Projektowanie filtrów cyfrowych NOI na podstawie charakterystyk prototypów analogowych

System (filtr) w dziedzinie czasu dyskretnego można opisać za pomocą równania różnicowego:

$$y(n) + a_1 y(n-1) + \cdots + a_N y(n-N) = b_0 x(n) + b_1 x(n-1) + \cdots + b_L x(n-L) \qquad (1)$$

gdzie:

n – oznacza kolejne chwile czasu (odpowiadające chwilom $t = nT$, gdzie T – okres próbkowania);
y(n) – wartość sygnału na wyjściu systemu w chwili *n*;
x(n) – wartość sygnału na wejściu systemu w chwili *n*;
N – maksymalna ilość opóźnień sygnału na wyjściu;
L – maksymalna ilość opóźnień sygnału na wejściu;
$a_1, a_2, \ldots, a_N$ – współczynniki modyfikujące wartości sygnału z wyjścia;
$b_1, b_2, \ldots, b_L$ – współczynniki modyfikujące wartości sygnału z wejścia;

Równanie (1) opisuję klasę systemów dyskretnych nazwanych *systemami rekursywnymi*, ponieważ sygnał na wyjściu zależy zarówno od sygnału na wejściu jak i od poprzednich wartości na wyjściu. Inną klasą systemów są *systemy nierekursywne*, w których sygnał na wyjściu zależy jedynie od sygnału na wejściu (dla tych systemów współczynniki $a_1, a_2, \ldots, a_N$ równania różnicowego są równe zero). *Rząd* filtru rekursywnego jest równy liczbie opóźnień *N* (bez względu na liczbę opóźnień *L*). System opisany równaniem (1) posiada transmitancję *H(z)*, która jest również transformatą *z* odpowiedzi impulsowej filtru *h(n)*.

$$H(z) = \frac{\sum_{l=0}^{L} b_l z^{-l}}{1 + \sum_{m=1}^{N} a_m z^{-m}} \qquad (2)$$

Równanie (2) opisuje transmitancję filtru NOI (filtru o **N**ieskończonej **O**dpowiedzi **I**mpulsowej) (IIR – **I**nfinite **I**mpulse **R**esponse). System jest *stabilny* w sensie *OWOW* (Ograniczone Wejście Ograniczone Wyjście) jeżeli przy ograniczonym sygnale na wejściu systemu, sygnał na wyjściu jest również ograniczony. System przyczynowy jest stabilny jeżeli wszystkie bieguny transmitancji *H(z)* (pierwiastki mianownika równania (2)) znajdują się wewnątrz okręgu jednostkowego na płaszczyźnie zmiennej zespolonej *z*.

Transmitancja filtru analogowego $H_a(s)$ opisana jest wzorem:

$$H_a(s) = \frac{\sum_{k=0}^{M} c_k s^k}{s^N + \sum_{i=0}^{N-1} d_i s^i}, \qquad (3)$$

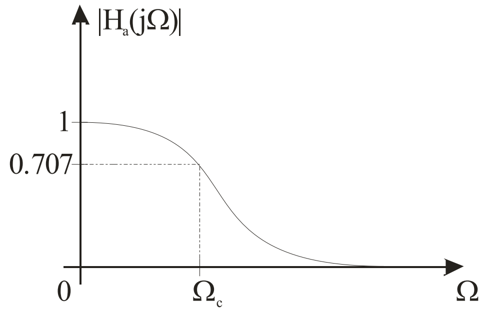
$$\text{jeżeli } s = j\Omega \text{ to } H_a(j\Omega) = \frac{\sum_{k=0}^{M} c_k (j\Omega)^k}{(j\Omega)^N + \sum_{i=0}^{N-1} d_i (j\Omega)^i} \qquad (4)$$

gdzie Ω [rad/s] oznacza pulsację dla systemu analogowego (określonego dla chwil czasu będących zmienną ciągłą). Wzór (4) określa odpowiedź częstotliwościową $H_a(j\Omega)$ filtru.

Analogowy dolnoprzepustowy filtr Butterwortha rzędu N można opisać za pomocą kwadratu amplitudy odpowiedzi częstotliwościowej:

$$\left|H_a(j\Omega)\right|^2 = \frac{1}{1+\left(\dfrac{j\Omega}{j\Omega_c}\right)^{2N}} \tag{5}$$

gdzie Ω_c jest 3 dB częstotliwością odcięcia.

