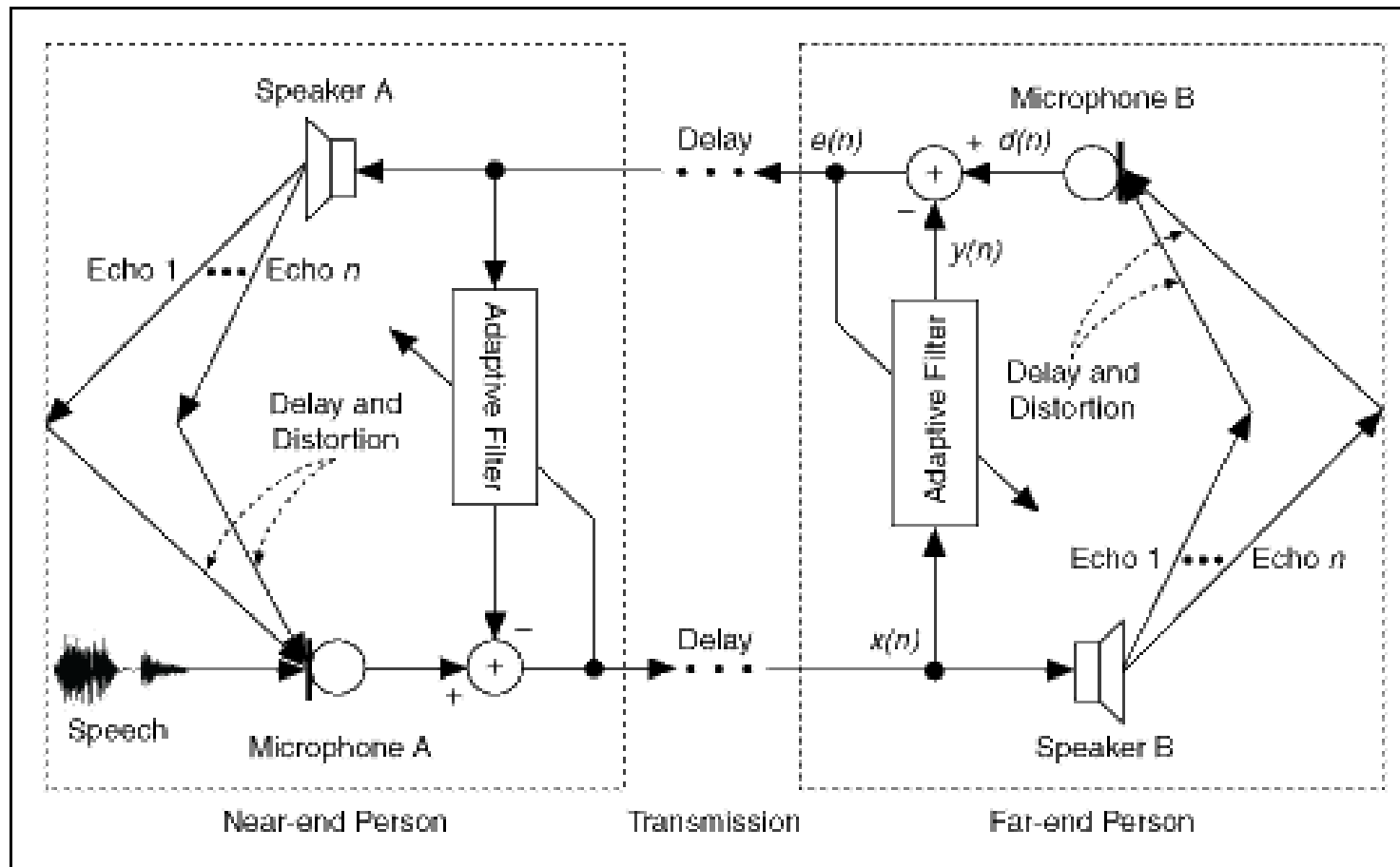
- LMS – *Least Mean Squares* – metoda najmniejszych kwadratów.
- Założenie: średnia kwadratów wartości sygnału błędu powinna być jak najmniejsza.
- Im większa jest wartość błędu, tym bardziej należy zmodyfikować współczynniki filtru.
- Adaptacja współczynników:

$$w_k(n+1) = w_k(n) + \mu x(n-k) e(n)$$

(μ - krok adaptacji – szybkość zmian)

