

LABORATORIUM CYFROWEGO PRZETWARZANIA SYGNAŁÓW

Ćwiczenie 2

Właściwości i projektowanie filtrów cyfrowych

1. Cel ćwiczenia

- Poznanie metod projektowania filtrów NOI i SOI.
- Zapoznanie się z właściwościami projektowanych filtrów.
- Utrwalenie poznanych wcześniej wiadomości związanych z przetwarzaniem sygnałów przez filtry cyfrowe.

2. Podstawy teoretyczne

W poprzednich ćwiczeniach analizowane były sygnały na wejściu i wyjściu filtra cyfrowego o znanych właściwościach transmisyjnych, opisanych odpowiedzią impulsową $h(n)$ lub transmitancją $H(z)$. Chodziło więc o *analizę* sygnałów i układów przetwarzających te sygnały.

W zagadnieniach praktycznych problem polega na usunięciu z sygnału wchodzącego do układu szumów i innych zakłóceń i wyodrębnieniu sygnału użytecznego przy zachowaniu dopuszczalnego poziomu zniekształceń. Jest to zadanie *filtracji*. Innym problemem jest *celowe przekształcenie* danego sygnału w inny. Przykładem mogą być np. filtry różniczkujące, filtry Hilberta lub banki filtrów pasmowych służące do dekompozycji sygnału (np. w celu jego kompresji). W obu przypadkach dochodzi się do problemu *syntezy* filtra cyfrowego na podstawie narzuconych wymagań projektowych.

2.1. Sformułowanie problemu

Wymagania projektowe do zadania projektowania filtra określa się najczęściej w *d dziedzinie częstotliwości*. W przypadku kiedy znane jest pasmo sygnału użytecznego, a widmo zakłóceń leży poza tym pasmem, filtr powinien mieć stałą (np. równą 1) charakterystykę amplitudową $M_0(\Omega)$ (*magnitude*) i liniową charakterystykę fazową $\varphi(\Omega)$ w tym paśmie (paśmie przepustowym). Poza pasmem użytecznym charakterystyka amplitudowa powinna mieć wartości bliskie zeru.

Problem jest znacznie trudniejszy, jeżeli widmo zakłóceń pokrywa się z widmem sygnału użytecznego. Odfiltrowanie zakłóceń prowadzi wówczas do zniekształceń sygnału i konieczne jest znalezienie kompromisu. Optymalnym rozwiązaniem tego typu problemu jest tzw. *filtracja Wienera*.

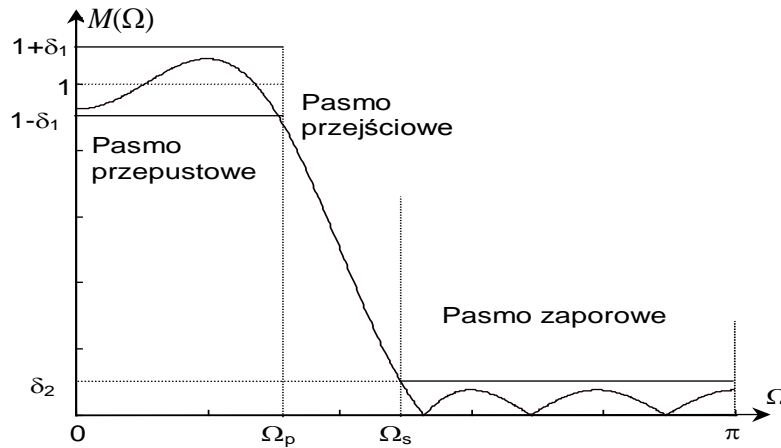
W ćwiczeniu ograniczymy się do prostszego przypadku: wyznaczania transmitancji filtra na podstawie wymagań narzuconych na przebieg *charakterystyki amplitudowej* $M(\Omega)$ i charakterystyki *opóźnienia grupowego*

$$\tau(\Omega) = - \frac{d\varphi(\Omega)}{d\Omega} \quad (3.1)$$

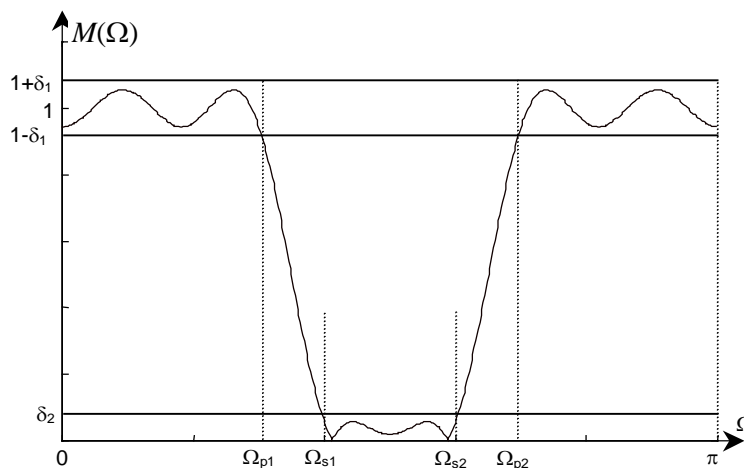
gdzie $\Omega = \omega T_s = 2\pi f / f_s$ jest unormowaną pulsacją kątową, $\varphi(\Omega) = \arg H(z) \big|_{z=e^{j\Omega}}$.

Stała (w przybliżeniu) wartość opóźnienia grupowego w określonym przedziale częstotliwości oznacza liniowy przebieg fazy $\varphi(\Omega)$, co powoduje zachowanie kształtu przebiegów w czasie i często jest niezbędne np. w przypadku przetwarzania sygnałów zmodulowanych fazowo.

Na Rys. 1 przedstawione są typowe warunki określające kształt charakterystyki amplitudowej filtra dolnoprzepustowego (LP) oraz pasmowozaporowego (BS). Charakterystyka $M(\Omega)$ powinna przebiegać w przedziałach tolerancji, które – ze względu na okresowość charakterystyki – wystarczy podać dla $0 \leq \Omega \leq \pi$.



a) filtr dolnoprzepustowy (*lowpass* – LP)



b) filtr pasmowozaporowy (*bandstop* – BS)

Rys. 1. Przedziały tolerancji charakterystyk amplitudowych w funkcji pulsacji unormowanej względem pulsacji próbkowania ($\Omega=\pi$ odpowiada pulsacji Nyquista – połowie pulsacji próbkowania)

Zadanie polega na takim doborze transmitancji $H(z)$, aby charakterystyka amplitudowa

$$M(\Omega) = |H(z)|_{z=e^{j\Omega}} \quad (3.2)$$

spełniała warunki:

- $1 - \delta_1 \leq M(\Omega) \leq 1 + \delta_1$ w pasmach przepustowych,
- $M(\Omega) \leq \delta_2$ w pasmach zaporowych.

Ω_p i Ω_s są pulsacjami granicznymi pasm. Tolerancje δ_1 , δ_2 są podawane przeważnie w decybelach.

Zagadnienie polega zatem na aproksymacji zadanej charakterystyki amplitudowej $M_0(\Omega)$ funkcją $M(\Omega)$ określonej klasy. Dalej będziemy rozpatrywać układy liniowe i stacjonarne, których transmitancja ma postać funkcji wymiernej zmiennej zespolonej z :

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_N z^{-N}}{a_0 + a_1 z^{-1} + \dots + a_N z^{-N}}, \quad a_0 = 1 \quad (3.3)$$

W takim przypadku funkcja aproksymująca $M^2(\Omega)=H(z)H(z^{-1})$, $z=e^{j\Omega}$, jest zawsze ilorazem dwóch wielomianów trygonometrycznych, co utrudnia dokładną aproksymację. Projektowany filtr powinien być układem *stabilnym* (wszystkie bieguny $H(z)$ muszą leżeć wewnątrz okręgu jednostkowego) i *przyczynowym* (stopień licznika $B(z)$ nie może być większy od stopnia mianownika $A(z)$).

2.2. Układy minimalnofazowe

Chociaż nie ma żadnych ograniczeń co do położenia zer transmitancji (3.3), z reguły wymaga się, aby zera transmitancji również leżały wewnątrz okręgu jednostkowego. Filtry stabilne spełniające taki dodatkowy warunek nazywamy układami *minimalnofazowymi*. Charakteryzują się one wzajemną zależnością pomiędzy charakterystyką amplitudową i fazową. Jeżeli utworzymy funkcję zespoloną

$$\hat{H}(e^{j\Omega}) = \ln H(e^{j\Omega}) = \ln M(\Omega) + j\varphi(\Omega) \quad (3.4)$$

to widać, że logarytmiczna charakterystyka amplitudowa i charakterystyka fazowa są odpowiednio jej składową rzeczywistą i urojoną. Składowe te są ze sobą powiązane, jeżeli funkcja $\hat{H}(z)$ jest holomorficzną w obszarze $|z| \geq 1$, co oznacza, że nie może tam być zer transmitancji $H(z)$. Związek składowych określają tzw. wzory Hilberta. Projektując filtr minimalnofazowy o zadanej np. charakterystyce amplitudowej $M(\Omega)$, otrzymuje się więc automatycznie wynikający z niej przebieg charakterystyki fazowej $\varphi(\Omega)$.

2.3. Filtry wszechprzepustowe

Jeżeli zrezygnuje się z warunku minimalnofazowości - wprowadzając do transmitancji zera $|z| > 1$ - można nakładać *niezależne* wymagania na przebieg amplitudy i fazy, ale projektowanie jest bardziej skomplikowane. Filtry nieminimalnofazowe stosuje się jako *korektory fazy*, ze względu na możliwość niezależnego kształtowania $\varphi(\Omega)$.

Jeżeli wielomian licznika w transmitancji (3.3) jest wielomianem odwrotnym do wielomianu mianownika, tzn. $B(z) = z^{-N}(a_N z^N + a_{N-1} z^{N-1} + \dots + a_1 z + 1) = z^{-N} A(z^{-1})$, to otrzymujemy transmitancję tzw. *filtra wszechprzepustowego*:

$$H(z) = \frac{z^{-N} A(z^{-1})}{A(z)} \quad (3.5)$$

Charakterystyka amplitudowa takiego filtra jest w całym zakresie częstotliwości stała i równa 1. Zera transmitancji (3.5) są zwierciadlanym odbiciem biegunów względem okręgu jednostkowego, tzn. każdemu (stabilnemu) biegunowi $z_1 = r_1 \exp(j\Omega_1)$ leżącemu wewnątrz okręgu $|z|=1$ towarzyszy niestabilne zero $z_0 = 1/z_1$ leżące na zewnątrz okręgu. Charakterystykę opóźnienia grupowego $\tau(\Omega)$ takiego układu można odpowiednio ukształtować przez odpowiedni wybór wielomianu $A(z)$.

2.4. Projektowanie filtrów NOI (IIR)

Filtry o nieskończonej odpowiedzi impulsowej (NOI lub IIR – *Infinite Impulse Response*) mają transmitancję wymierną postaci (3.3). Ograniczymy się do omówienia metod spełniających ograniczenia narzucone na tylko przebieg charakterystyki amplitudowej. W wyniku projektowania otrzymuje się współczynniki wielomianów $B(z)$ i $A(z)$ lub – równoważnie – położenie zer i biegunów transmitancji $H(z)$.

