INNE ALGORYTMY PRZETWARZANIA SYGNAŁÓW

w zastosowaniach telekomunikacyjnych

Opracowanie: Grzegorz Szwoch

Politechnika Gdańska, Katedra Systemów Multimedialnych

Wstęp

Znamy podstawowe algorytmy przetwarzania sygnałów stosowane na procesorach sygnałowych: przekształcenie Fouriera (FFT) oraz filtry FIR i IIR.

Na tym wykładzie przedstawimy inne algorytmy, wybrane spośród bardziej zaawansowanych metod przetwarzania sygnałów stosowanych w telekomunikacji, przedstawione w praktycznych zastosowaniach:

- zmiana częstotliwości próbkowania, decymacja i interpolacja sygnału,
- usuwanie zmiennych zakłóceń – filtry adaptacyjne,
- detekcja częstotliwości – autokorelacja,
- pomiar odpowiedzi impulsowej – korelacja,
- demodulacja sygnału – sygnał analityczny i transformator Hilberta.

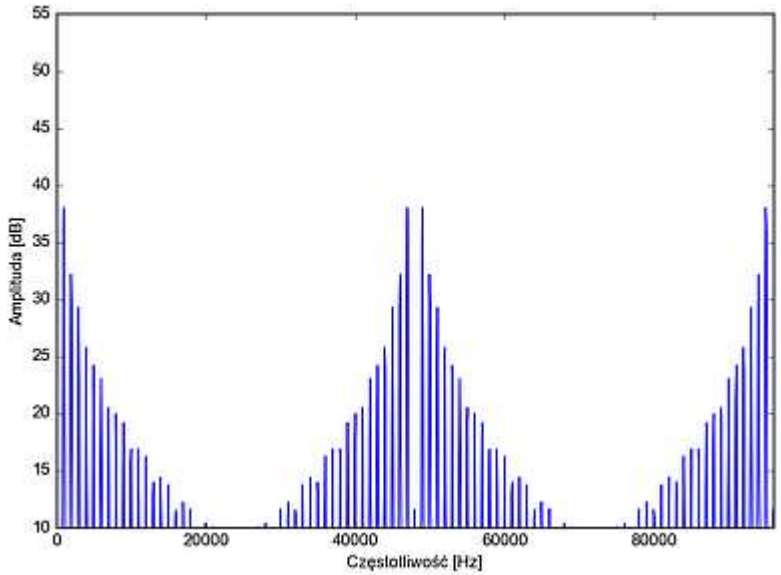
Zmiana częstotliwości próbkowania

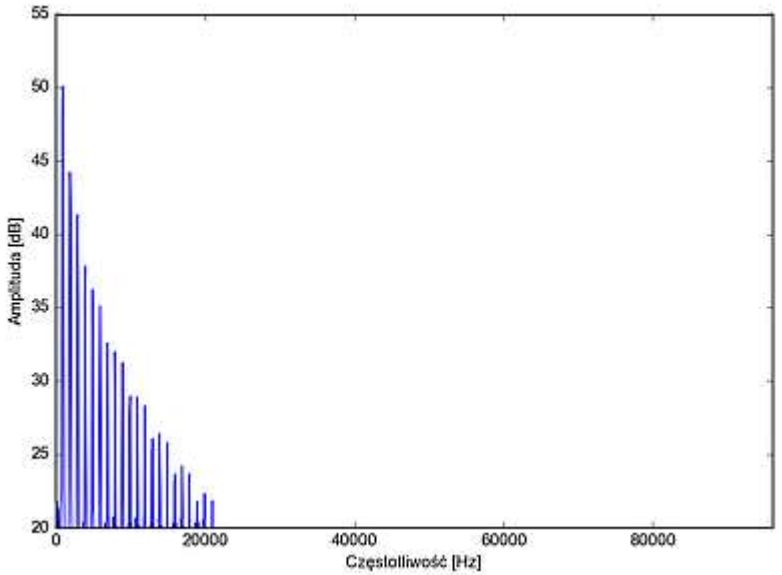
PROBLEM #1: sygnał wejściowy ma inną częstotliwość próbkowania niż potrzebujemy.

- Nadpróbkowanie (*oversampling*) – przetwarzanie sygnału z większą częstotliwością próbkowania niż docelowa.
- W technice audio często stosuje się 4-krotne nadpróbkowanie, czyli fs = 192 kHz.
- Wada: więcej obliczeń, przy 4-krotnym nadpróbkowaniu DSP musi wykonać 4 razy więcej operacji w tym samym czasie.
- Zaleta: większa dokładność i większa odporność na aliasing.
- Interpolacja – zwiększenie częstotliwości próbkowania *L* razy.
- Decymacja – zmniejszenie częstotliwości próbkowania *D* razy.
- Częsty schemat: interpolacja, przetwarzanie z wyższą częstotliwością próbkowania, decymacja (powrót do początkowej częstotliwości próbkowania).