Filtry adaptacyjne

Inny przykład: usuwanie echa telekomunikacyjnego



Filtry adaptacyjne w DSPLIB

dlms

Adaptive Delayed LMS Filter

Function

ushort oflag = dlms (DATA *x, DATA *h, DATA *r, DATA *des, DATA *dbuffer,
DATA step, ushort nh, ushort nx)

- x – sygnał wejściowy
- h – współczynniki filtru
- r – bufor na sygnał po filtracji
- des – sygnał referencyjny
- $step$ – krok adaptacji

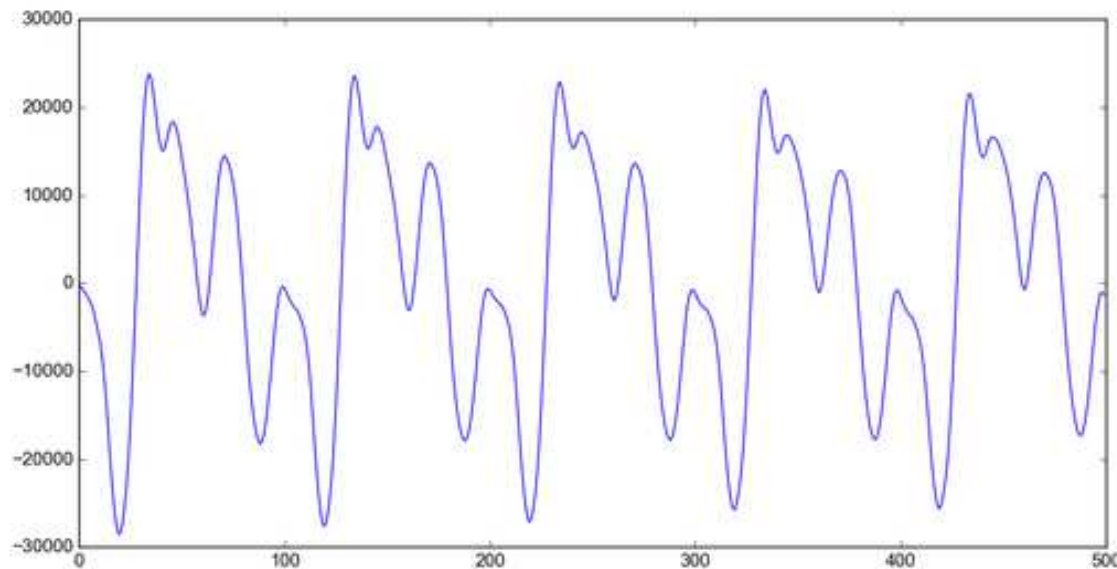
Wyznaczanie wysokości dźwięku

PROBLEM #3:

- Mamy nagrany dźwięk instrumentu muzycznego, np. trąbki. Chcemy znać jego wysokość.
- Sygnały muzyczne są pseudo-okresowe i harmoniczne.
- **Pseudo-okres** – najkrótszy powtarzalny fragment.
- **Częstotliwość podstawowa** – odwrotność pseudo-okresu, wyznacza **wysokość** dźwięku na skali muzycznej, np. a^1 .