2.4.1. Dyskretyzacja prototypów analogowych

Metoda ta umożliwia projektowanie filtrów o standardowych charakterystykach LP, HP, BS oraz BP i polega na zaprojektowaniu najpierw *filtra analogowego* o transmitancji ciągłej $H_a(s)$ spełniającego postawione wymagania, a następnie przekształceniu $H_a(s)$ do docelowej postaci dyskretniej $H(z)$. Zaletą tej metody są stosunkowo proste, bezpośrednie formuły projektowe, a wadą brak możliwości projektowania filtrów o dowolnym przebiegu charakterystyki $M(\Omega)$.

Jako prototypy analogowe wykorzystuje się kilka typów filtrów różniących się właściwościami:

- filtr Butterwortha (o maksymalnie płaskiej charakterystyce amplitudowej w paśmie przepustowym),
- filtr Czebyszewa I rodzaju (o charakterystyce równomiernie pofalowanej w paśmie przepustowym i monotonicznej w paśmie zaporowym),
- filtr Czebyszewa II rodzaju (o charakterystyce równomiernie pofalowanej w paśmie zaporowym i monotonicznej w paśmie przepustowym),
- filtr eliptyczny (Cauera, o charakterystyce równomiernie pofalowanej w obu pasmach i najwęższym (najbardziej stromym – dla określonego rzędu N) paśmie przejściowym),
- filtr Bessela (wersja analogowa ma prawie stałą wartość opóźnienia grupowego $\tau(\Omega)$ w paśmie przepustowym i monotoniczną charakterystykę amplitudową).

Przy projektowaniu filtra trzeba osiągnąć kompromis pomiędzy stromością pasma przejściowego a liniowością charakterystyki fazowej, która w wielu zastosowaniach, m.in. w technice audio, może mieć istotne znaczenie. Rozwiązania krańcowe to filtr eliptyczny oraz filtr Bessela, natomiast filtr Butterwortha ma właściwości pośrednie.

Charakterystykę amplitudową prototypów analogowych można opisać wspólną zależnością:

$$M_a^2(\omega_a) = H_a(j\omega)H_a^*(j\omega) = \frac{1}{1 + \varepsilon^2 F^2(\omega_a / \omega_{a0})} \quad (3.6)$$

gdzie F jest pewną funkcją zależną od typu i rzędu filtra (np. dla filtra Butterwortha $F(\omega_a / \omega_{a0}) = (\omega_a / \omega_{a0})^N$), ω_{a0} jest "analogową" pulsacją nominalną (odcinka), a ε parametrem charakteryzującym pofalowanie charakterystyki (zależnym od tolerancji δ_1 lub δ_2 ; dla filtra Butterwortha $\varepsilon=1$).

Zaprojektowany prototyp analogowy przekształca się do postaci dyskretnej korzystając z określonej metody dyskretyzacji. Najczęściej stosuje się:

- metodę odwzorowania biliniowego płaszczyzny s na płaszczyznę z , określoną podstawieniem

$$H(z) = H_a(s) \Big|_{s = \frac{2}{T_s} \frac{1-z^{-1}}{1+z^{-1}}} \quad (3.7)$$

gdzie $H_a(s)$ jest transmitancją Laplace'a analogowego prototypu filtra.

Przekształcenie (3.7) odwzorowuje lewą półpłaszczyznę $\text{Re } s < 0$ we wnętrze okręgu jednostkowego $|z| < 1$, a zatem po dyskretyzacji zachowana jest stabilność (lub niestabilność) układu analogowego. Wadą tej metody jest zniekształcanie skali częstotliwości określone zależnością

$$\Omega = 2 \arctg \frac{\omega_a T_s}{2} \quad (3.8)$$

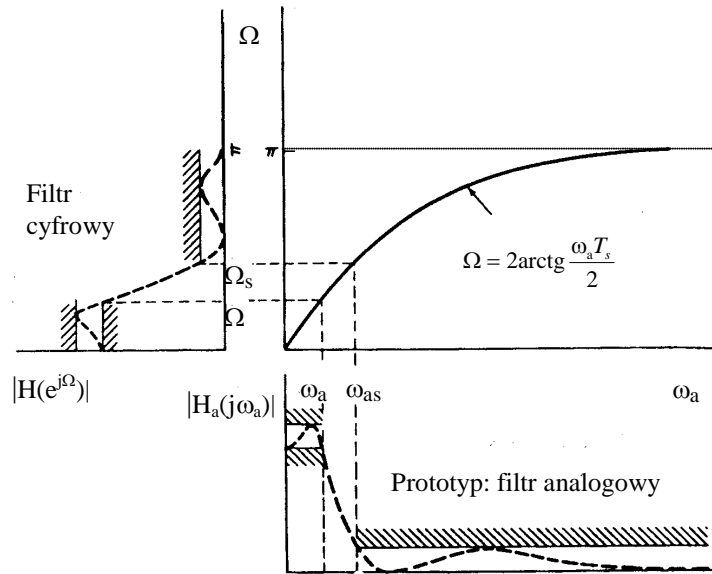
gdzie $\Omega = \omega_a T_s$ jest unormowaną dyskretną pulsacją kątową z zakresu $[-\pi, \pi]$. Zniekształcenie to należy uwzględnić przy projektowaniu częstotliwości granicznych pasm filtra (Rys. 2).

- metodę *niezmienniczej odpowiedzi impulsowej*. Jest to przekształcenie opierające się na warunku takich samych wartości odpowiedzi impulsowej prototypu analogowego i filtra dyskretnego w momentach próbkowania: $h_d(n) = h_a(t) |_{t=nT_s}$. Procedura polega na rozłożeniu $H_a(s)$ na ułamki proste:

$$H_a(s) = K + \sum_{k=1}^N \frac{R_k}{s - s_k} \quad (3.9)$$

i przekształceniu biegunów ciągłych na dyskretne według zależności $z = e^{sT_s}$. W rezultacie otrzymuje się transmitancję dyskretną

$$H(z) = K + \sum_{k=1}^N \frac{R_k}{1 - e^{s_k T_s} z^{-1}} \quad (3.10)$$



Rys. 2. Zniekształcenie skali częstotliwości powodowane przez odwzorowanie biliniowe

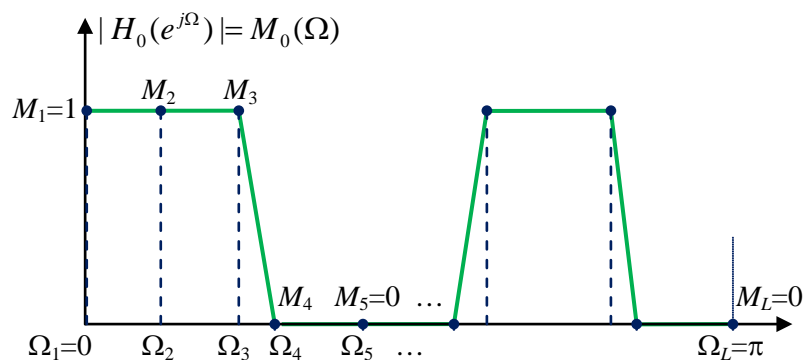
Przekształcenie częstotliwości jest liniowe, ale ze względu na okresowość funkcji $e^{j\omega T_s}$ występuje efekt sumowania się okresowo poprzesuowanych charakterystyk (*aliasing*):

$$H(e^{j\Omega}) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} H_a\left(j\frac{\Omega}{T_s} + jk\omega_s\right) \quad (3.11)$$

Z tego powodu metoda niezmienniczej odpowiedzi impulsowej nie nadaje się do dyskretyzacji prototypów analogowych, które mają pasmo przepustowe w zakresie wysokich częstotliwości, czyli filtrów typu HP i BS (górnoprzepustowych i pasmowozaporowych), ponieważ wynikowa charakterystyka filtra dyskretnego jest wtedy bardzo zniekształcona.

2.4.2. Optymalizacyjne metody projektowania

Metody te umożliwiają projektowanie filtrów o dowolnym (kawałkami liniowym) kształcie charakterystyki amplitudowej, której wymagany przebieg $M_0(\Omega)$ jest określony na zbiorze pulsacji dyskretnych $\{\Omega_i : i = 1, 2, \dots, L\}$ z przedziału $[0, \pi]$ (współrzędne Ω_i określają zwykle punkty załamania charakterystyki, czyli granice pasm przepustowych i zaporowych, Rys. 3).



Rys. 3. Zadany przebieg kawałkami liniowej charakterystyki amplitudowej filtra

Jako wskaźnik jakości optymalizacji definiuje się błąd aproksymacji. Przy projektowaniu filtrów NOI jest to najczęściej *błąd średniokwadratowy*

okręgu jednostkowym) ma swe zwierciadlane odbicie $1/z_i^*$, co oznacza, że filtry SOI o liniowej fazie nie są minimalnofazowe.

2.5.1. Metoda okienkowa (aproxymacja za pomocą sumy skończonej szeregu Fouriera)

Dla idealnych filtrów selektywnych wymagana charakterystyka $H_0(e^{j\Omega})$ ma nieciągłości między pasmami, co sprawia, że można ją uzyskać tylko za pomocą nieskończonej i nieprzyczynowej (niezerowe próbki dla $n < 0$) odpowiedzi impulsowej

$$H_0(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} h_0(n)e^{-j\Omega n}, \quad \text{gdzie } h_0(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_0(e^{j\Omega})e^{j\Omega n} d\Omega \quad (3.17)$$

Metoda polega na wycięciu z $h_0(n)$ fragmentu $h_L(n)$ o skończonej długości dla $-L \leq n \leq L$, co odpowiada pomnożeniu przez symetryczne okno (prostokątne): $h_L(n) = h_0(n)w(n)$. Otrzymuje się *symetryczną*

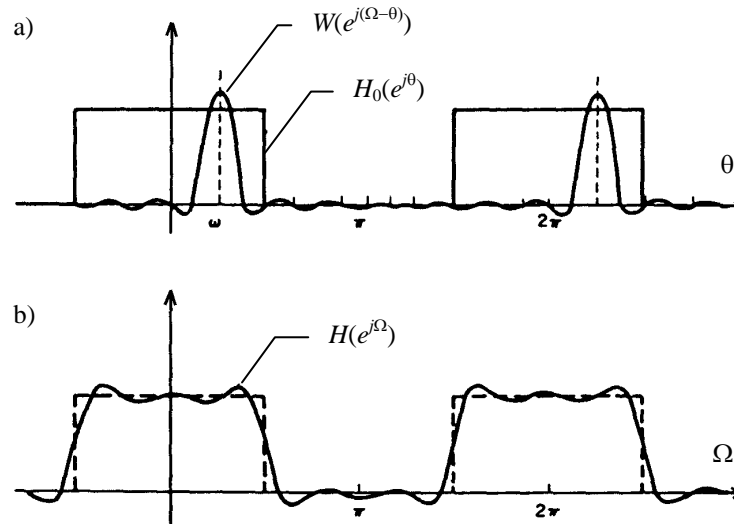
transmitancję $H_L(e^{j\Omega}) = \sum_{n=-L}^L h_L(n)e^{-j\Omega n}$ aproksymującą $H_0(e^{j\Omega})$ za pomocą skończonego szeregu

Fouriera. Aby otrzymać transmitancję przyczynową należy $H_L(z) = \sum_{n=-L}^L h_L(n)z^{-n}$ pomnożyć przez z^{-L} ,

co odpowiada przesunięciu w czasie. W efekcie otrzymujemy transmitancję filtra SOI o *liniowej fazie*:

$$H(z) = z^{-L}H_L(z) = \sum_{n=0}^N h(n)z^{-n}, \quad N = 2L, \quad h(n) = h_L(n - N/2) \quad (3.18)$$

Efekt splatania wymaganej charakterystyki częstotliwościowej $H_0(e^{j\Omega})$ z charakterystyką okna $W(e^{j\Omega})$ wynikający z obciążenia nieskończonej odpowiedzi impulsowej $h_0(n)$ jest pokazany na Rys. 5. Charakterystyka amplitudowa filtra $H(e^{j\Omega})$ jest „rozmytą” aproksymacją $H_0(e^{j\Omega})$ z widocznym *efektem Gibbsa* na granicach pasm (maksymalny błąd aproksymacji nie maleje tam do 0 gdy $L \rightarrow \infty$, ale dąży do wartości ok. 0.09). Pofalowania charakterystyki filtra można zmniejszyć przez zastosowanie okna $w(n)$ innego niż prostokątne (np. Hamminga, Blackmana czy Kaisera), o charakterystyce częstotliwościowej z niższymi listkami bocznymi (por. ćwiczenie *Dyskretne sygnały deterministyczne i analiza widmowa*), ale płaci się za to cenę w postaci poszerzenia pasma przejściowego.



