

Cyfrowa implementacja analogowego filtra dolnoprzepustowego I rzędu.

Na rys.1 przedstawiono filtr dolnoprzepustowy zbudowany w oparciu o dyskretny element R i C oraz jego odpowiednik przedstawiony za pomocą impedancji tych elementów (1b).

Sygnał wyjściowy tego filtra można wyznaczyć metodą oczkową (II prawo Kirchhoffa) lub operatorową traktując układ jako dzielnik impedancyjny.

Wszystkie sygnały należy traktować jako wielkości zależne od czasu ($u_2(t)$, $y(t)$ itp.). Dla uproszczenia analizy można przyjąć warunki początkowe równe zero.



Rys.1. Dolnoprzepustowy filtr analogowy I rzędu.

- a) realizacja analogowa typu RC
- b) dzielnik impedancyjny

$$\begin{aligned}
 u_1 &= u_R + u_C \\
 u_1 &= i_C \cdot R + u_C \\
 \left\{ \begin{aligned} u_2 &= u_C \\ i_C &= C \cdot \frac{du_C}{dt} = C \cdot \frac{du_2}{dt} \end{aligned} \right\} \\
 u_1 &= C \cdot \frac{du_2}{dt} \cdot R + u_2 \\
 \frac{du_2}{dt} &= (u_1 - u_2) \cdot \frac{1}{RC} \\
 y &= x \cdot \frac{Z_C}{Z_R + Z_C} \\
 y &= x \cdot \frac{\frac{1}{\omega \cdot C}}{R + \frac{1}{\omega \cdot C}} \\
 y &= x \cdot \frac{1}{1 + \omega \cdot RC} \\
 \left\{ \begin{aligned} \omega &\rightarrow s \\ s &- \text{operator Laplace'a} \end{aligned} \right\} \\
 y &= x \cdot \frac{1}{1 + s \cdot RC} \\
 y \cdot s &= (x - y) \cdot \frac{1}{RC} \\
 \left\{ \begin{aligned} &\text{dokonujemy odwrotnego} \\ &\text{przekształcenia Laplace'a} \end{aligned} \right\} \\
 \frac{dy}{dt} &= (x - y) \cdot \frac{1}{RC}
 \end{aligned} \tag{1}$$

W obu przypadkach otrzymujemy tożsame równania różniczkowe na pochodną sygnału wyjściowego o postaci:

$$\frac{dy}{dt} = (x - y) \cdot \frac{1}{T} \quad (2)$$

gdzie T oznacza stałą czasową filtru:

$$T = R \cdot C \quad (3)$$

stąd 3-decybelowa częstotliwość graniczna f_g jest równa:

$$f_g = \frac{1}{2 \cdot \pi \cdot RC} = \frac{1}{2 \cdot \pi \cdot T} \quad (4)$$

W celu cyfrowej (dyskretnej) realizacji wyliczenia sygnału wyjściowego filtru należy równanie różniczkowe (2) przekształcić w równanie różnicowe:

$$dy[n] = (x[n] - y[n]) \cdot \frac{1}{T} \quad (5)$$

Gdzie:

n - kolejny krok akwizycji danych

$dy[n]$ - numeryczne wyliczenie pochodnej sygnału wyjściowego dy/dt – wzór (2)

$x[n]$ - obecna wartość próbki sygnału odczytana z przetwornika ADC

$y[n]$ - wartość sygnału wyjściowego filtru obliczona w danym kroku

a następnie scałkować.

Istnieje wiele algorytmów całkowania numerycznego. Różnią się one przede wszystkim czasochłonnością (ilością operacji) i dokładnością aproksymacji całki. W systemach czasu rzeczywistego wybór metody jest kompromisem pomiędzy możliwościami obliczeniowymi, a wymaganą dokładnością. Dokładność obliczeń można zwiększyć dwoma sposobami: wybierając dokładniejszą aproksymację (bardziej złożony algorytm) lub zmniejszając czas próbkowania (dyskretyzacji).