Wyznaczanie wysokości dźwięku

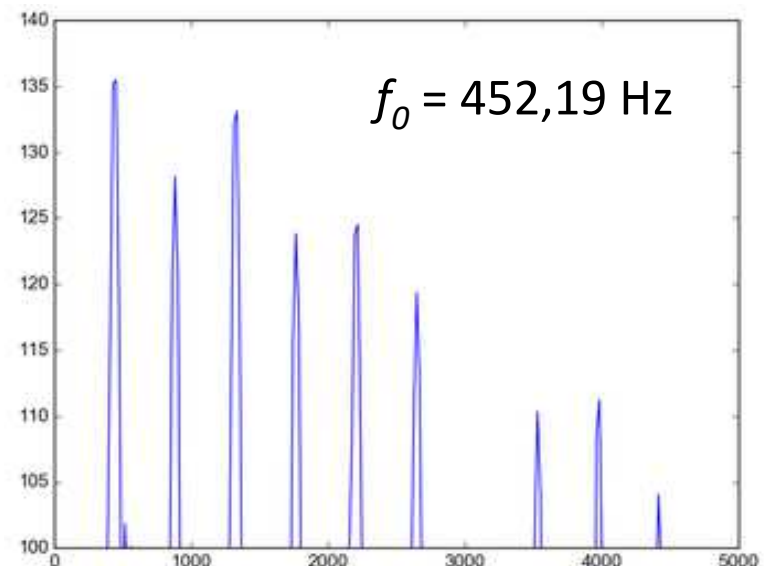
- Fragment sygnału dźwięku trąbki:



- Jeśli znajdziemy długość pseudo-okresu, wyznaczymy też wysokość dźwięku.

Wysokość dźwięku przez FFT

- Podejście intuicyjne: liczymy FFT i szukamy pierwszego prążka.
- Metoda bardzo zawodna:
 - często znalezienie pierwszego prążka jest trudne (nie musi on być dominujący),
 - mała dokładność – zbyt mała rozdzielczość FFT

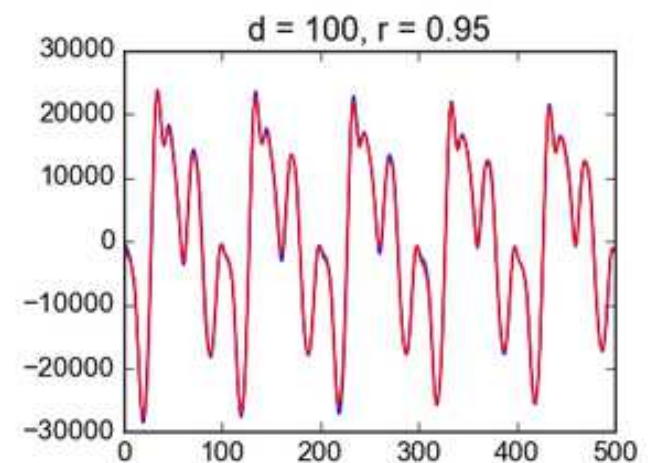
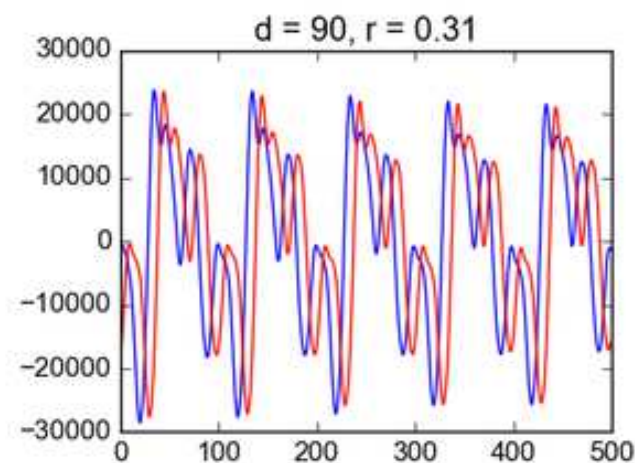
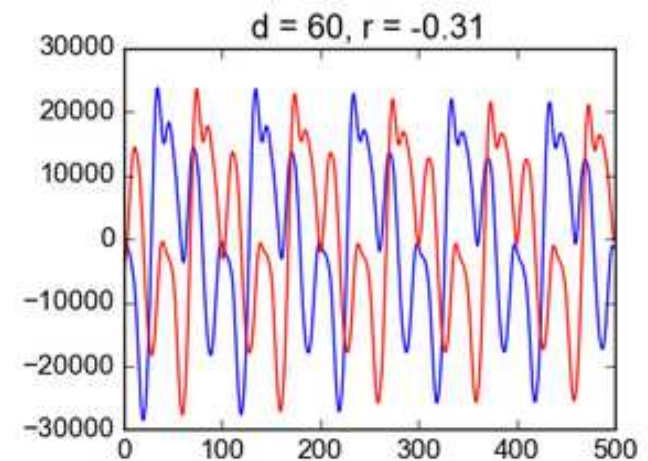
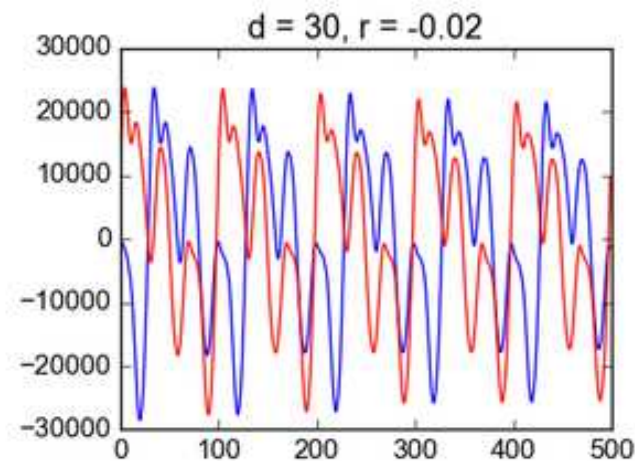