Interpolacja

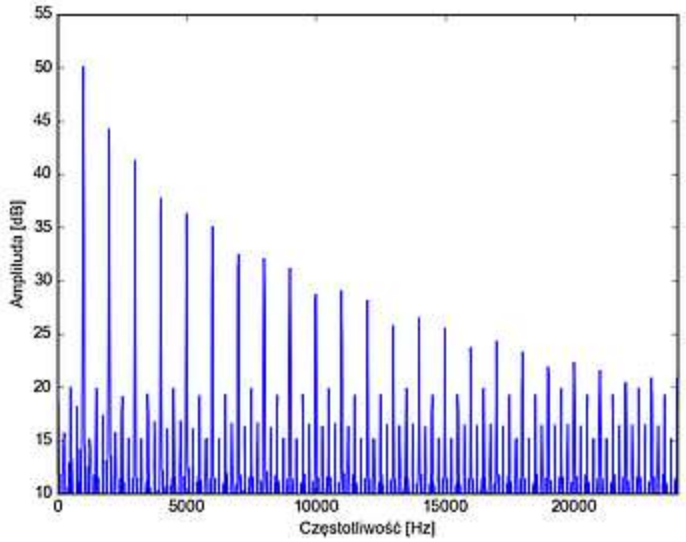
- Mamy sygnał z *fs* = 48 kHz. Jak go nadpróbkować do 192 kHz?
- Wstawiamy 3 zera pomiędzy każdą parę próbek. Mamy tyle próbek, ile potrzeba.
- Widmo: prawie dobrze, ale mamy „niepotrzebne” kopie widma.
- Musimy zastosować na wyjściu filtr dolnoprzepustowy o częstotliwości granicznej równej oryginalnej częstotliwości Nyquista (tutaj: 24 kHz).

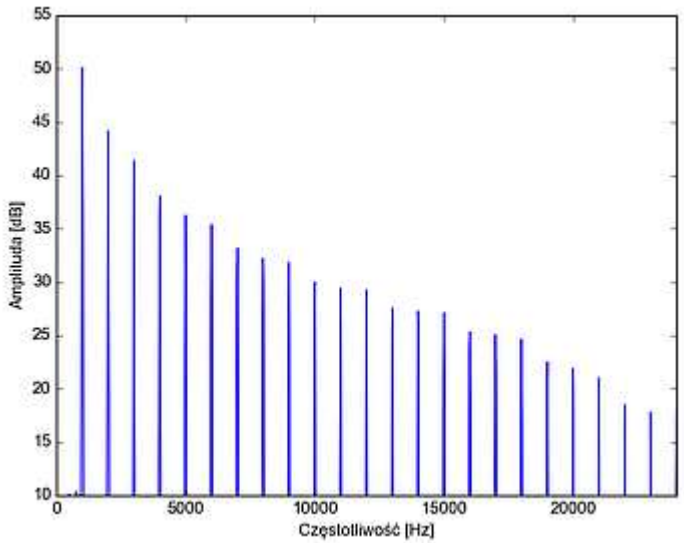




Decymacja

- Na wyjściu potrzebujemy sygnał z *fs* = 48 kHz.
- Załóżmy *fs* = 192 kHz. Mamy 4 razy za dużo próbek.
- Weźmy tylko co czwartą próbkę. Mamy tyle próbek, ile potrzebujemy, ale wystąpił aliasing widma.
- Zanim pominiemy próbki, musimy zastosować filtr dolnoprzepustowy o częstotliwości granicznej równej docelowej częstotliwości Nyquista. W naszym przypadku: 24 kHz.





Przepróbkowanie

- Problem: mamy sygnał o *fs1* = 44 100 Hz, a potrzebujemy *fs2* = 48 000 Hz.
- Przepróbkowanie (*resampling*) – zmiana częstotliwości próbkowania (*L*/*D*) razy:
  - wstawienie (*L*–1) zer między każdą parę próbek,
  - filtr dolnoprzepustowy (wystarczy jeden, tutaj na 22 050 Hz),
  - wzięcie co *D* próbkę.
- *L* / *D* = 48000 / 44100 = 160 / 147, a więc interpolacja *L* = 160 i decymacja *D* = 147.
- W tym przypadku wymagana jest więc bardzo duża liczba operacji.
- W praktyce, konwersję 44,1 ↔ 48 kHz wykonuje się najczęściej za pomocą FFT.

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)

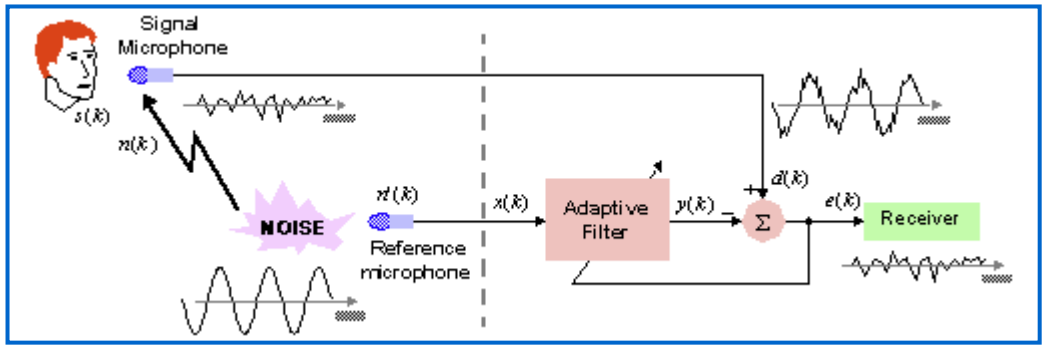
Usuwanie zakłóceń

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 adaptował swoją charakterystykę do zmiennego charakteru zakłóceń.