Najprostsza aproksymacja całki nazywana metodą Eulera ma postać następującą:

$$y[n] = y[n - 1] + dy[n] \cdot \Delta t \quad (6)$$

Δt – okres próbkowania

Podstawiając równanie (5) do (6) otrzymamy:

$$y[n] = y[n - 1] + (x[n] - y[n]) \cdot \frac{\Delta t}{T} \quad (7)$$

Przekształcając je dalej otrzymujemy:

$$\begin{aligned}
 y[n] &= y[n-1] + x[n] \cdot \frac{\Delta t}{T} - y[n] \cdot \frac{\Delta t}{T} \\
 y[n] + y[n] \cdot \frac{\Delta t}{T} &= x[n] \cdot \frac{\Delta t}{T} + y[n-1] \\
 y[n] \cdot (1 + \frac{\Delta t}{T}) &= x[n] \cdot \frac{\Delta t}{T} + y[n-1] \\
 y[n] &= x[n] \cdot \frac{\Delta t}{T + \Delta t} + y[n-1] \cdot \frac{T}{T + \Delta t}
 \end{aligned} \tag{8}$$

Podstawiając:

$$\alpha = \frac{\Delta t}{T + \Delta t} \tag{9}$$

Otrzymujemy końcową postać wartości sygnału na wyjściu cyfrowego filtra dolnoprzepustowego:

$$y[n] = x[n] \cdot \alpha + y[n-1] \cdot (1 - \alpha) \tag{10}$$

Z kolei podstawiając:

$$\beta = \frac{T + \Delta t}{\Delta t} = \frac{1}{\alpha} \tag{11}$$

otrzymujemy trochę inną postać filtra. Ponieważ zawsze $\beta > 1$, to taka forma ułatwia realizację filtra za pomocą liczb całkowitych:

$$y[n] = \frac{x[n] + y[n-1] \cdot (\beta - 1)}{\beta} \tag{12}$$

Powyższa realizacja sprzętowa filtra cyfrowego dostępna dla użytkownika zaimplementowana jest np. w sensorach firmy Bosch, BMP280 i innych.

Wykorzystując zależność (4) można wyrazić częstotliwość graniczną filtra za pomocą współczynników α lub β :

$$f_g = \frac{1}{2 \cdot \pi \cdot T} = \frac{\alpha}{2 \cdot \pi \cdot \Delta t \cdot (1 - \alpha)} = \frac{1}{2 \cdot \pi \cdot \Delta t \cdot (\beta - 1)} \tag{13}$$

Zależność (10) prawdziwa jest jednak tylko dla $\alpha \leq 0.1$. W przeciwnym razie charakterystyka przenoszenia filtru cyfrowego odbiega od charakterystyki filtru analogowego dla tej samej wartości stałej czasowej T i częstotliwości próbkowania – wykażemy to dalej.

Można starać się analitycznie wyznaczyć te różnice, ale prościej jest w tym celu użyć metod numerycznych ponieważ wynik końcowy zawsze zależy od przyjętej metody aproksymacji całki, a przez to od struktury filtru.

Poniżej wyznaczmy charakterystykę filtru analogowego i odpowiadającego mu filtru cyfrowego o strukturze opisanej równaniem (10).

Równanie (10) przekształcamy w ten sposób, że po lewej stronie równania grupujemy wyrazy z 'y', a po prawej wyrazy z 'x' do postaci:

$$y[n] - y[n-1] \cdot (1 - \alpha) = x[n] \cdot \alpha \quad (14)$$

Dokonyjemy obustronnego przekształcenia z powyższego równania i zapisujemy w postaci transmitancji:

$$\begin{aligned} Z[y[n]] - Z[y[n-1]] \cdot (1 - \alpha) &= Z[x[n]] \cdot \alpha \\ Y[z] - Y[z] \cdot z^{-1} \cdot (1 - \alpha) &= X[z] \cdot \alpha & \{Z[y[n-k]] \rightarrow Y[z] \cdot z^{-k}\} \\ Y[z] \cdot (1 - z^{-1} \cdot (1 - \alpha)) &= X[z] \cdot \alpha \\ \frac{Y[z]}{X[z]} &= \frac{\alpha}{1 - z^{-1} \cdot (1 - \alpha)} \end{aligned} \quad (15)$$

Pomijając błędy obliczeń wynikające z różnej reprezentacji liczbowej danych (liczby całkowite, zmiennoprzecinkowe, błędy zaokrągleń), błąd aproksymacji wynikający tylko z przyjętej struktury filtru można oszacować poprzez porównanie jego charakterystyki amplitudowo-częstotliwościowej dla przyjętego okresu próbkowania Δt z charakterystyką ciągłego filtru analogowego.