Autokorelacja

- **Współczynnik autokorelacji:** miara podobieństwa sygnału do jego kopii przesuniętej w czasie. Wartości od -1 do 1.
- **Funkcja autokorelacji:** wartości współczynnika dla różnych przesunięć.
- Obliczamy współczynnik przesuwając sygnał kolejno o coraz większą liczbę próbek.
- Maksimum funkcji oznacza największe podobieństwo sygnałów dla danego przesunięcia.

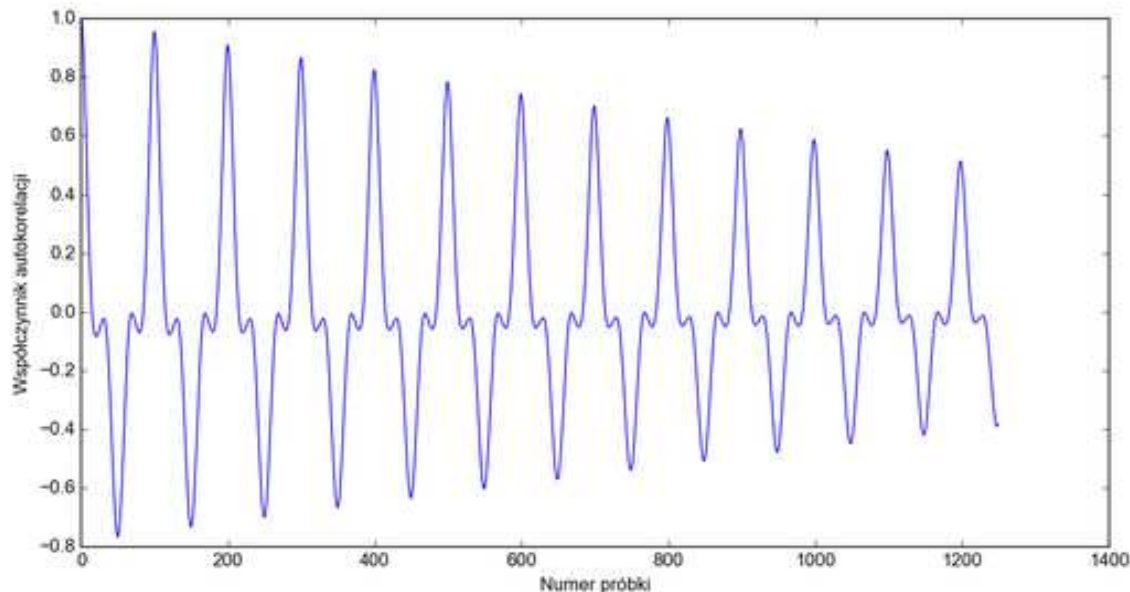
Autokorelacja

Współczynnik autokorelacji r dla różnych przesunięć d



Wysokość metodą autokorelacji

- Jeżeli przesunięcie będzie równe pseudo-okresowi, wartość współczynnika będzie największa!
- Maksimum dla $d = 100$, czyli:
 $f_0 = 44100 / 100 = 441 \text{ Hz}$



Autokorelacja

Inny przykład: wykrywanie obecności mowy w zaszumionym sygnale.