Filtry adaptacyjne

- Filtry adaptacyjne: 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.
- W naszym przypadku: filtr adaptacyjny przetwarza szum, próbuje dostosować go tak, aby był podobny do szumu obecnego w głównym mikrofonie, po czym odejmuje go od sygnału z mikrofonu, pozostawiając sygnał mowy.



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.
- Adaptacja współczynników:

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

- μ - krok adaptacji (*step*) – szybkość zmian współczynników.
- Funkcja DSPLIB do obliczania adaptacji LMS filtru:

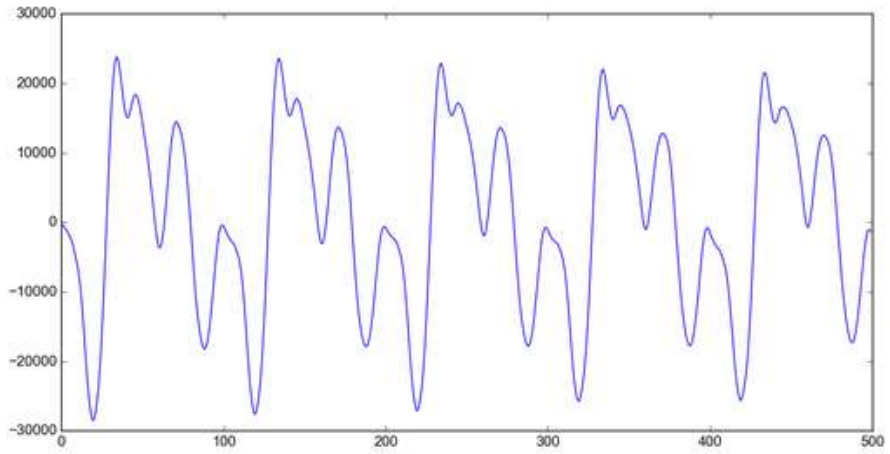
dlms *Adaptive Delayed LMS Filter*

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

Wyznaczanie wysokości dźwięku

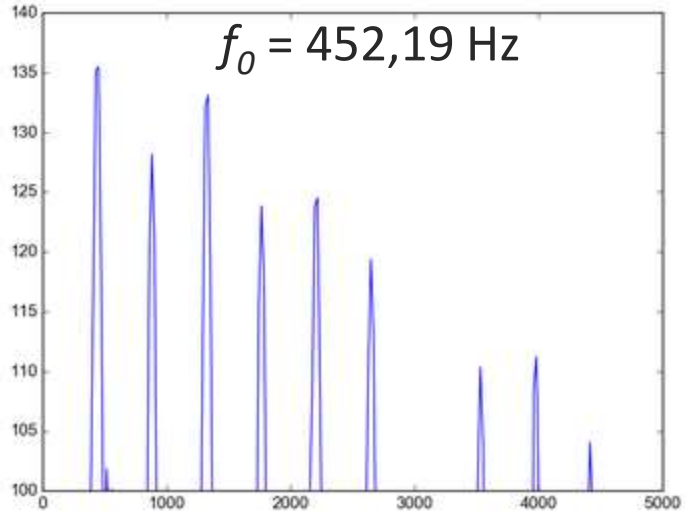
PROBLEM #3:

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



Wysokość dźwięku przez FFT

- Podejście intuicyjne: liczymy FFT i szukamy pierwszego maksimum.
- Metoda bardzo zawodna:
  - często znalezienie pierwszego maksimum jest trudne,
  - mała dokładność – zbyt mała rozdzielczość częstotliwościowa FFT,
  - możemy nieco dokładniej oszacować częstotliwość maksimum metodą paraboli opisaną na wcześniejszym wykładzie.