Rys. 5. Rozmywanie charakterystyki filtra FIR wynikające z obciążenia nieskończonej odpowiedzi impulsowej $h_0(n)$ odpowiadającej wymaganej transmitancji $H_0(e^{j\Omega})$: a) operacja splotu wynikająca z nałożenia okna: $DTFT[h_0(n)w(n)] = H_0(e^{j\Omega}) * W(e^{j\Omega})$, b) typowa aproksymacja wymaganej charakterystyki za pomocą transmitancji $H(e^{j\Omega})$ odpowiadającej obciętej odpowiedzi $h(n)$

2.5.2. Optymalizacyjne metody projektowania (aproxymacja minimaksowa)

Metody te umożliwiają projektowanie filtrów o dowolnym (kawałkami liniowym) kształcie wymaganej charakterystyki amplitudowej $H_0(e^{j\Omega})$ przy pomocy procedur optymalizacyjnych minimalizujących błąd średniokwadratowy postaci (3.12) lub, co stosowane jest częściej, tzw. metody *minimaksowe* minimalizujące maksymalną odchyłkę aproksymacji postaci

$$\max_{\Omega_i} E(\Omega_i) = W(\Omega_i) (|H_0(e^{j\Omega_i})| - |H(e^{j\Omega_i})|) \quad (3.19)$$

z wagami $W(\Omega_i)$. Aproksymacja minimaksowa (Czebyszewa) daje charakterystykę amplitudową *równomiernie pofalowaną (equiripple)*. Znajomość maksymalnego odchylenia od charakterystyki wymaganej daje projektantowi kontrolę nad utrzymaniem charakterystyki filtra w zadanych granicach tolerancji, chociaż błąd średniokwadratowy aproksymacji minimaksowej jest większy. Do obliczeń wykorzystuje się procedurę Parksa-McClellana z iteracyjnym algorytmem wymiany Remez z wyrównującym maksymalne odchyłki błędów (3.19) na zadanych przedziałach pulsacji Ω i w punktach granicznych pasm.

2.6. Realizacje filtrów SOI

W typowym przypadku przetwarzania sygnału wejściowego „próbka po próbce” algorytm pracy filtra wynika bezpośrednio z transmitancji (3.12):

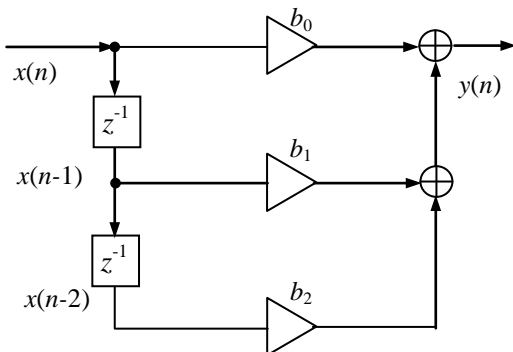
$$y(n) = \sum_{k=0}^N h(k)x(n-k), \quad h(k) = b_k \quad (3.20)$$

Zależność (3.20) stanowi splot i odpowiada jej struktura transwersalna pokazana na Rys. 5 (opóźnione próbki $x(n)$ są przechowywane w pamięci). Do realizacji filtrów SOI (również filtrów NOI z samymi biegunami $H(z)=1/A(z)$) często stosuje się tzw. struktury kratownicowe (*lattice filters*) ze względu na dobre właściwości numeryczne.

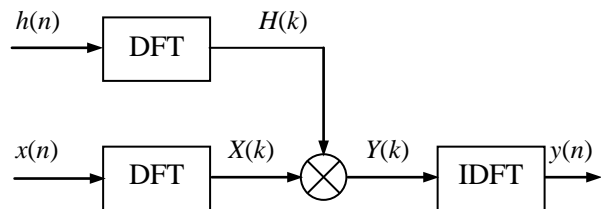
W przypadku przetwarzania *blokowego*, tzn. obliczania odpowiedzi na kolejne *ciągi próbek* sygnału wejściowego, i filtrów wysokiego rzędu (np. kilka tysięcy) stosuje się bardziej efektywny algorytm mnożenia transformat obliczanych metodą FFT (Rys. 7). Oblicza się mianowicie dwie L -punktowe dyskretne transformaty Fouriera: ciągu próbek sygnału wejściowego $x(n), n=0,1,\dots,N_x-1$ oraz ciągu współczynników filtra $h(n), n=0,1,\dots,N-1$ (z ewentualnym uzupełnieniem zerami, aby obie DFT miały jednakowe długości), a następnie mnoży transformaty i oblicza przekształcenie odwrotne

$$y_p(n) = IDFT[H(k)X(k)], \quad (3.21)$$

które odpowiada splotowi cyklicznemu, który daje sygnał okresowy $y_p(n)$ o okresie L . Aby uniknąć aliasingu w czasie i uzyskać równość $y_p(n) = y(n)$ dla $n=0,1,\dots,L-1$, należy przyjmować $L \geq N_x + N - 1$, czyli stosować *uzupełnianie zerami* bloków danych wejściowych do DFT.



Rys. 6. Struktura transwersalna filtra SOI drugiego rzędu ($N=2$)



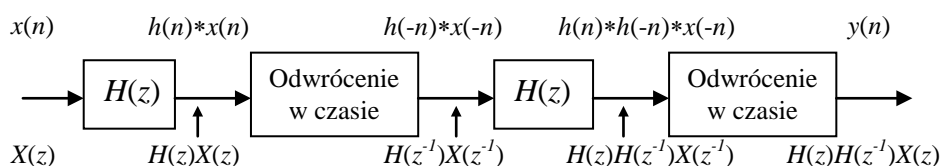
Rys. 7. Realizacja filtra SOI metodą mnożenia transformat DFT przy przetwarzaniu blokowym

2.7. Porównanie filtrów NOI i SOI

Filtry SOI są preferowane w zastosowaniach, w których podstawowym wymaganiem jest liniowość charakterystyki fazowej. Nie zawierają sprzężenia zwrotnego z wyjścia, co upraszcza strukturę i zapewnia stabilne działanie. Filtry NOI mają nieliniowe, szczególnie na brzegach pasm, charakterystyki fazowe, ale przy takim samym rzędzie filtra zapewniają znacznie bardziej strome przejścia charakterystyki amplitudowej między pasmami, czyli lepszą selektywność. W efekcie pożądany przebieg amplitudy można osiągnąć przy (nawet kilkakrotnie) mniejszym nakładzie obliczeń.

2.8. Nieprzyczynowy filtr o zerowym przesunięciu fazowym

Dla określonej transmitancji $H(z)$ możliwe jest zrealizowanie nieprzyczynowego filtra o strukturze przedstawionej na Rys. 7, który ma transmitancję $H(z)H(z^{-1})$ i nie wprowadza przesunięcia fazowego, ponieważ charakterystyka widmowa jest rzeczywista i określona wzorem $|H(e^{j\Omega})|^2$. Dzieje się tak dlatego, że filtr wykorzystuje informacje z punktów leżących w czasie zarówno przed jak i po punkcie bieżącym. Fragment sygnału „z przyszłości” musi więc być z góry znany (zarejestrowany).



Rys. 1. Struktura filtra nieprzyczynowego o zerowym przesunięciu fazowym

3. Obliczenia komputerowe - zadania do wykonania wg instrukcji

Uwaga:

- Bloki instrukcji Matlab'a do realizacji poszczególnych zadań wygodnie jest kopiować do edytora Matlab'a i tam dokonywać edycji parametrów, dopisywania nowych linii itp., a następnie uruchamiać jako skrypty komendą **Run** ▷.
 - Przy przechodzeniu do następnego punktu ćwiczenia zwracać uwagę, żeby nie dochodziło do ewentualnego nakładania się na siebie w tym samym oknie wykresów różnych wielkości. Normalnie rysowanie nowego wykresu kasuje poprzednią zawartość otwartego okna, o ile ta zawartość nie została przeznaczona do zachowania za pomocą instrukcji `hold on` (domyślne `hold off` kasuje poprzednią zawartość pola wykresu w otwartym oknie). Wstawianie `hold on` jest potrzebne, kiedy instrukcja zaleca nakładanie na siebie dla porównania kilku wykresów tej samej wielkości otrzymanych przez powtarzanie jakiegoś bloku instrukcji dla różnych wartości pewnych parametru(-ów). Po zakończeniu danego zadania takie okna wykresów należy zamykać.
 - Do projektowania filtrów wykorzystywane są funkcje biblioteki Matlab'a **Signal Processing Toolbox**. Listing funkcji zapisanej w formie skryptu można otrzymać w oknie komend za pomocą instrukcji `type nazwa_funkcji`.
- W oknie **Current Folder** przejść do folderu **Moje dokumenty/MATLAB/DSP**. Jest to folder roboczy dla ćwiczeń z *Cyfrowego przetwarzania sygnałów*.