Charakterystykę amplitudowo-częstotliwościową (a/f) filtru analogowego otrzymujemy podstawiając :

$$s \rightarrow j\omega \quad (16)$$

Stąd równanie filtru analogowego:

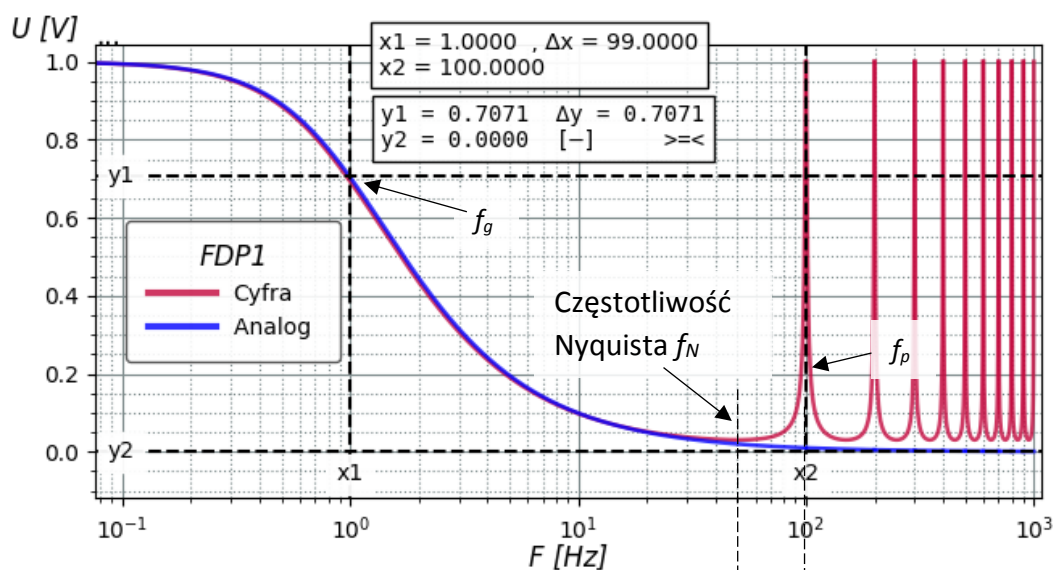
$$H_A(j \cdot \omega) = \frac{1}{1 + j\omega T} \quad (17)$$

Natomiast dokonując podstawienia:

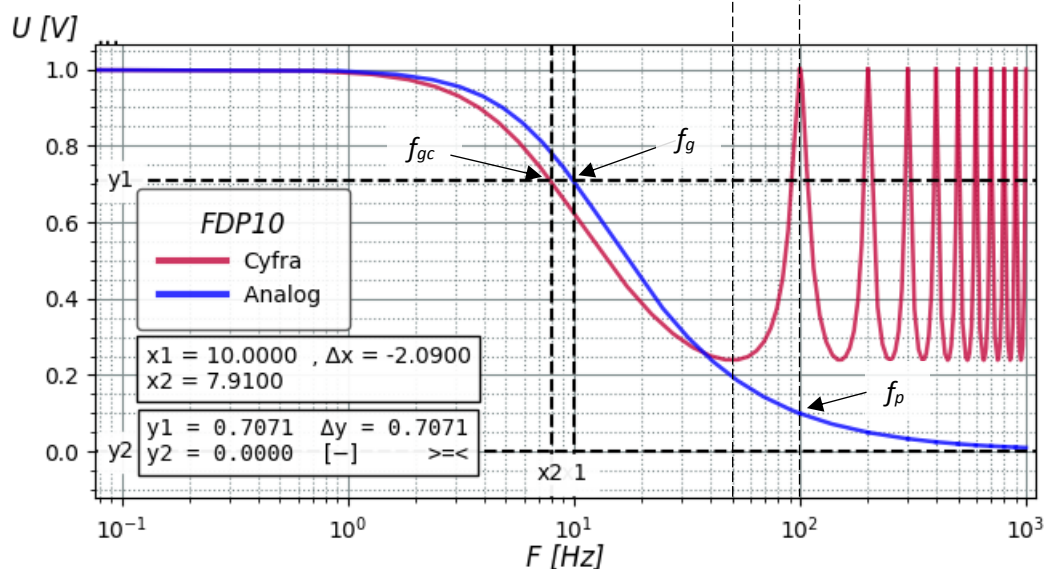
$$z^{-1} \rightarrow e^{-j\omega \Delta t} \quad (18)$$