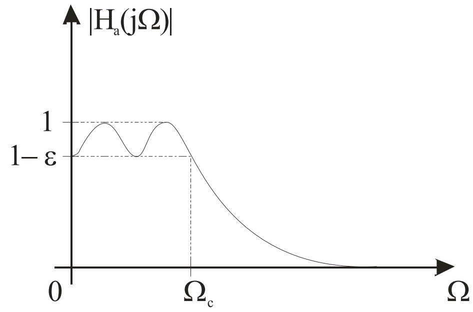


Analogowy filtr Czebyszewa I-ego rodzaju charakteryzuje się równomiernie falistym przebiegiem modułu charakterystyki częstotliwościowej w pasmie przenoszenia. Dolnoprzepustowy filtr Czebyszewa I-ego rodzaju rzędu N można opisać za pomocą kwadratu amplitudy odpowiedzi częstotliwościowej:

$$\left|H_a(\Omega)\right|^2 = \frac{1}{1+\varepsilon^2 V_N^2\left(\dfrac{\Omega}{\Omega_c}\right)} \tag{6}$$

gdzie Ω_c jest częstotliwością odcięcia, ε określa falistość charakterystyki, a $V_N(x)$ jest wielomianem Czebyszewa N-tego rzędu:

$$V_N(x) = \cos(N \cdot \arccos x) \tag{7}$$



Analogowy filtr Czebyszewa II-ego rodzaju charakteryzuje się równomiernie falistym przebiegiem modułu charakterystyki częstotliwościowej w pasmie zaporowym.

Aby otrzymać odpowiedź częstotliwościową filtru górnoprzepustowego GP (HP – highpass), pasmowo-przepustowego PP (BP – bandpass) lub pasmowo-zaporowego PZ (BS – bandstop) mając odpowiedź filtru dolnoprzepustowego DP (LP – lowpass) należy dokonać podstawienia:

DP → DP	$s \rightarrow \frac{\Omega_o}{\Omega_d} s$	(A)
DP → GP	$s \rightarrow \frac{\Omega_o \Omega_d}{s}$	(B)
DP → PP	$s \rightarrow \frac{\Omega_o (s^2 + \Omega_l \Omega_u)}{(\Omega_u - \Omega_l)s}$	(C)
DP → PZ	$s \rightarrow \frac{\Omega_o (\Omega_u - \Omega_l)s}{s^2 + \Omega_u \Omega_l}$	(D)

gdzie Ω_o jest częstotliwością odcięcia prototypowego filtru DP (najczęściej korzysta się z filtrów znormalizowanych $\Omega_o = 1$), natomiast Ω_d jest częstotliwością odcięcia projektowanego filtru (DP lub GP). Parametry Ω_l oraz Ω_u oznaczają odpowiednio częstotliwości odcięcia niższą i wyższą dla projektowanego filtru PP lub PZ.

Aby zaprojektować filtr cyfrowy korzystając z prototypu analogowego musi być określona wzajemna zależność pomiędzy zmiennymi *s* i *z*. Umożliwia to przekształcenie biliniowe:

$$s = m(z) = \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \tag{8}$$

$$z = m^{-1}(s) = \frac{1 + \frac{sT}{2}}{1 - \frac{sT}{2}} \tag{9}$$

Filtr cyfrowy *H(z)* jest związany ze swym analogowym prototypem $H_a(s)$ poprzez przekształcenie biliniowe:

$$H(z) = H_a(m(z)) \tag{10}$$

Przekształcenie zachowuje rząd i stabilność filtru. Częstotliwości analogowe Ω związane są z częstotliwościami cyfrowymi θ przez równania: (*T* okres próbkowania)