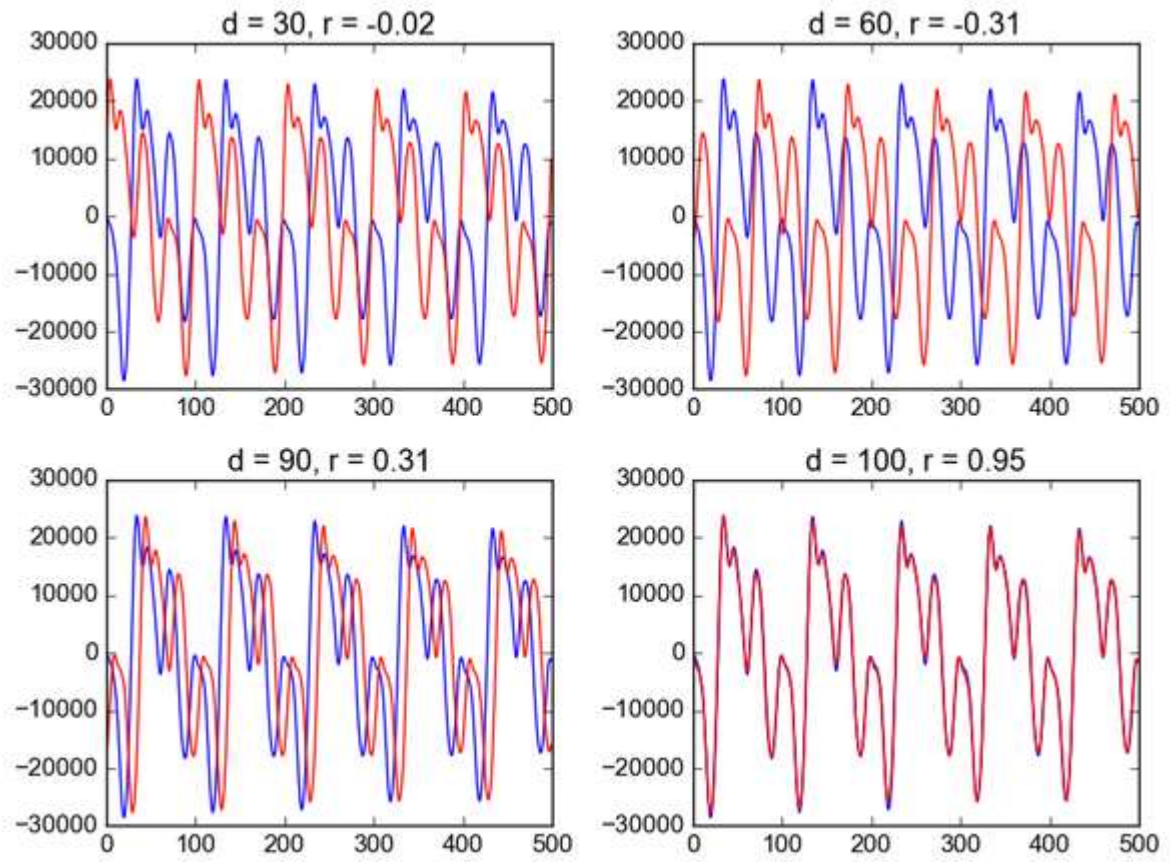
Autokorelacja

- Współczynnik autokorelacji (*autocorrelation*): miara podobieństwa sygnału do jego kopii przesuniętej w czasie.
- Wartości od -1 do 1. Wartość 0: brak korelacji.
- Funkcja autokorelacji: wartości współczynnika dla różnych przesunięć (*lag*).
- Obliczamy współczynnik przesuwając sygnał kolejno o coraz większą liczbę próbek.
- Maksimum funkcji autokorelacji oznacza największe podobieństwo sygnałów dla danego przesunięcia.
- Czyli: pierwsze maksimum funkcji korelacji (pomijając zerowe przesunięcie) wyznaczy nam pseudo-okres sygnału, a więc i częstotliwość dźwięku.

Autokorelacja

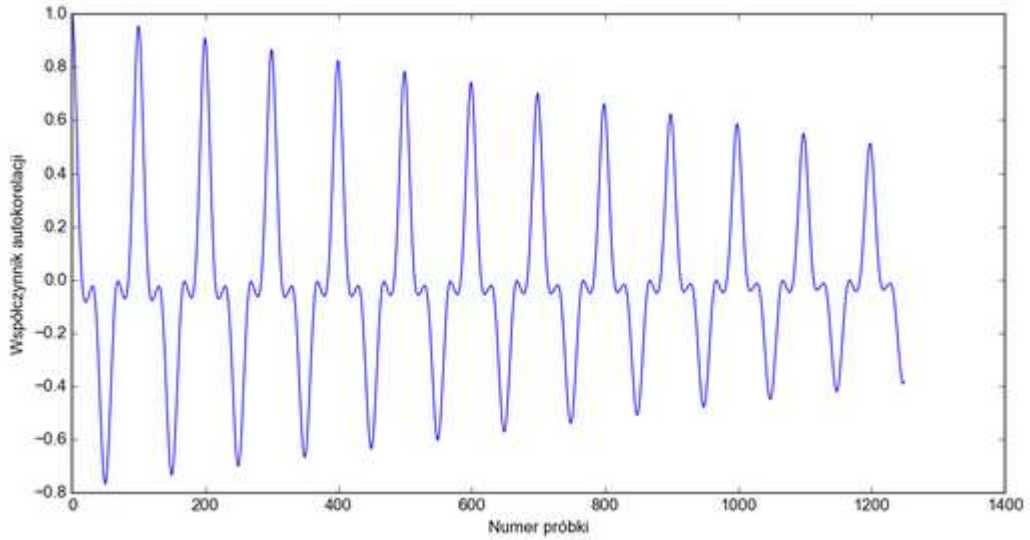
Współczynnik autokorelacji *r* dla różnych przesunięć *d.*

Maksimum dla *d* = 100, więc to jest długość okresu sygnału.