3.1. Projektowanie filtrów NOI (IIR) metodą prototypów analogowych

- A.** Zależność charakterystyki częstotliwościowej $H(\Omega)$ i odpowiedzi impulsowej $h(n)$ dyskretnego dolnoprzepustowego (*lowpass* - LP) filtra Butterwortha o transmitancji $H(z)=B(z)/A(z)$ (otrzymanego po dyskretyzacji biliniowej prototypu analogowego $H_a(s)=B_a(s)/A_a(s)$) od rzędu transmitancji N .

```
N=3; w0a=0.3; % rząd i pulsacja nominalna (graniczna) prototypu (rad/s)
fs=1; % częstotl. próbkowania (Hz = 1/s)
wa = linspace(0,6,400); % wektor pulsacji rozłożonych liniowo
disp('Ha(s) prototypu analogowego')
[Ba,Aa]=butter(N,w0a,'low','s') % prototyp analog. Ha(s)=Ba(s)/Aa(s)
disp('H(z) po dyskretyzacji biliniowej')
[B,A]=bilinear(Ba,Aa,fs) % dyskretyzacja Ha(s) dla fs metodą bilinear
[H,W]=freqz(B,A,wa/fs); % ch-ka częstotl. filtra dyskr. H(z)=B(z)/A(z)
tfBut=tf(B,A,1/fs); % utworz.obiektu dyskr.transfer function H(z)=B(z)/A(z)
himp=impz(tfBut,99); % wyzn. L=100 próbek odp. impulsowej [0,L-1]
figure(1)
subplot(211)
plot(W,abs(H)), hold on, % wykres ch-ki amplitudowej
axis([0,1,0,1.1]); grid on,
ylabel('M(\Omega) = |H(\Omega)|'),
subplot(212)
plot(W,unwrap(angle(H))*180/pi), hold on, % wykres ch-ki fazowej
ylabel('\phi(\Omega) (stopnie)'), xlabel('\Omega (rad)'),
xlim([0,1]), grid on
figure(2)
stem(0:99,himp), % wykres odp. impulsowej filtra
ylabel('odp. impulsowa h(n)'), xlabel('n')
```

- Powtórzyć obliczenia dla rzędu $N=5$ i 8 i porównać wyniki. Charakterystyki częstotliwościowe są domyślnie nakładane na jednym wykresie, natomiast nowa charakterystyka impulsowa kasuje na wykresie poprzednią.
- B.** Porównanie charakterystyk częstotliwościowych filtrów *dyskretnych* projektowanych metodą prototypów analogowych dla takich samych parametrów transmitancji.

Projektowane filtry będą filrami LP: 1) tego samego rzędu $N=3$, 2) o tej samej unormowanej pulsacji nominalnej $\Omega_0=0.3\Omega_{Nyq}$ (w odniesieniu do pulsacji Nyquista $\Omega_{Nyq}=\pi$), 3) dopuszczalnym

tłumieniu w paśmie przepustowym $r_p=1\text{dB}$ i minimalnym tłumieniu w paśmie zaporowym $r_s=20\text{dB}$ (parametry te są odpowiednikami granic tolerancji δ_1 i δ_2). Prototypy analogowe będą dyskretyzowane (domyślnie) metodą odwzorowania biliniowego.

Współczynniki wielomianów transmitancji są podawane w oknie komend Matlaba w kolejności: wektor współczynników licznika $B = [b_0 \ b_1 \ \dots \ b_N]$, wektor współczynników mianownika $A = [1 \ a_1 \ \dots \ a_N]$.

- Zamknąć okna wykresów i wyczyścić przestrzeń roboczą (*Workspace*) i okno komend Matlaba

```
clear all, close all, clc
```

- Zaczniemy od projektu filtrów Butterwortha: analogowego i jego dyskretnego odpowiednika:

```
N=3; W0=0.3; % rząd i pulsacja nomin. dyskret.(0,1) (param. transmitancji)
fs = 1; % częst. próbkowania (do porównania z prototypem analog.)
W = linspace(0,2*pi,400); % wektor pulsacji rozłożonych liniowo [0,2pi]

[Ba,Aa] = butter(N,W0,'low','s') % prototyp Ha(s)=Ba(s)/Aa(s) (W0 to rad/s)
[Ha,wa] = freqs(Ba,Aa,W*fs); % odp. częstotl. Ha(w) filtra analogowego
[B,A] = butter(N,W0,'low') % f. Butterwortha LP, współczynniki B(z),A(z)
[H,W] = freqz(B,A,W); % odp. częstotliwościowa H(W) filtra dyskret.

subplot(211) % ch-ka amplitudowa
plot(wa,abs(Ha),':', W/pi,abs(H)),
axis([0,2,0,1.1]);
legend('prototyp analogowy H_a(s)', 'filtr dyskretny H(z)')
ylabel('|H(\Omega)|'),
subplot(212) % ch-ka fazowa
plot(wa,unwrap(angle(Ha))*180/pi,':', W/pi,unwrap(angle(H))*180/pi),
axis([0,2,-360,0]);
xlabel('\Omega / \pi'), ylabel('\phi(\Omega) (stopnie)')
```

Uwaga: Okna wykresów z „subplotami” rozciągnąć w pionie dla uzyskania lepszej widoczności.

- Na podstawie współczynników transmitancji $H_a(s)=B_a(s)/A_a(s)$ i $H(z)=B(z)/A(z)$ z okna komend Matlaba napisać ich transmitancje oraz równanie różnicowe filtra dyskretnego (pamiętać, że w wielomianach $H(z)$ występują ujemne potęgi z).
- W jakim zakresie pulsacji charakterystyki prototypu analogowego i filtra dyskretnego i pokrywają się?
- Rozszerzyć obliczenia dla filtrów dyskretnych innych typów nakładając ich charakterystyki na jednym wykresie z innym kolorem i stylem linii. Zwrócić uwagę, jakie są argumenty funkcji projektowych.

```
N=3; W0=0.3; % rząd i pulsacja nominalna (0,1) (parametr transmitancji)
rp=1; % max tłumienie w paśmie przepustowym (passband rippling)
rs=20; % min tłumienie w paśmie zaporowym (stopband rippling)
W = linspace(0,2*pi,400); % wektor pulsacji rozłożonych liniowo [0,2pi]

[B,A] = butter(N,W0); % f. Butterwortha
[Hb,W] = freqz(B,A,W);

[B,A] = cheby1(N,rp,W0); % f. Czebyszewa I rodzaju
[Hc1,W] = freqz(B,A,W);

[B,A] = cheby2(N,rs,W0); % f. Czebyszewa II rodzaju
[Hc2,W] = freqz(B,A,W);

[B,A] = ellip(N,rp,rs,W0); % f. eliptyczny
[He,W] = freqz(B,A,W);
f = W/pi;

subplot(211) % ch-ki amplitudowe
plot(f,abs(Hb),'b', f,abs(Hc1),'--r', f,abs(Hc2),'m', f,abs(He),'-.g')
legend('Butterworth', 'Czebyszew I', 'Czebyszew II', 'eliptyczny')
```

```
axis([0,2,0,1.1]);
subplot(212) % ch-ki fazowe
plot(f,unwrap(angle(Hb))*180/pi,'b', f,unwrap(angle(Hc1))*180/pi,'--r',...
      f,unwrap(angle(Hc2))*180/pi,':m', f,unwrap(angle(He))*180/pi,'-.g')
axis([0,2,-360,90]);
```

- Na podstawie współczynników transmitancji $H(z)=B(z)/A(z)$ dla poszczególnych filtrów z okna komend Matlaba napisać transmitancje i równania różnicowe zaprojektowanych filtrów (pamiętać, że w wielomianach występują ujemne potęgi z).
- Z czego wynika symetria charakterystyk względem $\Omega/\Omega_{Nyq}=1$?
- Porównać charakterystyki filtrów. Zwrócić uwagę na związek charakterystyk z wartościami parametrów wejściowych r_p lub r_s . Sprawdzić, czy zachowane są zadane wartości r_p i r_s . Określić poziomy wzmacnienia poszczególnych filtrów dla Ω_0 .
- Który filtr ma najbardziej strome, a który najłagodniejsze przejście od pasma przepustowego do zaporowego (najlepszą/najgorszą selektywność)?
- Zwrócić uwagę na nieliniowość charakterystyk fazowych (która jest najmniej nieliniowa, w którym miejscu charakterystyki są najbardziej nieliniowe) i wyjaśnić (na płaszczyźnie zespolonej) skoki fazy o 180° w miejscach punktów zerowych charakterystyki amplitudowej.
- Dla ułatwienia porównania wykreślić charakterystyki amplitudowe na jednym wykresie w skali logarymicznej (oś OX za pomocą `semilogx`, oś OY za pomocą jednostek dB):

```
...
[B,A] = butter(N,W0);
[Hb,W] = freqz(B,A,W);

[B,A] = cheby1(N,rp,W0);
[Hc1,W] = freqz(B,A,W);

[B,A] = cheby2(N,rs,W0);
[Hc2,W] = freqz(B,A,W);

[B,A] = ellip(N,rp,rs,W0);
[He,W] = freqz(B,A,W);
f = W/pi;

semilogx(f,20*log10(abs(Hb)), 'b', f,20*log10(abs(Hc1)), '--r', ...
         f,20*log10(abs(Hc2)), ':m', f,20*log10(abs(He)), '-.g')
axis([0,2,-30,2]), grid, % siatka na wykresie
ylabel('20log |H(\Omega)| (dB)'), xlabel('\Omega / \pi')
legend('Butterworth', 'Czebyszew I', 'Czebyszew II', 'eliptyczny')
```

- Zaobserwować różnice szerokości pasma przejściowego dla poszczególnych filtrów i pofalowanie charakterystyk amplitudowych.

C. Zaprojektować filtry LP typów określonych w pkt. A dla *tej samej specyfikacji projektowej* dotyczącej tolerancji przebiegu charakterystyki amplitudowej i szerokości pasma przejściowego:

- pulsacja graniczna pasma przepustowego $\Omega_p=0.3\Omega_{Nyq}$,
- pulsacja graniczna pasma zaporowego $\Omega_s=0.4\Omega_{Nyq}$,
- maksymalne tłumienie (zafalowanie w dół) w paśmie przepustowym $r_p=1\text{dB}$,
- minimalne tłumienie w paśmie zaporowym $r_s=40\text{dB}$.

Projektowanie według powyższych specyfikacji rozpoczyna się od wyznaczenia rzędu filtra N oraz jego pulsacji nominalnej Ω_0 spełniających wymagania. Parametry te służą następnie do obliczenia współczynników transmitancji jak w pkt. A.