$$\Omega = \frac{2}{T} tan\left(\frac{\theta}{2}\right) \tag{11}$$

$$\theta = 2 \cdot tan^{-1}\left(\frac{\Omega T}{2}\right) \tag{12}$$

Procedura projektowania filtru NOI

1. Określenie rzędu filtru *N*, częstotliwości granicznej θ, częstotliwości próbkowania;
   `(N = 2; f = 8000; teta = 0.2;)`
2. Przekształcenie częstotliwości cyfrowych θ na częstotliwości analogowe Ω za pomocą wzoru (11) (w zależności od typu filtru DP, PP mamy jedną lub dwie częstotliwości do obliczenia);
   `(omega = 2*f*tan(pi*teta/2);)`
3. znalezienie znormalizowanego prototypu analogowego filtru (Z-zera, P-bieguny, K-wzmocnienie) (funkcje *buttap*, *cheb1ap*, *cheb2ap*)
   `([Z,P,K] = buttap(N); figure(1);zplane(Z,P);)`
4. Znalezienie transmitancji filtru znormalizowanego
   `([NUM,DEN] = zp2tf(Z,P,K);)`
   oraz denormalizacja względem częstotliwości granicznych za pomocą wzoru:
   (A) DP→DP `([NUMT,DENT] = lp2lp(NUM,DEN,omega);)`
   (B) DP→GP `([NUMT,DENT] = lp2hp(NUM,DEN,omega);)`
   (C) DP→PP `([NUMT,DENT] =`
   `= lp2bp(NUM,DEN, omegaH-omegaL, omegaL+Bw/2);)`
   (D) DP→PZ `([NUMT,DENT] =`
   `= lp2bs(NUM,DEN, omegaH-omegaL, omegaL+Bw/2);)`
   `(figure(2); freqs(NUMT,DENT);)`
5. Znalezienie transmitancji filtru cyfrowego $H(z)$ za pomocą przekształcenia biliniowego wzór (8)
   ```
   ([BB,AA] = bilinear(NUMT,DENT,f);
    figure(3); freqz(BB,AA);)
   ```

Matlab umożliwia także obliczanie szukanej transmitancji filtru cyfrowego za pomocą tylko jednej funkcji:

(A) LP→LP `([BBB,AAA] = butter (N,teta);)`
(B) LP→HP `([BBB,AAA] = butter (N,teta,'high');)`
(C) LP→BP `([NUMT,DENT] = butter (N,[tetaL tetaH]);)`
(D) LP→BS `([NUMT,DENT] =`
`= butter(N,[tetaL tetaH],'stop');)`

```
([BBB,AAA] = butter (N,teta); figure; freqz(BBB,AAA); )
```

Zadania

1. Zaprojektować cyfrowy filtr dolnoprzepustowy NOI korzystając z prototypów dolnoprzepustowych analogowych filtrów Butterwortha oraz Czebyszewa I i II-ego rodzaju. Parametry filtru: rząd $N = 10$, częstotliwość graniczna $f_d = 1600$ Hz, częstotliwość próbkowania $f_s = 16000$ Hz. Zaobserwować charakterystyki amplitudowe i fazowe filtrów analogowych (funkcja *freqs*) oraz cyfrowych (funkcja *freqz*). Wykonać obliczenia postępując według procedury projektowania filtrów NOI, a następnie porównać wyniki z działaniem funkcji *butter*, *cheby1*, *cheby2*. Porównać charakterystyki filtrów uzyskanych z wykorzystaniem różnych prototypów.
2. Zaprojektować cyfrowy filtr górnoprzepustowy NOI korzystając z prototypów dolnoprzepustowych analogowych filtrów Butterwortha oraz Czebyszewa I i II-ego rodzaju. Parametry filtru: rząd $N = 10$, częstotliwość graniczna $f_g = 1600$ Hz, częstotliwość próbkowania $f_s = 16000$ Hz. Zaobserwować charakterystyki amplitudowe i fazowe filtru analogowego (funkcja *freqs*) oraz cyfrowego (funkcja *freqz*). Wykonać obliczenia postępując według procedury projektowania filtrów NOI, a następnie porównać wyniki z działaniem funkcji *butter*, *cheby1*, *cheby2*.
3. Przy użyciu zaprojektowanych filtrów dokonać filtracji (funkcja *filter*) sygnału będącego sumą trzech sinusoid o częstotliwościach: 800 Hz, 2000 Hz, 4500 Hz. Zaobserwować sygnały na wejściu i wyjściu filtru. Porównać widmo sygnałów przed i po filtracji.
4. Porównać właściwości filtrów NOI z filtrami SOI.

Matlab – użyteczne funkcje: *filter, bilinear, zp2tf, zplane, buttap, butter, cheb1ap, cheby1, cheb2ap, cheby2, freqs, freqz, lp2lp, lp2hp, lp2bp, lp2bs*