

Ćwiczenie 4

Stałoprzecinkowa filtracja IIR (NOI)

Wprowadzenie

Celem tego ćwiczenia jest zaprojektowanie przy pomocy pakietu Matlab filtrów o nieskończonej odpowiedzi impulsowej, a następnie zaimplementowanie ich i uruchomienie w czasie rzeczywistym na procesorze SHARC. Część druga, to znaczy implementacja stałoprzecinkowa, z dodatkowym ograniczeniem modułu współczynników do zakresu $(-1, 1)$ wprowadza pewne komplikacje, które inżynier musi pokonać.

Podstawy teoretyczne

Postać ogólna filtru IIR reprezentowana jest następującym wzorem:

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

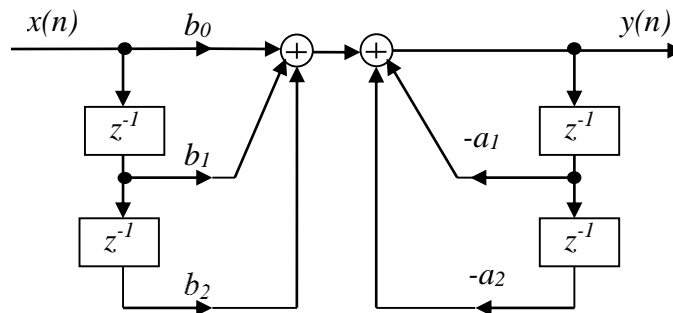
Postać ta jest mało przydatna w implementacji na mikroprocesorze z uwagi na konieczność wykonywania operacji dzielenia, której efektywność (czas wykonania) jest bardzo niska.

Warto zauważyć, że w wypadku kiedy współczynniki w mianowniku (a_m) są równe zeru otrzymujemy równanie filtru FIR implementowanego w poprzednim ćwiczeniu.

Wstawiając ten wzór do równania opisującego zależność dyskretnego sygnału cyfrowego na wyjściu filtru, a następnie przemnażając obie strony tego równania przez mianownik $H(z)$ i rozwijając sumy otrzymujemy tak zwane równanie różnicowe będące pierwszą formą bezpośredniej realizacji filtru IIR (ang. direct form I realization):

$$y(n) = b_0 x(n) + b_1 x(n-1) + \dots + b_{L-1} x(n-L+1) - a_1 y(n-1) - a_2 y(n-2) - \dots - a_M y(n-M)$$

Schemat blokowy tej formy realizacji dla filtru drugiego rzędu ($L, M=2$) przedstawiony jest na rysunku poniżej.



Rysunek 1 Pierwsza forma realizacji filtru IIR.

Dokonując kolejnych przekształceń zależności w filtrze IIR otrzymujemy tak zwaną drugą postać kanoniczną:

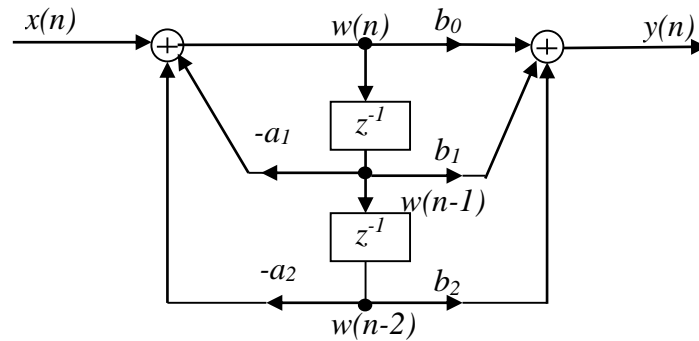
$$w(n) = x(n) - a_1 w(n-1) - \dots - a_M w(n-M)$$
$$y(n) = b_0 w(n) + b_1 w(n-1) + \dots + b_{L-1} w(n-L+1)$$

Schemat blokowy tej formy realizacji przedstawia rysunek 2.

Projektowanie w Matlab-ie filtrów IIR składa się z następujących kroków:

1. Określamy rząd filtru potrzebny do osiągnięcia zakładanych parametrów filtru (głównie stromości zboczy charakterystyki częstotliwościowej). Służą do tego następujące funkcje:

Ćwiczenie 4
Stałoprzecinkowa filtracja IIR (NOI)



Rysunek 2 Druga forma realizacji filtra IIR drugiego rzędu.

```
[N, Wn]=buttord(Wp, Ws, Rp, Rs) ;
[N, Wn]=cheblord(Wp, Ws, Rp, Rs) ;
[N, Wn]=cheb2ord(Wp, Ws, Rp, Rs) ;
[N, Wn]=ellipord(Wp, Ws, Rp, Rs) ;
```

Parametry W_p i W_s są znormalizowanymi częstotliwościami brzegowymi pasma przepustowego i zaporowego, a R_p i R_s są wyrażonymi w decybelach zafalowaniach charakterystyki w tych pasmach (ang. ripple).