Wysokość metodą autokorelacji

- Wykres funkcji autokorelacji wykazuje maksima dla kolejnych wielokrotności pseudo-okresu.
- Pierwsze maksimum powyżej zera dla d = 100, czyli:
  f_0 = 44100 / 100 = 441 Hz
- Muzyk prawdopodobnie miał zagrać f_0 = 440 Hz (a^1).
- FFT dało nam f_0 = 452,19 Hz, więc znacznie mniej dokładnie niż dla autokorelacji.



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, decyzja: sygnał czy szum.

Obliczanie autokorelacji za pomocą DSPLIB:

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

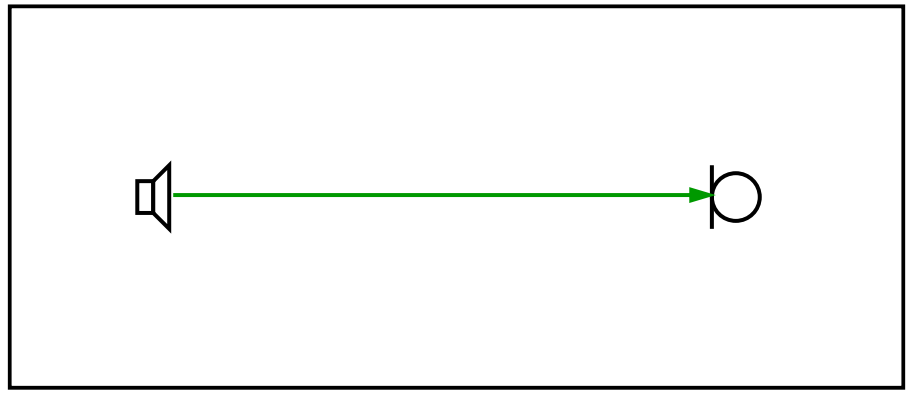
Funkcję autokorelacji można również obliczyć za pomocą FFT:

- obliczenie FFT sygnału,
- obliczenie kwadratu modułu widma,
- obliczenie odwrotnej transformaty (IFFT).

Pomiar odpowiedzi impulsowej

PROBLEM #4: chcemy pomierzyć charakterystykę częstotliwościową zestawu głośnikowego (w jaki sposób przenosi różne częstotliwości).

- Pomiar wykonujemy w komorze bezechowej – brak odbić dźwięku.
- Z głośnika odtwarzamy sygnały testowe.
- Mikrofon pomiarowy wysokiej klasy nagrywa dźwięk.
- Jako sygnału testowego używamy takiego sygnału, którego funkcja autokorelacji jest równa delcie (δ): 1 w zerze, 0 poza nim. Przykładowe sygnały:
  - *sweep*: sinus o liniowo lub logarytmicznie przestrajanej częstotliwości,
  - *MLS* (*Maximum Length Sequence*)
    – zapętlony sygnał pseudoprzypadkowy.