- Dla każdego rodzaju filtra LP, korzystając ze standardowych funkcji Matlaba wyznaczyć parametry transmitancji

```
Wp=0.3; Ws=0.4; rp=1; rs=40; % specyfikacje projektowe
[Nb,Wb] = buttord(Wp,Ws,rp,rs) % filtr Butterwortha
[Nc1,Wc1] = cheblord(Wp,Ws,rp,rs) % filtr Czebyszewa I rodzaju
```

```
[Nc2,Wc2] = cheb2ord(Wp,Ws,rp,rs) % filtr Czebyszewa II rodzaju
[Ne,We] = ellipord(Wp,Ws,rp,rs) % filtr eliptyczny
```

- Zanotować (minimalny) rząd filtra każdego typu zapewniający spełnienie specyfikacji. Zwrócić uwagę, jaki jest związek wyznaczonych pulsacji nominalnych z pulsacjami granicznymi Ω_p i Ω_s .
- Zaprojektować filtry spełniające sformułowane wymagania i sprawdzić, czy charakterystyki amplitudowe przebiegają w przedziałach tolerancji (skala liniowa, $0 \leq \Omega/\Omega_{Nyq} \leq 1$):

```
...
W = linspace(0,pi,400);
[B,A] = butter(Nb,Wb); % wyzn. współczynników liczn. i mian. H(z)
Fbutter = filt(B,A); % tworzenie modelu transmitancyjnego
[Hb,W] = freqz(B,A,W); % wyzn. odp. częstotliwościowej
[B,A] = cheby1(Nc1,rp,Wc1);
Fcheby1 = filt(B,A);
[Hc1,W] = freqz(B,A,W);
[B,A] = cheby2(Nc2,rs,Wc2);
Fcheby2 = filt(B,A);
[Hc2,W] = freqz(B,A,W);
[B,A] = ellip(Ne,rp,rs,We);
Fellip = filt(B,A);
[He,W] = freqz(B,A,W);
f = W/pi;

figure(1)
subplot(211)
plot(f,abs(Hb),'b', f,abs(Hc1),'--r', f,abs(Hc2),':m', f,abs(He),'-.g')
axis([0,1,0,1.1]), grid, % siatka na wykresie
ylabel('|H(\Omega)|'),
legend('Butterworth','Czebyszew I','Czebyszew II','eliptyczny')
subplot(212)
semilogx(f,20*log10(abs(Hb)),'b', f,20*log10(abs(Hc1)),'--r',...
f,20*log10(abs(Hc2)),':m', f,20*log10(abs(He)),'-.g')
axis([0,2,-50,3]), grid, % siatka na wykresie
ylabel('20 log |H(\Omega)| (dB)'), xlabel('\Omega / \pi')
figure(2)
pzmap(Fbutter,Fcheby1,Fcheby2,Fellip) % mapa biegunów i zer (pole-zero map)
legend('Butterworth','Czebyszew I','Czebyszew II','eliptyczny')
title('Rozmieszczenie biegunów (x) i zer (o) filtrów dyskretnych')
xlabel('Re z'), ylabel('Im z')
```

- Zaobserwować mapę rozmieszczenia biegunów (wewnątrz okręgu jednostkowego) i zer zaprojektowanych transmitancji poszczególnych filtrów na płaszczyźnie zespolonej.

D. Porównanie dyskretyzacji prototypu analogowego LP metodami:

- przekształcenia biliniowego (bilinear),
- równoważnej odpowiedzi impulsowej (impinvar)

dla filtra Czebyszewa I rodzaju. Pulsacja nominalna filtra analogowego ω_{0a} zadawana w rad/s.

Pulsacja Nyquista filtra zdyskretyzowanego $\omega_{Nyq} = \pi f_s$ rad/s. Przeprowadzić obliczenia dla $w_{0a}=0.3$ i $w_{0a}=1$.

```
N=3; w0a=0.3; % rząd i pulsacja nominalna (graniczna) prototypu (rad/s)
rp=1; % stopband rippling w dB
fs = 1; % częstotl. próbkowania (1/s)
wa = linspace(0,6,400); % wektor pulsacji rozłożonych liniowo

disp('Ha(s) prototypu analogowego')
[Ba,Aa]=cheby1(N,rp,w0a,'low','s') % prototyp analog. Ha(s) Czebyszewa I
[Ha,wa]=freqs(Ba,Aa,wa);

disp('H(z) po dyskretyzacji "bilinear"')
[Bb,Ab]=bilinear(Ba,Aa,fs) % dyskretyzacja Ha(s) dla fs metodą bilinear
```

```

disp(' H(z) po dyskretyzacji "impinvar"')
[Bi,Ai]=impinvar(Ba,Aa,fs) % dyskretyzacja Ha(s) dla fs metoda impinv
[Hb,wb]=freqz(Bb,Ab,wa/fs); % ch-ki częstotl. filtrów dyskretnych
[Hi,wi]=freqz(Bi,Ai,wa/fs);
semilogx(wa,abs(Ha), wb*fs,abs(Hb),'r-.', wi*fs,abs(Hi),'m--'),
axis([0,6,0,1.1]); grid on,
ylabel('|H_a(s)|, |H(z)|'), xlabel('\omega (rad/s)'),
legend('prototyp analog.', 'dyskretyzacja bilinear', 'dyskretyzacja impinv')

```

- Na podstawie współczynników wielomianów prototypu i filtrów dyskretnych z okna komend Matlaba napisać transmitancje filtrów. Porównać transmitancje otrzymanych filtrów dyskretnych.
- Czy obie metody dyskretyzacji dają filtry o charakterystyce amplitudowej pokrywającej się z prototypem? Która metoda i dlaczego powoduje rozbieżność częstotliwości granicznej filtra dyskretnego w stosunku do prototypu przy większej wartości w_0a ? Zwrócić uwagę, jaki jest w obu przypadkach stosunek ω_{0a} do pulsacji Nyquista $\omega_{Nyq}=\pi f_s$ filtrów dyskretnych.
- Przeprowadzić obliczenia dla filtra pasmowoprzepustowego (*bandpass*, BP) o paśmie przepustowym od 0.3 do 1 i porównać zgodność charakterystyk:

```

w0a=[0.3 1]; % pulsacje graniczne pasma przepustowego prototypu BP
...
[Ba,Aa]=cheby1(N,rp,w0a,'bandpass','s') % prototyp analog. Ha(s)
... (reszta jak poprzednio)

```

- Przeprowadzić obliczenia dla filtra górnoprzepustowego (*highpass*, HP) o paśmie przepustowym od 0.3 i porównać zgodność charakterystyk:

```

w0a=0.3; % pulsacja początku pasma przepustowego prototypu HP
...
[Ba,Aa]=cheby1(N,rp,w0a,'high','s') % prototyp analog. Ha(s)
... (reszta jak poprzednio)

```

- Dlaczego metoda *impinvar* dała zupełnie zły rezultat przy dyskretyzacji filtra HP?

3.2. Projektowanie filtrów NOI o dowolnej charakterystyce amplitudowej

Do projektowania filtrów NOI o *dowolnej* (kawałkami liniowej) charakterystyce amplitudowej służy funkcja:

```
[B,A]=yulewalk(N,F,M);
```

gdzie: N – rząd filtra, F – wektor częstotliwości punktów załamania charakterystyki (z zakresu $[0,1]$, $F=1$ oznacza częstotliwość Nyquista), M - wektor amplitud dla punktów załamania odcinków charakterystyki.

Funkcja ta dopasowuje charakterystykę amplitudową filtra do charakterystyki zadanej za pomocą wektorów (F,M) przez minimalizację sumy kwadratów błędów niedopasowania w poszczególnych węzłach.

- Zamknąć okna wykresów i wyczyścić przestrzeń roboczą (*Workspace*) i okno komend Matlaba.

- Zaprojektować filtr NOI rzędu $N=8$ o dwóch pasmach przepustowych, opisany podanymi wektorami częstotliwości unormowanych i magnitud (wzmocnień):

```

N = 8; % rząd filtra
F = [0 0.15 0.2 0.3 0.35 0.55 0.6 0.8 0.85 1]; % wektor częstotl. unorm.
M = [0 0 1 1 0 0 1 1 0 0]; % wektor magnitud

disp('Filtr NOI Yule-Walkera')
[B,A] = yulewalk(N,F,M) % projektowanie H(z)=B(z)/A(z) metoda Yule-Walkera
[H,W] = freqz(B,A,500); % Wyznaczanie charakterystyki częstotliwościowej

subplot(211)
plot(F,M,':', W/pi,abs(H),'m'),
ylabel('|H(\Omega)|'),

```

```

title('Filtr NOI, metoda Yule-Walkera')
axis([0,1,0,1.1]); grid on, hold on
subplot(212)
phase=unwrap(angle(H))*180/pi;
plot(W/pi,phase,'m'),
axis([0,1,min(phase),max(phase)]); grid on, hold on
xlabel('\Omega / \pi'), ylabel('\phi(\Omega) (stopnie)')

```

- Powtórzyć obliczenia dla $N=20$ nakładając wykresy w jednym oknie (zmienić kolor wykresów dla ułatwienia rozróżnienia). Porównać dokładność aproksymacji zadanej charakterystyki wzmocnienia.

3.3. Projektowanie filtrów SOI (FIR) metodą okienkową

Do projektowania metodą okienkową filtrów SOI o standardowych charakterystykach (LP, HP, BP, BS) służy funkcja:

```
B = fir1(N,W0,'typ',okno); % typ = low, high, stop, bandpass
```

(znaczenie parametrów N, W_0 jak w pkt.3.1, okno określa typ okna czasowego, domyślnie `boxcar(N+1)`). Filtry SOI o dowolnej kawałkami liniowej charakterystyce amplitudowej można projektować za pomocą funkcji:

```
B=fir2(N,F,M,okno);
```

gdzie: N – rząd filtra, F – wektor częstotliwości punktów załamania charakterystyki (z zakresu $[0,1]$, $F=1$ oznacza częstotliwość Nyquista), M - wektor amplitud dla punktów załamania odcinków charakterystyki, okno – typ okna czasowego (o długości $N+1$).

- Zaprojektować filtry LP o ustalonym rzędzie i zadanym przebiegu charakterystyki magnitudy z nałożeniem okna prostokątnego $w(n)=1$ oraz okna Hamminga (charakterystyka filtra będzie aproksymacją przebiegu zadanego; w specyfikacji tej nie podaje się wymaganej tolerancji amplitudy na brzegach pasma przejściowego):

```

N = 20; % rząd filtra
F = [0 0.3 0.35 1]; % wektor częstotl. unorm.
M = [1 1 0 0]; % wektor magnitud

disp('Filtr SOI fir2')
Bb = fir2(N,F,M,boxcar(N+1)) % filtr z oknem prostokątnym
[Hb,W] = freqz(Bb,1,500); % wyzn. ch-ki częstotl. Uwaga: A=1
Bh = fir2(N,F,M) % filtr z domyślnym oknem hamming(N+1)
[Hh,W] = freqz(Bh,1,500);

```

```

subplot(211) % ch-ka amplitudowa
plot(F,M,':', W/pi,abs(Hb),'m', W/pi,abs(Hh),'r--'),
legend('H_0','okno prostokątne','okno Hamminga'),
ylabel('|H(\Omega)|'),
title('Filtr SOI, metoda okienkowa fir2')
axis([0,1,0,1.1]); grid on,
subplot(212) % ch-ka fazowa
phib = unwrap(angle(Hb))*180/pi;
phih = unwrap(angle(Hh))*180/pi;
plot(W/pi,phib,'m', W/pi,phih,'r--'),
grid on,
xlabel('\Omega / \pi'), ylabel('\phi(\Omega) (stopnie)')

```

- Zwrócić uwagę na liniowość charakterystyki fazowej.

- Dla porównania wykreślić dodatkowo charakterystyki amplitudowe w skali logarytmicznej (w dB):

```

...
subplot(211) % ch-ka amplitudowa
plot(F,M,':', W/pi,abs(Hb),'m', W/pi,abs(Hh),'r--'),
legend('H_0','okno prostokątne','okno Hamminga'),
ylabel('|H(\Omega)|'),

```

```

title('Filtr SOI, metoda okienkowa fir2')
axis([0,1,0,1.1]); grid on,
subplot(212) % ch-ka fazowa
Mb = 20*log10(abs(Hb));
Mh = 20*log10(abs(Hh));
phih = unwrap(angle(Hh))*180/pi;
plot(W/pi,Mb,'m', W/pi,Mh,'r--'),
axis([0,1,-100,10]);grid on,
xlabel('\Omega / \pi'), ylabel('|H(\Omega)| (dB)')
...