- Sygnał mowy jest pseudo-okresowy: w funkcji autokorelacji będą obecne maksima.
- Szum jest nieskorelowany – brak maksimów.
- **Bramkowanie sygnału**: sprawdzenie ile wartości funkcji autokorelacji dla sygnału w buforze przekracza zadany próg.

Autokorelacja w DSPLIB

acorr

Autocorrelation

Function

ushort oflag = acorr (DATA *x, DATA *r, ushort nx, ushort nr, type)

Parametr *type* określa sposób obliczania autokor.:

- bez normalizacji (*raw*),
- obciążona (*biased*) – dzielona przez l. próbek
- nieobciążona (*unbiased*)

Analiza złożonego sygnału

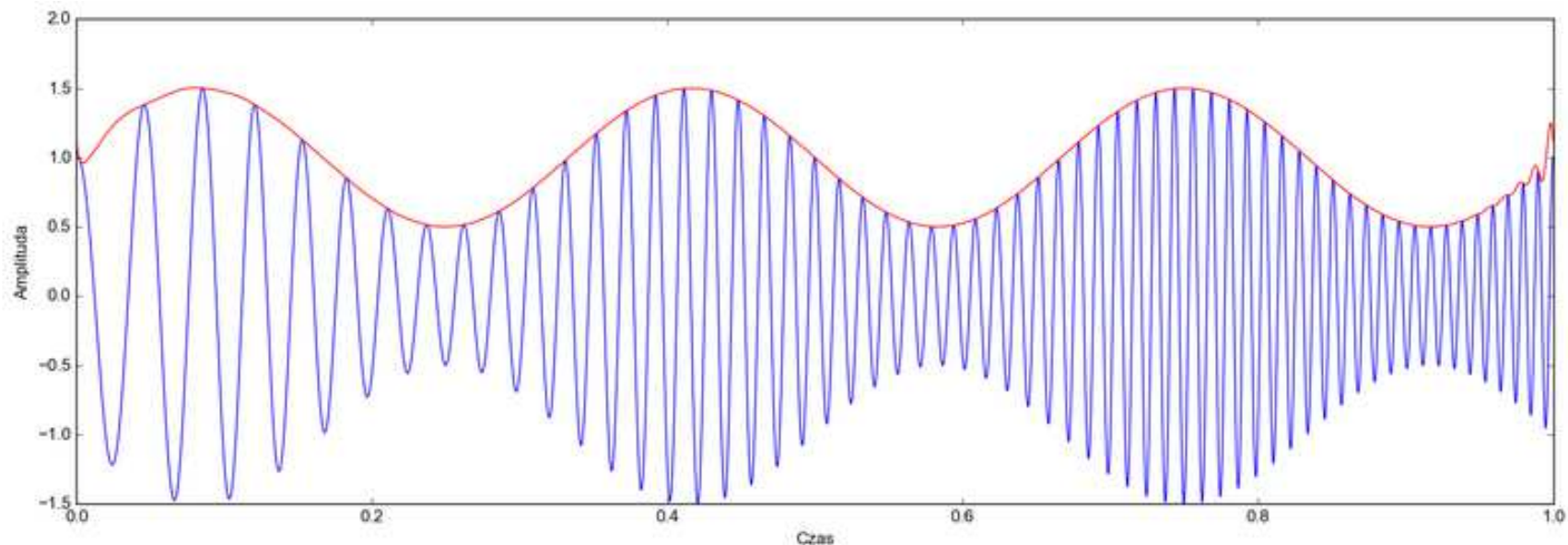
PROBLEM #4

- Mamy następujący złożony sygnał:
 - sygnał świergotowy (*chirp*) o liniowo zwiększającej się częstotliwości od 20 do 100 Hz,
 - zmodulowany amplitudowo – amplituda sygnału przemnożona przez sygnał sinus $f = 3$ Hz.
- Dla dowolnej chwili chcemy znaleźć:
 - amplitudę obwiedni sygnału,
 - aktualną wartość częstotliwości.