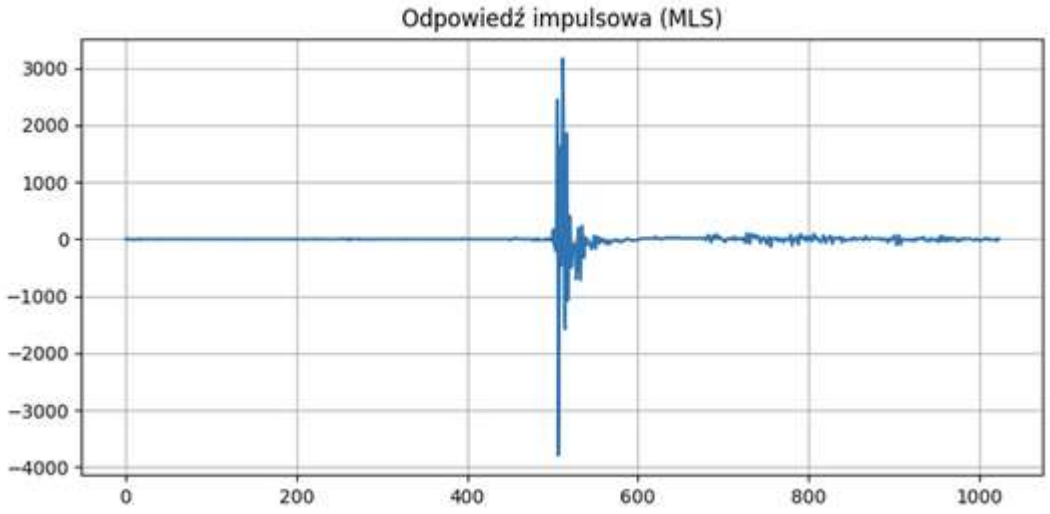
Funkcja korelacji

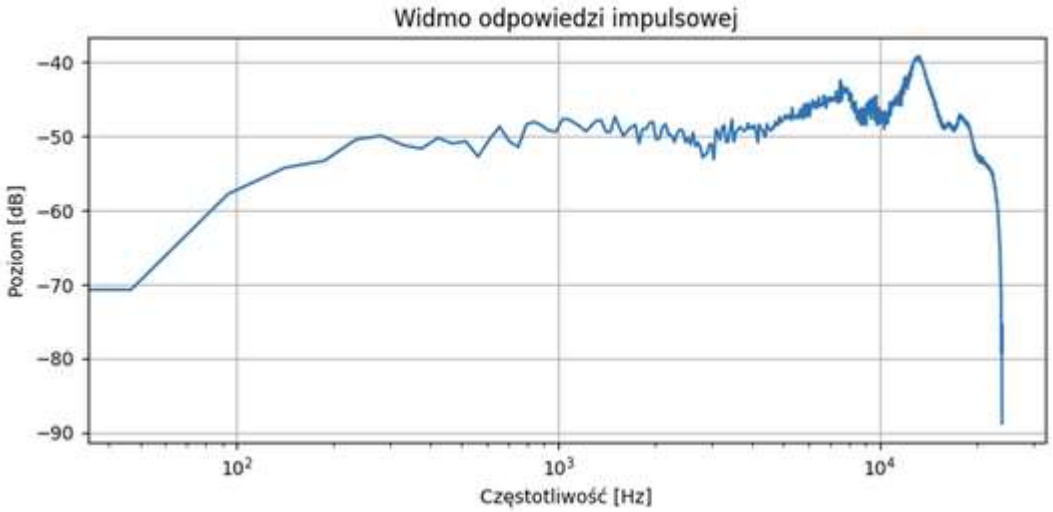
- Funkcja korelacji: miara podobieństwa między dwoma sygnałami, dla różnego przesunięcia w czasie między sygnałami.
- Podobnie do autokorelacji, ale tutaj mamy dwa różne sygnały.
- Metody obliczania korelacji:
  - w dziedzinie czasu: obliczenie splotu, przy czym drugi sygnał jest odwrócony „tyłem do przodu”,
  - w dziedzinie częstotliwości (zwykle szybsza metoda):
    - obliczenie FFT obu sygnałów,
    - przemnożenie FFT pierwszego sygnału przez sprzężone FFT drugiego (sprzężenie: odwracamy znak części urojonej),
    - obliczenie odwrotnej FFT z wyniku.

Pomiar odpowiedzi impulsowej

Wracamy do naszego problemu – pomiaru charakterystyki głośnika.

- Obliczamy funkcję korelacji sygnału odebranego przez mikrofon z sygnałem nadanym z głośnika (*sweep* lub MLS, autokorelacja = δ).
- Wynik (funkcja korelacji) jest odpowiedzią impulsową układu.
- FFT z odpowiedzi impulsowej jest charakterystyką częstotliwościową.