```

- Porównać otrzymane charakterystyki. Zaobserwować wpływ typu okna na stopień pofalowań charakterystyk i ich stromość w paśmie przejściowym. Porównać poziom tłumienia obu filtrów w paśmie zaporowym.
- W oknie komend zwrócić uwagę na symetrię współczynników transmitancji filtra $H(z)=B(z)$. Naszkicować wykres ich wartości, czyli charakterystykę impulsową $h(n)$, np.:

```

stem(0:N,Bh) % (indeksy Bh zaczynają się od 1)
xlabel('n'), ylabel('h(n)')

```

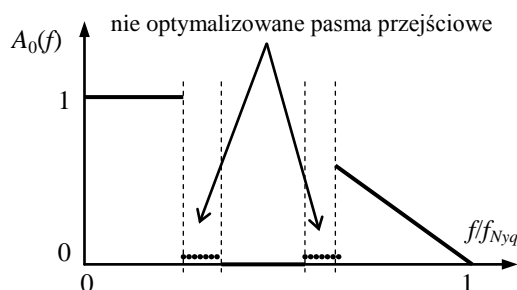
- Powtórzyć obliczenia zwiększając rząd filtra, np. do $N=40$ i zaobserwować różnice.

3.4. Projektowanie filtrów SOI metodą minimaxową

Do projektowania filtrów SOI o dowolnym kształcie charakterystyki metodą aproksymacji minimaxowych (minimalizacja maksymalnego błędu aproksymacji) Parks-McClellana służy funkcja:

```
B = firpm(N,F,M,W);
```

gdzie: N – rząd filtra (transmitancja ma $N+1$ współczynników), F – wektor częstotliwości punktów załamania charakterystyki (z zakresu $[0,1]$), $F=1$ oznacza częstotliwość Nyquista, M – wektor amplitud dla punktów załamania charakterystyki, W – opcjonalny wektor wag błędów aproksymacji w poszczególnych *pasmach* filtra (podaje się tylko jedną wartość wagi dla każdego z pasm).



Uwaga: Procedury `firpm` i `firls` przeprowadzają optymalizację tylko w pasmach pomiędzy elementami F i M o parzystym i nieparzystym indeksie (licząc od 1, por. rysunek). Pozostałe są traktowane jako nieoptymalizowane pasma przejściowe

Aproksymacja minimaxowa daje *jednakowe* pofalowania (*equiripple*) w całym paśmie.

- A.** Zaprojektować dolnoprzepustowy filtr *equiripple* spełniający wymagania sformułowane w poprzednim punkcie dla różnych rzędów N (np. $N=20, 40$ z nałożeniem wyników na jednym wykresie):

```

F = [0 0.3 0.35 1]; % wektor częstotl. optymalizowane pasma: 0-0.3 i 0.35-1
M = [1 1 0 0]; % wektor magnitud

disp('Filtr SOI equiripple Parks-McClellana')
Br1 = firpm(20,F,M) % filtr rzędu N=20 (stara funkcja remez(.))
[ Hr1,W ] = freqz(Br1,1,500); % wyzn. ch-ki częstotl. Uwaga: A=1
Br2 = firpm(40,F,M) % filtr rzędu N=40
[ Hr2,W ] = freqz(Br2,1,500);

subplot(211) % ch-ka amplitudowa
plot(F,M,':', W/pi,abs(Hr1),'m', W/pi,abs(Hr2),'r--'),
legend('H_0', 'N=20', 'N=40'),
ylabel('|H(\Omega)|'),
title('Filtr SOI, metoda equiripple')
axis([0,1,0,1.2]); grid on,

```



```
subplot(212) % ch-ka fazowa
phi1 = unwrap(angle(Hr1))*180/pi;
phi2 = unwrap(angle(Hr2))*180/pi;
plot(W/pi,phi1,'m', W/pi,phi2,'r--'),
grid on,
xlabel('\Omega / \pi'), ylabel('\phi(\Omega) (stopnie)')
```

- Zaobserwować jednakowy poziom pofalowań na całej szerokości pasm. Zanotować poziomy wzmocnienia na końcu pasma przepustowego i początku zaporowego (stromość w paśmie przejściowym) i porównać je z otrzymanymi w poprzednim punkcie metodą okienkową.

B. Zaprojektować filtry jednakowego rzędu zadając w jednym z nich dodatkowy wektor wag błędów: $W_g = [5 \ 1]$;

Podane wartości oznaczają, że waga błędu w pierwszym paśmie (przepustowym) jest 5 razy większa niż w drugim (zaporowym). Fragment charakterystyki w paśmie z większą wagą jest lepiej aproksymowany. Porównać przebiegi i błędy aproksymacji w obu przypadkach.

```
N = 20;
F = [0 0.3 0.35 1]; % wektor częstotl. optymalizowane pasma: 0-0.3 i 0.35-1
M = [1 1 0 0]; % wektor magnitud
```

```
disp('Filtr SOI equiripple Parks-McClellana')
Br1 = firpm(N,F,M)
[Hr1,W] = freqz(Br1,1,500); % wyzn. ch-ki częstotl. Uwaga: A=1
Wg = [5 1];
Br2 = firpm(N,F,M,Wg) % filtr z różnymi wagami błędu w pasmach
[Hr2,W] = freqz(Br2,1,500);
```

```
subplot(211) % ch-ka amplitudowa
plot(F,M,':', W/pi,abs(Hr1),'m', W/pi,abs(Hr2),'r--'),
legend('H_0','wagi [1 1]','wagi [5 1]'),
ylabel('|H(\Omega)|'),
title('Filtr SOI, metoda equiripple')
axis([0,1,0,1.2]); grid on,
subplot(212) % ch-ka fazowa
phi1 = unwrap(angle(Hr1))*180/pi;
phi2 = unwrap(angle(Hr2))*180/pi;
plot(W/pi,phi1,'m', W/pi,phi2,'r--'),
grid on,
xlabel('\Omega / \pi'), ylabel('\phi(\Omega) (stopnie)')
```

C. Zaprojektować filtr LP o zadanej poprzednio charakterystyce stosując metodę minimalizacji błędu *średniokwadratowego* (metodą najmniejszych kwadratów – *least squares* LS) w całym zakresie charakterystyki. Tego typu aproksymację wyznacza funkcja `firls(N,F,M,W)` o takich samych parametrach jak dla funkcji `firpm`. Porównać otrzymaną charakterystykę amplitudową z charakterystyką minimaxową (na jednym wykresie).

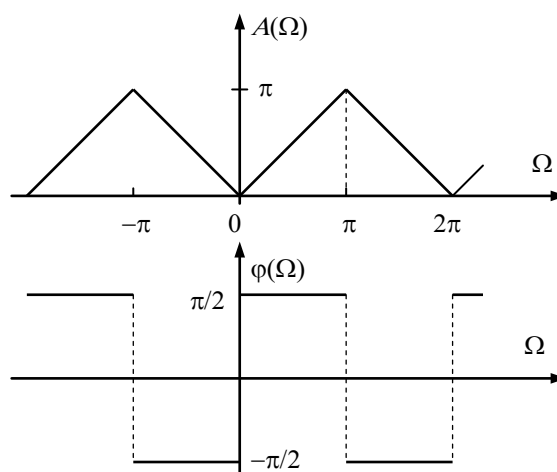
```
...
Br = firpm(N,F,M) % filtr equiripple
[Hr,W] = freqz(Br,1,500); % wyzn. ch-ki częstotl. Uwaga: A=1
Bls = firls(N,F,M) % filtr średniokwadratowy LS
[Hls,W] = freqz(Bls,1,500);

plot(F,M,':', W/pi,abs(Hr),'m', W/pi,abs(Hls),'r--'),
legend('H_0','filtr equiripple','filtr LS'),
xlabel('\Omega / \pi'), ylabel('|H(\Omega)|'),
axis([0,1,0,1.2]), grid on
```

- Zaobserwować i wyjaśnić różnice w poziomie dokładności aproksymacji w punktach granicznych pasm i wewnątrz pasm. Który filtr ma większy błąd maksymalny?

3.5. Filtr różniczkujący

W punkcie tym należy zaprojektować filtr dyskretny aproksymujący operację różniczkowania sygnału. Charakterystyki amplitudowa i fazowa idealnego filtra różniczkującego są pokazane na rysunku. Do aproksymacji tych charakterystyk w zakresie od $\Omega=0$ do $\Omega\approx\pi$ można wykorzystać procedurę Parks-McClellana projektowania filtrów SOI metodą minimaxową. Stałe w przybliżeniu przesunięcie fazowe uzyskuje się przyjmując *antysymetryczną* odpowiedź impulsową $h(n)=-h(N-n)$ z nieparzystym N , co eliminuje efekt Gibbsa w pobliżu $\Omega=\pi$. Wagi projektowe $1/\Omega$ dają małe błędy aproksymacji dla małych Ω .



- Zamknąć okna wykresów i wyczyścić przestrzeń roboczą (*Workspace*) i okno komend Matlaba.
- Zaprojektować metodą minimaxową filtr rzędu $N=3$ różniczkujący zadany sygnał wejściowy. Przeprowadzić obliczenia dla trzech sygnałów wejściowych: sygnału kosinusoidalnego, sygnału prostokątnego i sygnału trójkątnego.

% W tym miejscu należy wstawić linie z instrukcjami generowania jednego z sygnałów wejściowych
% filtra. Przykładowe sposoby generowania ciągów próbek sygnałów wejściowych są podane poniżej

```
N = 3; % rząd filtra
F=[0 0.95]; M=[0 1]; % domyślne Ho: wzmacnienie rośnie liniowo od 0 do 1
disp('Filtr różniczkujący SOI equiripple')
B = firpm(N,F,M,'differentiator') % nowa funkcja firpm(.) (Parks-McClellan)
[H,W] = freqz(B,1,256); % ch-ka częstotliwościowa
A = 5; % wsp. wzmacnienia sygnału wyjściowego y
y = A * filter(B,1,x); % filtracja (różniczkowanie) sygnału x

figure(1)
plot(0:length(x)-1,x,':', 0:length(x)-1,y,'r'), % wykresy x(n), y(n)
xlabel('n'), ylabel('x(n), y(n)')
title('Różniczkowanie sygnału wejściowego')
legend('sygnał wejściowy x', 'sygnał wyjściowy y')

figure(2) % nowe okno wykresu
subplot(211)
plot(F,M,':', W/pi,abs(H)) % ch-ka amplitudowa
ylabel('|H(\Omega)|'), grid on
axis([0 1 0 1.1])
title('Filtr różniczkujący SOI equiripple')
subplot(212)
phase=unwrap(angle(H))*180/pi; % ch-ka fazowa
plot(W/pi,phase), grid on
xlabel('\Omega / \pi'), ylabel('\phi(\Omega)')
```

Uwaga: wykresy z charakterystykami filtra dla danego rzędu N wystarczy zarejestrować jeden raz.

- Przykładowe sposoby generowania ciągów próbek sygnałów wejściowych do filtracji różniczkującej.