2. Mając określony rząd i wektor częstotliwości W_n zawierający charakterystykę częstotliwościową filtra możemy przystąpić do wyliczenia współczynników wielomianów licznika i mianownika:

```
[b, a]=butter(N, Wn) ;
[b, a]=cheby1(N, Wn) ;
[b, a]=cheby2(N, Wn) ;
[b, a]=ellip(N, Wn) ;
```

Oczywiście, każda z linijek dotyczy projektowania jednego z czterech podstawowych typów filtrów: Butterworth-a, Chebyszewa 1 i 2 oraz eliptycznego.

Bezpośrednia implementacja postaci I i II w asemblerze stałoprzecinkowego procesora sygnałowego napotyka na problem nieunormowanych do zakresu akceptowalnego przez te procesory przedziału wartości $[-1, 1]$ współczynników, szczególnie mianownika (wektor a).

Problem ten można przezwyciężyć na kilka sposobów, z których pierwszy polega na unormowaniu współczynników:

$$y(n) = b_0 x(n) + b_1 x(n-1) + \dots + b_{L-1} x(n-L+1) - \left[\frac{a_1}{2^k} y(n-1) + \frac{a_2}{2^k} y(n-2) + \dots + \frac{a_M}{2^k} y(n-M) \right] 2^k$$

Unormowanie, jak widać powyżej, polega na przeskalowaniu (podzieleniu) na czas obliczeń współczynników (np. a) potęgą cyfry 2, a po zakończeniu obliczeń (splotu z tymi współczynnikami) przeskalowaniu wstecz (pomnożenie przez taką samą potęgę cyfry 2).

Drugim sposobem, jest zamiana postaci filtra z wielomianu na iloczyn jednomianów, które zawierają zera i bieguny transmitancji filtra:

```
[Z, P, K]=tf2zp(b, a) ;
```

(ang. TransFer_to_Zeroes_and_Poles). Otrzymane zera (Z) i bieguny (P , K -wsp. wzmocnienia) mogą być liczbami zespolonymi, dlatego możemy dokonać kolejnego przekształcenia do postaci kaskady dwumianów o rzeczywistych współczynnikach:

```
[sos, G]=zp2sos(Z, P, K) ;
```

(ang. Zero_and_Poles_to_Second_Order_Sections). Macierz sos zawiera trójki współczynników dwumianów licznika i mianownika.

Ćwiczenie 4

Stałoprzecinkowa filtracja IIR (NOI)

Program ćwiczenia:

1. Korzystając z programu filtracji próbkowej dla filtru FIR, należy zmodyfikować część napisaną w asemblerze, aby uzyskać filtrację IIR.
2. Program `fdatool` pozwala nam projektować całe spektrum filtrów IIR, ale, w odróżnieniu od filtrów FIR możemy wybrać do implementacji różne struktury(formy):
 - a. Single section
 - b. Direct-form I Second-Order Sections
 - c. Direct-form II Second-Order Sections

Domyślną strukturą jest wersja c), ale można ją zmienić naciskając nad jej nazwą prawy przycisk myszy. Można wtedy wybrać również 'Show filter structure', co pozwoli zobaczyć jak wygląda realizacja filtru i związek z nią otrzymanych współczynników $b(n)$ i $a(n)$. Struktura a) (jedna sekcja) daje nam TYLKO współczynniki wielomianu licznika i mianownika, gdzie często współczynniki mianownika są poza zakresem liczb stałoprzecinkowych (wartość bezwzględna większa od 1.0). Rozwiązaniem jest przeskalowanie przed obliczeniami tych współczynników, a następnie przeskalowanie wstecz wyniku, przed użyciem go w równaniu różnicowym.

Pozostałe struktury (b i c) posiadają znormalizowane współczynniki (czasem do 2.0, wtedy też je trzeba przeskalować), ale za to normalizacja jest ukryta we współczynniku wzmocnienia wejściowego (patrz widok struktury formy filtru).

Pytaniem o wnioski z tej części ćwiczenia jest powód wyboru przez `fdatool` jako domyślnej II formy SOS, oraz jak skalować współczynniki $a(n)$, aby zachować maksymalną precyzję obliczeń, a tym samym stabilność filtra ?

3. Modyfikacje programu bazowego FIR wykonuj iteratywnie, kompilując zmiany, aż do skutku, którym jest poprawne działanie filtru IIR obserwowane w programie Spectralab. Obserwacje w Spectralab-ie można poszerzyć o obserwację nieliniowej fazy w funkcji częstotliwości. W tym celu na jeden kanał należy podać sygnał wejściowy, a na drugi wyjściowy filtru IIR, którym powinien być sinus o wolno zmieniającej się częstotliwości w zakresie pasma przepustowego filtra. Najsilniejsze nieliniowości występują na granicy pasma przepustowego i zaporowego, które można obserwować włączając widok przebiegów czasowych i śledząc wzajemne położenie sinusoid z wejścia i wyjścia, lub, po prostu wybierając widok fazy ('Phase').