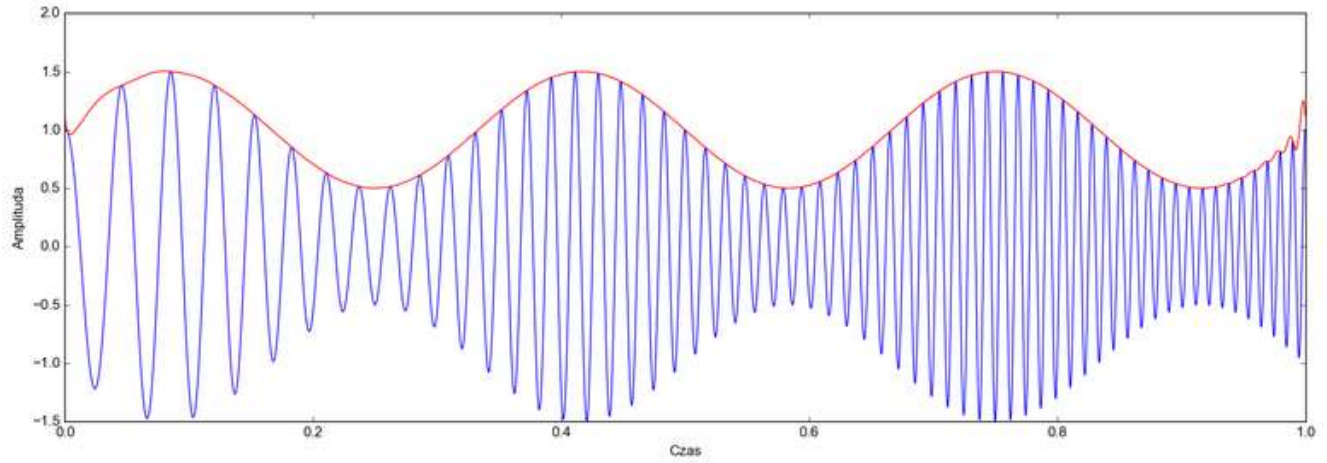




Analiza złożonego sygnału

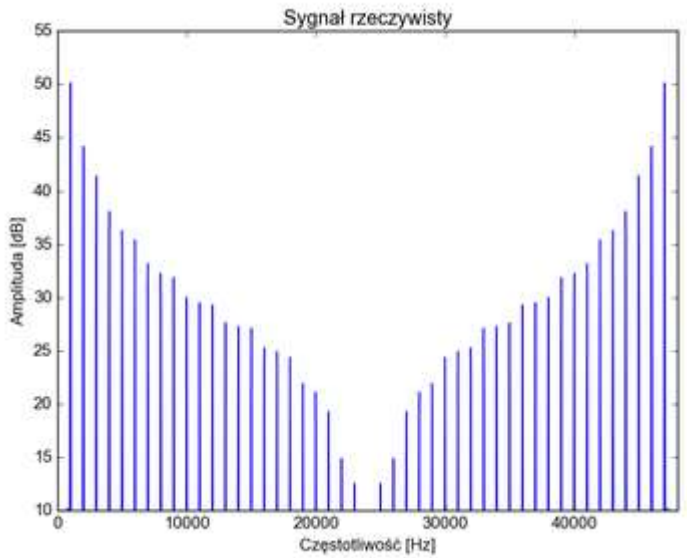
PROBLEM #5

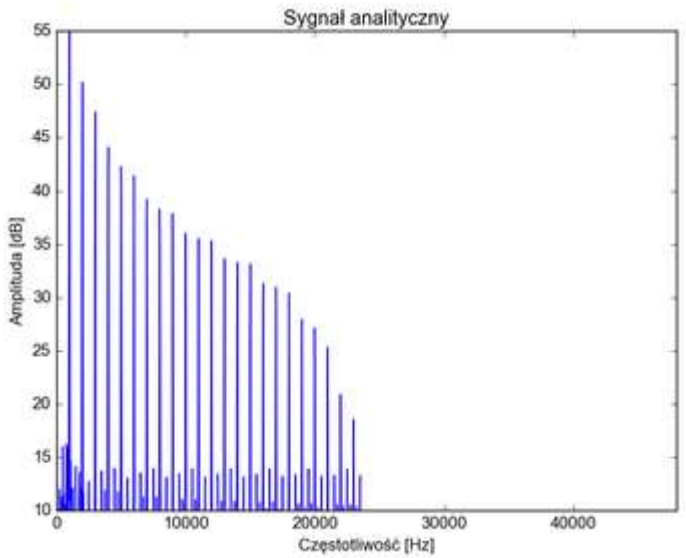
- Mamy następujący złożony sygnał:
  - sygnał świergotowy (*chirp*) o liniowo narastającej częstotliwości od 20 do 100 Hz,
  - zmodulowany amplitudowo: sygnał przemnożony przez sinus o częstotliwości 3 Hz (czerwona linia na wykresie).
- Dla dowolnej chwili chcemy znaleźć:
  - amplitudę obwiedni sygnału,
  - chwilową wartość częstotliwości.



Sygnał analityczny

- Sygnał analityczny: sygnał zespolony w którym:
  - część rzeczywista: oryginalny sygnał,
  - część urojona: sygnał przesunięty w fazie o 90 stopni.
- Jedna z metod: obliczenie widma, wyzerowanie jego drugiej części.





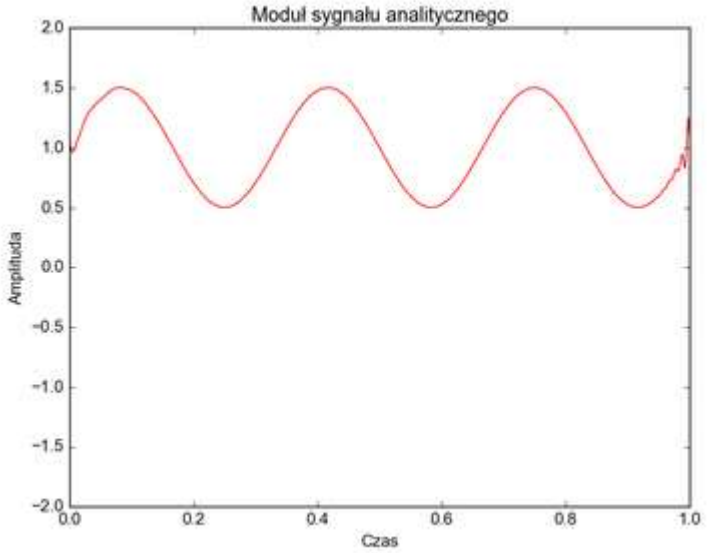
Obwiednia sygnału

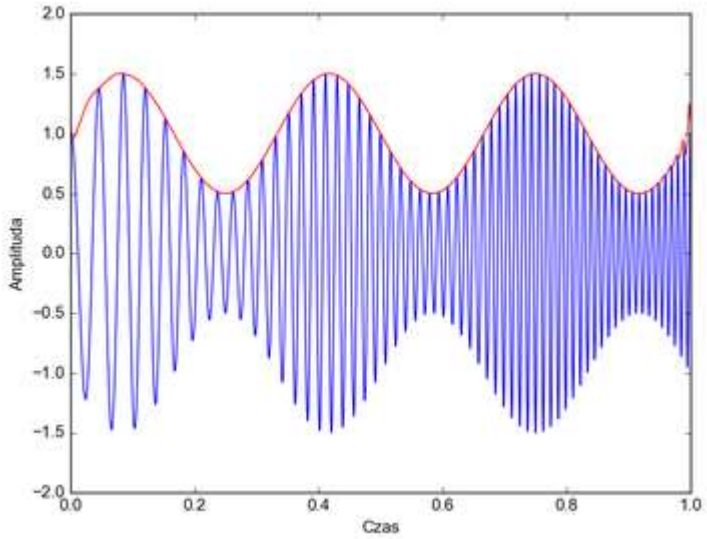
Co nam daje sygnał analityczny w naszym przypadku:

moduł sygnału analitycznego *r*(*n*):

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

jest obwiednią oryginalnego sygnału.