1) Sygnał kosinusoidalny:

```
f0 = 0.04; L = 100; % częstotliwość i długość syg. wejściowego
x = cos(2*pi*f0*[0:L-1]); % sygnał wejściowy kosinusoidalny
```

2) Sygnał prostokątny:

```
n0 = 50; nt = 2; % 2*no próbek/okres, nt okresów
```

```
x = [ones(1,n0) -ones(1,n0)]; % pół okresu +1 / pół okresu -1
xt = ones(nt,1); % nt okresów
x = (xt*x)'; % powielenie jednego okresu w wierszach macierzy
x = x(:); % ustaw. wierszy po kolei w wektorze, sygnał prostokątny
```

3) Sygnał trójkątny (liniowo zmienny):

```
x = (2*[0:n0-1]-n0)/n0; % 2*no próbek/okres, lin. narastanie
x = [x -x]; % pół okresu narastanie - pół okresu opadanie
xt = ones(nt,1); % nt okresów
x = (xt*x)'; % powielenie jednego okresu w wierszach macierzy
x = x(:); % ustaw. wierszy po kolei w wektorze, sygnał trójkątny
```

- Powtórzyć obliczenia dla filtra różniczkującego rzędu $N=15$ i trzech podanych sygnałów wejściowych.
- Przeanalizować uzyskane aproksymacje idealnej charakterystyki częstotliwościowej różniczkowania oraz przebiegi sygnałów wyjściowych, w tym stan nieustalony i opóźnienie sygnału na wyjściu filtra. Jak zmieniły się one po zmianie rzędu filtra?

3.6. Porównanie filtrów NOI i SOI

- Zaprojektować filtry pasmowozaporowe (BS) typu NOI i SOI dla charakterystyki zadanej $H_0(\Omega)$ z wąskimi pasmami przejściowymi i porównać aproksymacje charakterystykami filtrów różnego typu i o różnych wskaźnikach jakości optymalizacji.
 - 1) filtr NOI, metoda Yule-Walkera, $N_{yw}=20$,
 - 2) filtr SOI, metoda *least squares*, $N_{ls}=20$,
 - 3) filtr SOI *equiripple*, metoda minimaksowa Parksa-McClellana, $N_{pm}=20$.

```
Nyw=20; Nls=20; Npm=20; % rzędy filtrów
F = [0 0.28 0.3 0.5 0.52 1]; % wektor częstotl. unorm. do zakresu [0,1]
M = [1 1 0 0 1 1]; % zadany ch-ka magnitudy M(F)

[Byw,Ayw] = yulewalk(Nyw,F,M); % projektowanie H(z)=B(z)/A(z) metoda
Yule-Walkera
[Hyw,Wyw] = freqz(Byw,Ayw,500); % wyzn. ch-ki częstotliwościowej
Bls = firls(Nls,F,M); % filtr średniokwadratowy LS H(z)=B(z)
[Hls,W] = freqz(Bls,1,500); % wyzn. ch-ki częstotl. Uwaga: A=1
Bpm = firpm(Npm,F,M); % filtr minimaksowy (nowa funkcja firpm(.))
[Hpm,W] = freqz(Bpm,1,500);

subplot(211)
plot(F,M,':', W/pi,abs(Hyw), W/pi,abs(Hls),'r--', W/pi,abs(Hpm),'m-.')
legend('H_0','filtr NOI Yule-Walkera',Nyw,'filtr SOI LS',...
'filtr SOI minimax'),
ylabel('|H(\Omega)|'),
axis([0,1,0,1.4])
subplot(212)
Mdb = 20*log10(abs(M));
Hywdb = 20*log10(abs(Hyw));
Hlsdb = 20*log10(abs(Hls));
Hpmdb = 20*log10(abs(Hpm));
plot(F,Mdb,':', W/pi,Hywdb, W/pi,Hlsdb,'r--', W/pi,Hpmdb,'m-.')
xlabel('\Omega / \pi'), ylabel('|H(\Omega)| (dB)')
```

- Porównać błędy aproksymacji (średnie i maksymalne) charakterystyk amplitudowych filtrów tego samego rzędu wewnątrz i na granicach oraz stromości w pasmach przejściowych.
- Powtórzyć obliczenia zwiększając rzędy filtrów SOI do uzyskania porównywalnej zgodności aproksymacji charakterystyki zadanej.

3.7. Filtracja zakłóceń

Umiejętność projektowania filtrów zostanie wykorzystana w tym punkcie do wydzielenia sygnału użytecznego (w postaci sinusoidy lub sumy sinusoid) z sygnału zakłóconego wysokoczęstotliwościowym szumem losowym.

➤ Zamknąć okna wykresów i wyczyścić przestrzeń roboczą (*Workspace*) i okno komend Matlaba:

```
clear all, close all, clc
```

- Będziemy rozpatrywać prosty przypadek, kiedy pasmo szumu, którego widmo mocy jest określone przez pasmo przepustowe filtra „koloryzującego” HP i niezerowe tylko w górnym zakresie częstotliwości, *nie pokrywa się* z pasmem sygnału użytecznego. Sygnałem użytecznym jest suma dwóch sinusoid o częstotliwościach $f_1=0.1f_{Nyq}$ oraz $f_2=0.3f_{Nyq}$.

A. Generowanie próbek sygnału użytecznego i wysłanie go na wyjście audio.

(Uwaga: włączyć głośniki komputera).

```
L=8192;          % liczba próbek sygnału
d=200;          % fragment wykreślany na wykresach
fs=11025;       % częstotl. próbkowania w Hz (równa 1/4 * 44.1 KHz)
C=[0.5 0.5];    % wektor amplitud składowych harmoniczných
fn=[0.1 0.3];  % wektor częstotliwości unorm. f/fNyq = f/(fs/2)
W=pi*fn;        % unormowane pulsacje
% sygnał(n) = C(1)*sin(W(1)*n) + C(2)*sin(W(2)*n)
sygnał = C * sin(W' * [0:L-1]); % wykorzystanie mnożenia macierzowego
sound(sygnał,fs) % odtworzenie sygnału na wyjściu audio z częstotl. fs
```

B. Generowanie szumu zakłócenia. Sygnał zakłócenia powstaje w wyniku filtracji HP gaussowskiego szumu białego pokrywającego pełny zakres częstotliwości, w efekcie czego szum jest tłumiony w zakresie niższych częstotliwości.

```
szum = 2*randn(1,L); % gaussowski szum biały
Fs = [0 0.55 0.6 1]; % częstotliwości unorm. f/fNyq
Ms = [0 0 1 1]; % zadany wzmocnienie filtra HP koloryzowania szumu
Ns = 10; % rząd filtra szumu
Bs = firls(Ns,Fs,Ms); % filtr SOI LS szumu H(z)=B(z)
szum = filter(Bs,1,szum); % filtracja HP (koloryzowanie) szumu białego
```

C. Dodanie zakłócenia do sygnału użytecznego i wyznaczenie widma mocy $P_{xx}(\Omega)$ sygnału zakłóconego $x(n)$ i odtworzenie go na wyjściu audio.

```
x = sygnał + szum; % sygnał zaszumiony (np. pomiar)
sound(x,fs) % odtworzenie na wyjściu audio
% Wyzn. widma mocy Pxx sygnału zaszumionego (tematyka Ćwiczenia 3)
% w celu określenia potrzebnego pasma zaporowego filtra roboczego
[Pxx,w] = pwelch(x,512); % met. Welch, segmenty o dług. 512

figure(1) % Przebiegi sygnałów
subplot(211)
plot(sygnał(1:d))
ylabel('sygnał użyteczny'), xlabel('n')
subplot(212)
plot(x(1:d),'r')
ylabel('sygnał zaszumiony'),
figure(2)
plot(w/pi,Pxx,'r')
xlabel('\Omega / \pi'), ylabel('P_{xx}(\Omega)')
title('Widmo mocy sygnału zaszumionego')
```

➤ Na wykresie widma mocy sygnału zaszumionego zidentyfikować składowe sygnału użytecznego i zakres pulsacji unormowanych, w którym występuje szum zakłócenia. Zakres ten musi znajdować się w paśmie *zaporowym* filtra do usuwania zakłóceń.

D. Projektowanie filtra LP do usuwania zakłóceń. Uwaga: Podane poniżej wartości N, F, M to wartości wstępne, które należy zmodyfikować w celu uzyskania zadowalającego efektu filtracji (w szczególności: same jedynki w wektorze M oznaczają, że pasmo przepustowe obejmuje cały zakres częstotliwości - filtr będzie filtrem *wszechprzepustowym*)

```
F = [0 0.45 0.5 1]; % wektor częstotl. granic pasm
M = [1 1 1 1]; % wstępna ch-ka amplitudowa filtra
N=20; % rząd filtra
[B,A]=yulewalk(N,F,M); % wyzn.  $H(z)=B(z)/A(z)$  filtra NOI
[H,W] = freqz(B,A,500); % wyzn. ch-ki częstotl. filtra  $H(W)$ 

figure(3)
plot(F,M,':', W/pi,abs(H))
legend('H_0','filtr NOI Yule-Walkera'),
ylabel('|H(\Omega)|'), xlabel('\Omega / \pi'),
axis([0,1,0,1.2])
title('Charakterystyka amplitudowa filtra LP tłumiącego szum')
```

E. Filtrowanie sygnału zaszumionego zaprojektowanym filtrem i odtworzenie go na wyjściu audio. Zwrócić uwagę na początkowy *stan nieustalony* filtra w sygnale po filtracji.

```
xf = filter(B,A,x); % filtrowanie sygnału x
[Pxx,w] = pwelch(xf,512); % widmo mocy sygnału xf po filtracji
sound(xf,fs)
figure(4)
subplot(211)
plot(xf(1:d))
axis([0 d -1 1])
ylabel('sygnał po filtracji'), xlabel('n')
subplot(212)
plot(w/pi,Pxx)
xlabel('\Omega / \pi'), ylabel('P_{xx}(\Omega)')
title('Widmo mocy sygnału po filtracji')
```

- Zapamiętać sygnał uzyskany po filtracji filtrem NOI ($xfnoi = xf$;) i powtórzyć eksperyment filtracji zakłócenia za pomocą filtra SOI Parksa-McClellana (minimaxowego, o *liniowej fazie*) wyższego rzędu, np. :

```
N = 30; % rząd filtra SOI
B = firpm(N,F,M); % wyzn.  $H(z)=B(z)$  filtra SOI
A = 1;
[H,W] = freqz(B,A,500); % wyzn. ch-ki częstotl. filtra  $H(W)$ 
...
legend('H_0','filtr SOI minimaxowy'),
```

- Porównać uzyskane wyniki odtwarzając sygnały po filtracji jeden po drugim i skomentować ewentualne różnice. Uwaga: Nie rejestrować powtórnie wykresów, które pojawiły się w eksperymencie już wcześniej.

```
sound(xfnoi,fs) % zapamiętany efekt filtracji NOI
sound(xf,fs) % efekt filtracji SOI
```

- Zaprojektować filtr LP, którego zadaniem jest tłumienie usuwanie zakłócenia oraz składowej sygnału użytecznego o wyższej częstotliwości f_2 zawężając pasmo przepustowe filtra roboczego, np.:

```
F = [0 0.2 0.25 1];
M = [1 1 0 0];
```

3.8. Filtrowanie sygnału dźwiękowego

Zadanie polega na odfiltrowaniu z sygnału mowy wąskopasmowego zakłócenia w postaci składowej sinusoidalnej, której częstotliwość będzie określana na podstawie położenia jej prążka na

wykresie widma mocy sygnału z zakłóceniem. Znajomość częstotliwości zakłócenia umożliwi zaprojektowanie filtra pasmowozaporowego BS do wytłumienia zakłócenia.