Analiza złożonego sygnału

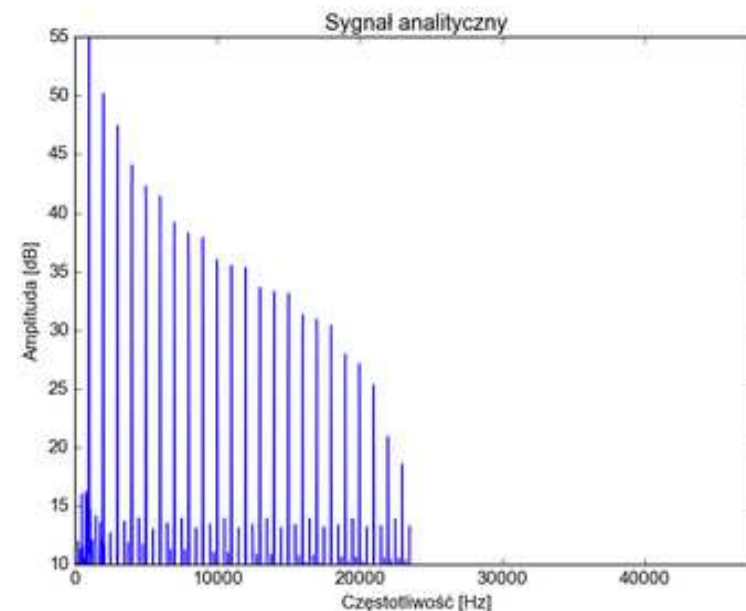
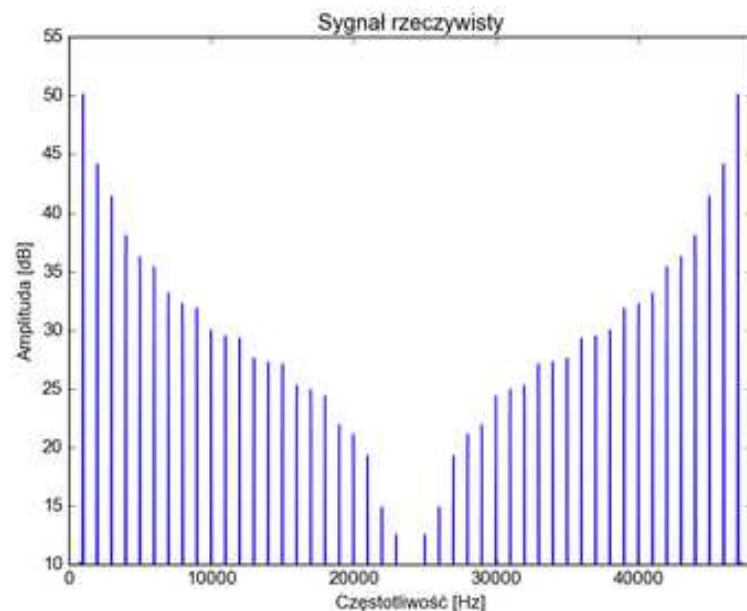
Tak wygląda nasz sygnał:

- kolor niebieski – wartości próbek sygnału,
- kolor czerwony – obwiednia amplitudy.



Sygnal analityczny

- Sygnal rzeczywisty: dwie kopie widma od 0 do f_s .
- Wyzerujemy „drugą” kopię widma.
- Powstanie sygnal zespolony, nazywany **sygnałem analitycznym**.