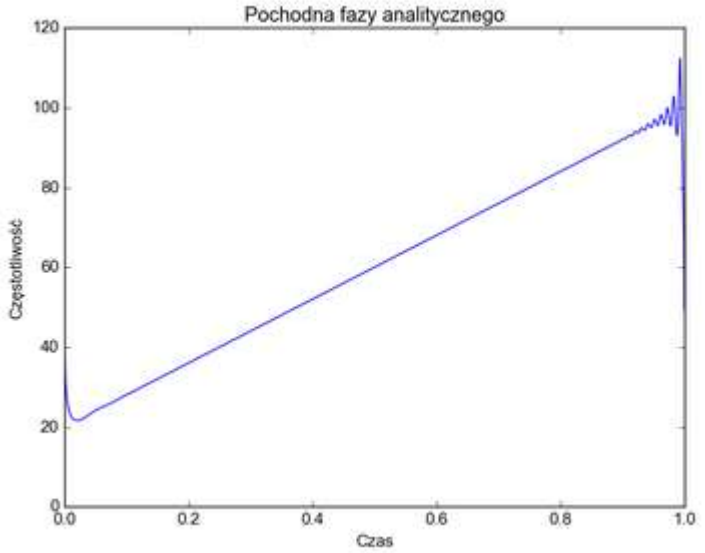
Częstotliwość chwilowa

- Faza chwilowa sygnału analitycznego:

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

- Częstotliwość chwilowa = pochodna fazy:

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



- W ten sposób uzyskujemy chwilową częstotliwość zmodulowanego sygnału.

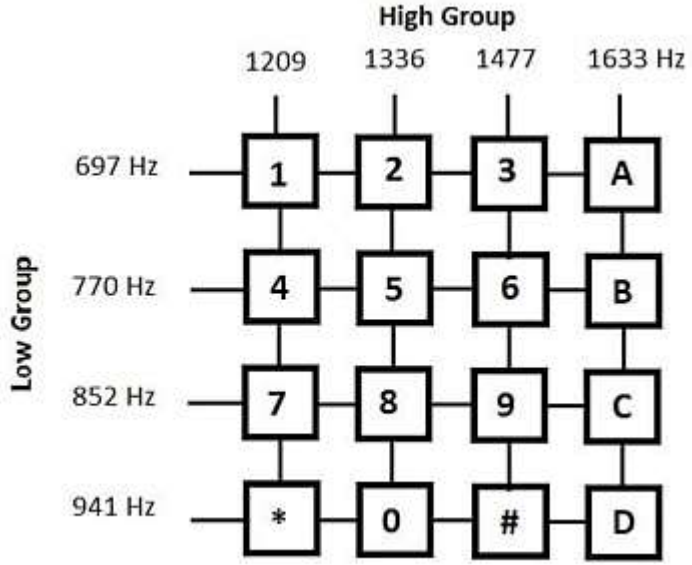
Transformator Hilberta w DSPLIB

- Algorytm przetwarzający sygnał rzeczywisty w analityczny nazywa się transformatorem Hilberta.
- Sumuje on wejściowy sygnał z wynikiem przetwarzania sygnału przez filtr Hilberta, który zmienia fazę sygnału o 90 stopni (zamienia sygnał rzeczywisty w urojony).
- Można też użyć FFT, zerując „wyższą” część widma.
- W bibliotece DSPLIB:

hilb16	*FIR Hilbert Transformer*
Function	ushort oflag = hilb16 (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nx, ushort nh)

Bonus - DTMF

- DTMF – *Dual-Tone Multi Frequency.*
- Metoda kodowania znaków w telekomunikacji.
- Każdy z 16 znaków jest reprezentowany przez dwuton
  – dwa sinusy, dwie spośród 8 częstotliwości.
- Np. cyfra 6:
  770 Hz + 1477 Hz
- Zastosowanie
  – np. wybieranie tonowe
  w telefonii analogowej.



Bonus - DTMF

„Zadanie domowe” – do samodzielnego przemyślenia.

Jak zrobić układ rozpoznawania kodów DTMF za pomocą procesora sygnałowego?

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

Podobny, ale trudniejszy problem:
jak zrobić detektor sygnałów nadawanych alfabetem Morse’a (. -)?

Pozostawiamy studentom do zastanowienia się, jako praktyczne wykorzystanie wiedzy z wykładów.