➤ Zamknąć okna wykresów i wyczyścić przestrzeń roboczą (*Workspace*) i okno komend Matlaba.

- A.** Załadowanie sygnału akustycznego z zakłóceniem z pliku *.wav* o podanej nazwie do zmiennej Matlaba (sygnał 8-bitowy mono, częstotliwość próbkowania $f_s=11025\text{Hz}$), wykreślenie przebiegu, odtworzenie dźwięku i wyznaczenie widma mocy sygnału. (Uwaga: włączyć głośniki komputera).

```
[x,fs]=wavread('rec1'); % ładowanie sygnału z zakłóceniem, fs- cz. próbkow.
L=length(x)           % liczba próbek sygnału
d=5000;               % liczba próbek sygnału do wykreślania
% Wyzn. widma mocy sygnału zakłóconego do określenia częstotl. zakłócenia
[Pxx,w]=pwelch(x,1024); % met. Welch, segmenty o dług. 1024
title('Widmo mocy sygnału zakłóconego')
sound(x,fs)           % odtworzenie sygnału z zakłóceniem

figure(1)
subplot(211)
plot([0:d-1],x(1:d)), grid on
ylabel('sygnał zakłócony x(n)'), xlabel('n')
subplot(212)
plot(w/pi,10*log10(Pxx),'r'), grid on % skala log widma mocy (dB)
xlabel('\Omega / \pi'), ylabel('P_{xx}(\Omega) (dB)')
```

➤ Określić częstotliwość unormowaną prążka zakłócenia na wykresie widma mocy.

- B.** Zaprojektowanie filtra BS o wąskim paśmie zaporowym do usuwania wąskopasmowego zakłócenia na podstawie określonej częstotliwości zakłócenia.

```
% --- Projekt filtra BS ---
F=[0 0.27 0.28 0.32 0.33 1]; % wektor częstotl. granic pasm
M=[1 1 0 0 1 1]; % wektor wzmocnienia filtra BS
N=200; % rząd filtra
A=1; B=firpm(N,F,M); % filtr SOI H(z)=B(z)
% -----
[H,W]=freqz(B,A,500); % wyzn. ch-ki częstotl. filtra H(W)
figure(2)
plot(W/pi,abs(H)) % skala liniowa magnitudy
title('Wzmocnienie filtra BP SOI'),
ylabel('|H(\Omega)|'), xlabel('\Omega / \pi'),
axis([0,1,0,1.2]), grid on
title('Charakterystyka amplitudowa filtra BS do tłumienia zakłócenia')
```

➤ Sprawdzić, czy pasmo zaporowe filtra obejmuje częstotliwość zakłócenia.

- C.** Filtrowanie sygnału z zakłóceniem zaprojektowanym filtrem, wykreślenie przebiegu i widma mocy sygnału otrzymanego w wyniku filtracji i odtworzenie go na wyjściu audio.

```
xfilt=filter(B,A,x); % sygnał po filtracji
[Pxx,f]=pwelch(xfilt,1024); % widmo mocy sygnału po filtracji
sound(xfilt,fs) % odtworzenie sygnału po filtracji

figure(3)
subplot(211)
plot([0:d-1],xfilt(1:d),'c'), grid on
ylabel('sygnał x_{filt}(n) po filtracji'), xlabel('n')
subplot(212)
plot(w/pi,10*log10(Pxx),'m'), grid on % skala log widma mocy po
filtracji
xlabel('\Omega / \pi'), ylabel('P_{xx}(\Omega) (dB)')
title('Widmo mocy sygnału po filtracji')
```

- Ocenic efekty tłumienia zakłócenia. Przeprowadzić obliczenia z filtrem rzędu $N=100$ i porównać efekty.
 - Powtórzyć eksperyment dla innego pliku .wav (zawierającego sygnał z zakłóceniem harmonicznym) wskazanego przez prowadzącego. Na podstawie widma mocy wczytanego sygnału określić częstotliwość zakłócenia i odpowiednio zmodyfikować wektory F i M charakterystyki zadanej projektowanego filtra.
- D.** Przeprowadzić filtrowanie sygnału dźwiękowego z zakłóceniem harmonicznym za pomocą specjalnego wąskopasmowego filtra wycinającego (*notch filter*) NOI 2-go rzędu o transmitancji:

$$H(z) = \frac{b_0(1 - 2\cos\Omega_0 z^{-1} + z^{-2})}{1 - (1+r^2)\cos\Omega_0 z^{-1} + r^2 z^{-2}}, \quad b_0 = \frac{1+r^2}{2}$$

gdzie: Ω_0 – unormowana pulsacja środka zaporowego pasma wycięcia (filtr ma parę zespolonych sprzężonych zer położonych na okręgu jednostkowym ($r=1$) pod kątem $\pm\Omega_0$ względem dodatniej półosi rzeczywistej, dzięki którym wzmacnienie $|H(\Omega_0)|=0$), $0 < r < 1$ – odległość od początku układu współrzędnych pary biegunów rezonansowych zawężających pasmo wycięcia, b_0 – współczynnik skalujący wzmacnienie DC filtra do 1.

Filtry wycinające są używane do tłumienia zakłóceń o określonej częstotliwości, np. przydźwięku sieci zasilającej.

- Powtórzyć obliczenia z pkt. B i C z projektem filtra wycinającego. Sygnał dźwiękowy do filtracji zapisany w zmiennej x .

```
% ---Projekt filtra notch ---
F0=0.3;           % unormowana częstotl. zakłócenia (odczytana z wykresu Pxx)
W0=F0*pi;        % unormowana pulsacja wycięcia zaporowego
r=0.9;           % odległość biegunów od początku układu współrz. 0<r<1
b0=(1+r^2)/2;    % wsp. normalizujący H(W=0)=H(W=pi)=1
B=b0*[1 -2*cos(W0) 1]; A=[1 -(1+r^2)*cos(W0) r^2]; % filtr wycinający notch
% -----
...

```

- Ocenic „na ucho” efekty wycinania zakłócenia w porównaniu z filtrem zaprojektowanym w pkt. B.

3.9. Filtr o zerowym przesunięciu fazowym

Eksperyment polega na filtracji zaszumionego sygnału sinusoidalnego przy pomocy filtra nieprzyczynowego (przeprowadzającego inwersję sygnału w czasie i korzystającego w ten sposób z próbek z przyszłości względem chwili bieżącej) o zerowym przesunięciu fazowym.

- Zamknąć okna wykresów i wyczyścić przestrzeń roboczą (*Workspace*) i okno komend Matlaba.

A. Generowanie sygnału sinusoidalnego z dodanym szumem.

```
L=200;           % liczba próbek sygnału
n=[0:L-1];      % wektor indeksów
A=[1];         % wektor amplitud harmoniczných
W=pi*[0.02];   % wektor pulsacji unormowanych do WNyq
sinus = A * sin(W' * n); % składowa deterministyczna sygnału
x = sinus + 0.1*randn(1,L); % sygnał zakłócony szumem białym Gaussa

```

B. Projektowanie N -punktowego filtra uśredniającego L_p (wyznaczającego średnią arytmetyczną z N próbek sygnału) wygładzającego sygnał zaszumiony i przeprowadzenie zwykłej filtracji przyczynowej oraz filtracji nieprzyczynowej, a następnie przedstawienie na wykresach charakterystyki częstotliwościowej filtra przyczynowego i sygnałów po filtracji.

```
N=15;           % rząd filtra równy N-1
B=ones(1,N)/N; % N-punkt. filtr uśredniający, wszystkie współcz. = 1/N
[H,wh]=freqz(B,1); % ch-ka częstotliwościowa filtra
y=filter(B,1,x); % filtracja filtrem przyczynowym (normalna) H(z)=B(z)
yzero=filtfilt(B,1,x); % filtracja nieprzyczynowa |H(z)|^2

```

```

figure(1) % ch-ki częstotl. filtra
subplot(211)
plot(wh/pi,abs(H))
ylabel('|H(\Omega)|'),
title('Charakterystyki częstotl. filtra uśredniającego SOI')
subplot(212)
plot(wh/pi,unwrap(angle(H))*180/pi)
xlabel('\Omega / \pi'), ylabel('\phi(\Omega)')
figure(2) % przebiegi po filtracji
plot(n,x,'m', n,y,'r--', n,yzero,'b')
legend('sygnał zaszumiony','po filtracji przyczynowej',...
       'po filtracji nieprzyczynowej, \phi(\Omega)=0')
xlabel('n'), ylabel('x(n), y(n)')

```

- Porównać efekty filtracji nieprzyczynowej i normalnej. Sprawdzić, czy przesunięcie fazowe dla filtracji normalnej zgadza się z wartością określoną przez charakterystykę fazową filtra dla zadanej pulsacji unormowanej Ω sygnału użytecznego.

4. Zadania do realizacji samodzielnej

Zadanie 4.1. Napisać własną funkcję filtracji sygnału zapisanego w wektorze x przez filtr dyskretny o współczynnikach licznika i mianownika transmitancji zapisanych w wektorach b i a . Wartości sygnału wyjściowego filtra mają być zapisane w wektorze y . Funkcja ma odpowiadać funkcji wbudowanej Matlaba `filter`. Algorytm jest opisany w podpunkcie **Algorithms** dokumentacji **Help** funkcji `filter`.

5. Opracowanie sprawozdania

W sprawozdaniu należy zawrzeć zarejestrowane wyniki eksperymentów obliczeniowych przeprowadzonych według instrukcji oraz zadanych przez prowadzącego zadań do realizacji samodzielnej z odpowiednimi opisami oraz wyjaśnieniami opartymi na wiedzy z wykładu i literatury, ze szczególnym uwzględnieniem problemów wskazanych symbolem ➤.

Literatura

1. Ingle V.K., Proakis J.G.: *Essentials of Digital Signal Processing Using Matlab*, 3 ed., Cengage Learning, 2012.
2. Mitra S.K.: *Digital Signal Processing. A Computer Based Approach*, 4 ed., McGraw-Hill, 2011.
3. Leis J.W.: *Digital Signal Processing Using Matlab for Students and Researchers*, John Wiley & Sons, 2011.
4. Orfanidis S.J.: *Introduction to Signal Processing*, Prentice Hall, 2010.
5. Zieliński T.: *Cyfrowe przetwarzanie sygnałów. Od teorii do zastosowań*, WKiŁ, 2005.
6. Smith S.W.: *Cyfrowe przetwarzanie sygnałów. Praktyczny poradnik dla inżynierów i naukowców*, Wyd. BTC, 2007.
7. Lyons R.G.: *Wprowadzenie do cyfrowego przetwarzania sygnałów*, WKiŁ, 1999.
8. Chaparro L.F.: *Signals and Systems Using Matlab*, Academic Press, 2011.

Opracował: Dr inż. Janusz Baran

Częstochowa, 1999-2013