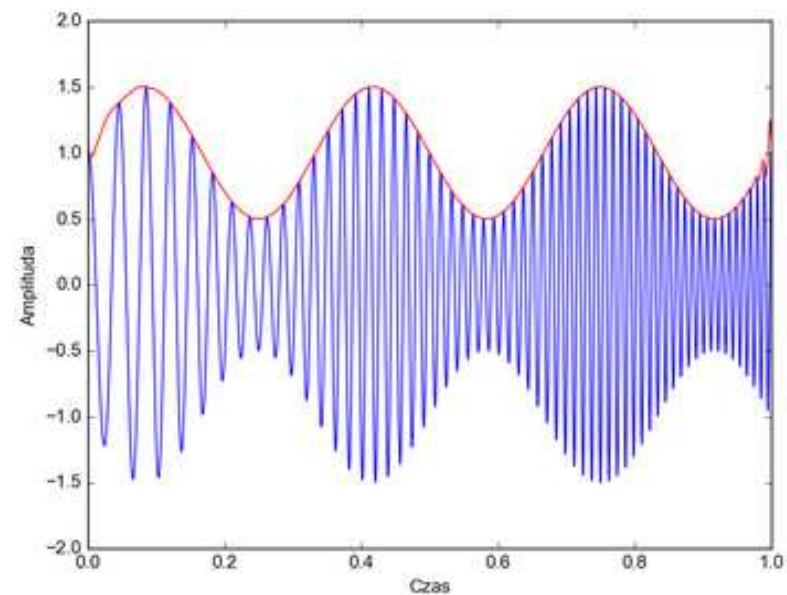
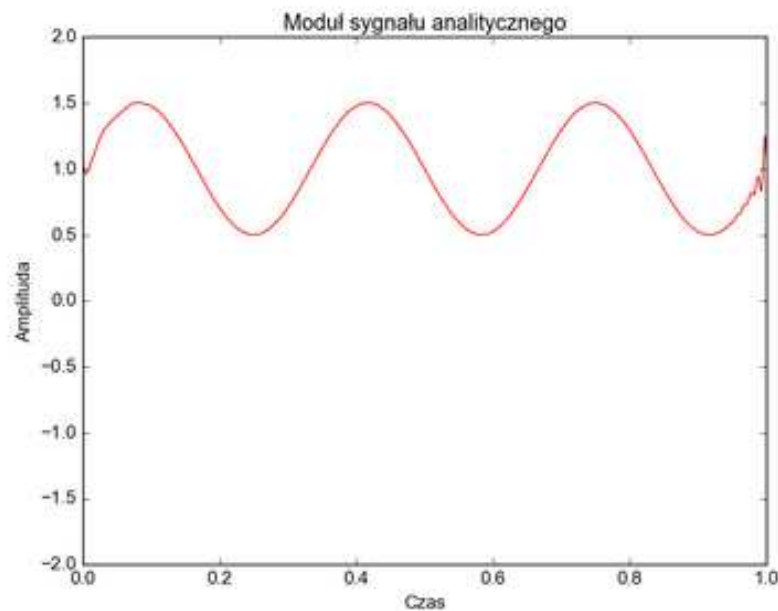


Obwiednia sygnału

Co nam to dało:

- moduł sygnału analitycznego $r(n)$:

$$y(n) = \sqrt{\operatorname{Re}(r(n))^2 + \operatorname{Im}(r(n))^2}$$



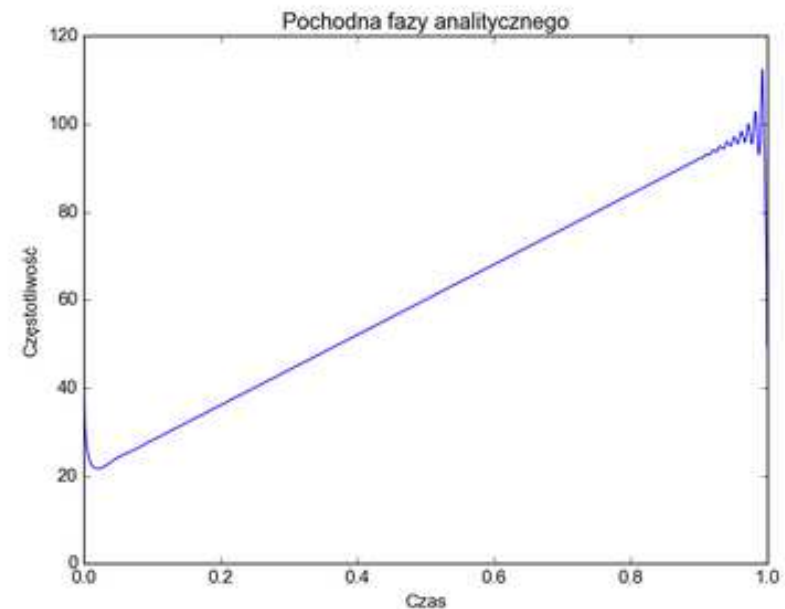
Częstotliwość chwilowa

- faza chwilowa sygnału analitycznego:

$$\varphi(n) = \arctan\left(\frac{\text{Im}(r(n))}{\text{Re}(r(n))}\right)$$

- częstotliwość chwilowa
= pochodna fazy:

$$f(n) = \frac{f_s}{2\pi} (\varphi(n) - \varphi(n-1))$$



Transformator Hilberta w DSPLIB

- Algorytm przetwarzający sygnał rzeczywisty w analityczny nazywa się transformatorem Hilberta.
- Może to być filtr FIR lub można wykonywać operacje na widmie (FFT).
- W bibliotece DSPLIB:

hilb16

FIR Hilbert Transformer

Function

ushort oflag = hilb16 (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nx, ushort nh)

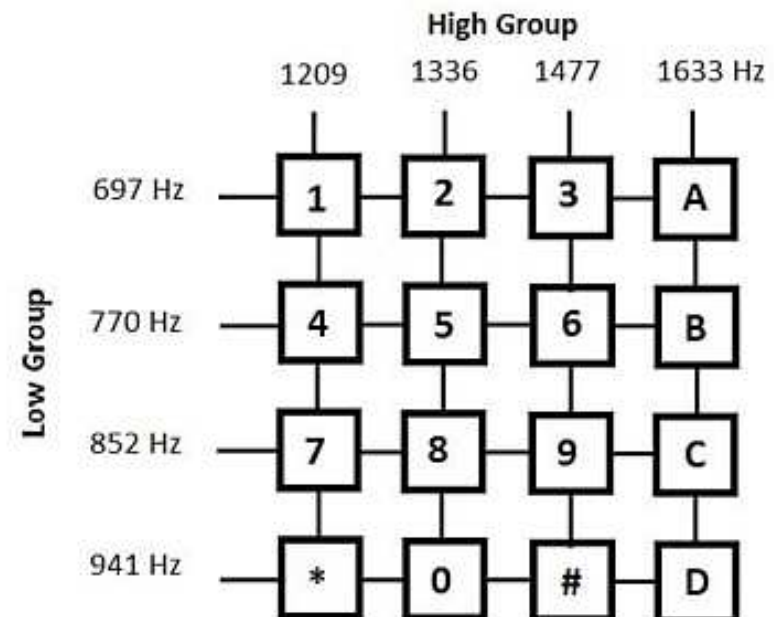
Podsumowanie

Co musimy wiedzieć:

- jak zwiększać lub zmniejszać częstotliwość próbkowania (interpolacja, decymacja),
- jak przepróbkować sygnał,
- co to są filtry adaptacyjne, jak działają, zastosowania,
- algorytm LMS w adaptacji współczynników,
- zastosowania funkcji autokorelacji,
- co to jest sygnał analityczny i do czego się przydaje.

Bonus - DTMF

- DTMF – *Dual-Tone Multi Frequency*
- Metoda kodowania cyfr w telekomunikacji.
- Każda cyfra jest reprezentowana przez dwuton - dwa sinusy, 2 spośród 8 częstotliwości.
- Np. cyfra 6:
770 Hz, 1477 Hz
- Zastosowanie
– np. wybieranie tonowe w telekomunikacji.



Bonus - DTMF

„Zadanie domowe” – do samodzielnego zastanowienia się. Jak zrobić detektor DTMF?

- Detekcja początku i końca dwutonu
 - jak wykryć cyfrę, ale jej nie powtarzać
- Detekcja częstotliwości obu tonów:
 - filtry? FIR czy IIR?
 - FFT?
 - może inna metoda?