otrzymujemy charakterystykę amplitudowo-częstotliwościową ekwiwalentnego filtru cyfrowego opisanego równaniem (15):

$$H_D(j \cdot \omega) = \frac{\alpha}{1 - (1 - \alpha) \cdot e^{-j\omega \Delta t}} \quad (19)$$



Rys.2. Pasma filtru dolnoprzepustowego: $f_g=1$ Hz, $f_p=100$ Hz



Rys.3. Pasma filtru dolnoprzepustowego: $f_g=10$ Hz, $f_p=100$ Hz

Na wykresach powyżej przedstawiona jest charakterystyka amplitudowo-częstotliwościowa analogowego filtru dolnoprzepustowego o częstotliwości granicznej 1 Hz (rys.2) i 10 Hz (rys.3), i odpowiadającego mu filtru cyfrowego. W obu przypadkach częstotliwość próbkowania sygnału wynosi $f_p=100$ Hz.

W pierwszym przypadku dla $f_g = 1$ Hz, $\alpha = \Delta t / (\Delta t + T) \approx 0.06$ charakterystyka a/f filtru cyfrowego pokrywa się z odpowiednią charakterystyką filtru analogowego aż do częstotliwości Nyquista ($f_N = f_p/2$).

W drugim przypadku dla $f_g = 10$ Hz, $\alpha \approx 0.39$ charakterystyka a/f filtru cyfrowego różni się dość znacznie od charakterystyki filtru analogowego. Częstotliwość graniczna (przekroczenie poziomu y_1 na rys.3) w tym przypadku wynosi $f_{gc} \approx 7.9$ Hz. Wraz ze wzrostem częstotliwości granicznej filtru analogowego na którym wzorowany jest filtr cyfrowy te różnice powiększają się, z tym że częstotliwość graniczna filtru cyfrowego jest zawsze mniejsza.

Przedstawiony wyżej przykład wyraźnie pokazuje, że w przypadku realizacji filtracji cyfrowej odwzorowującej analogową, istotnym czynnikiem jest stosunek częstotliwości próbkowania sygnału do częstotliwości granicznej filtru. Inaczej mówiąc, im wyższy ten stosunek, tym bardziej filtr cyfrowy emuluje filtr analogowy. W miarę jego malenia pasmo filtru cyfrowego zawęża się. W dodatku to kryterium częstotliwościowe jest znacznie ostrzejsze od kryterium Nyquista – w pierwszym przypadku wynosi aż 100, a w drugim 10.

Oczywiście można zaprojektować filtr cyfrowy emulujący filtr analogowy dla niskiej częstotliwości próbkowania, ale należy przyjmować inne wartości stałej α niż wynikałaby ona z parametrów filtru analogowego. Dla omawianego przypadku $f_g = 10$ Hz należy przyjąć $\alpha \approx 0.46$

Innym rozwiązaniem jest stosowanie filtrów wyższego rzędu lub, jak wspomniano wcześniej, inny algorytm aproksymacji całki. W praktyce stosuje się kaskady filtrów drugiego rzędu – mniejsze są wtedy wymagania na dokładność reprezentacji liczb niż w przypadku jednego filtru rzędu wyższego niż drugi. Powszechnie stosowaną algorytmem aproksymacji całki jest aproksymacja metodą trapezów (inne nazwy to przekształcenie biliniowe/Tustina), ale nie jest to sztywna reguła.

Przykładowo zastosowanie przekształcenia Tustina do filtru (1) wymaga podstawienia:

$$S \rightarrow \frac{2}{\Delta t} \cdot \frac{1-z^{-1}}{1+z^{-1}} \quad (20)$$

W wyniku czego otrzymamy transmitancję filtru cyfrowego o postaci:

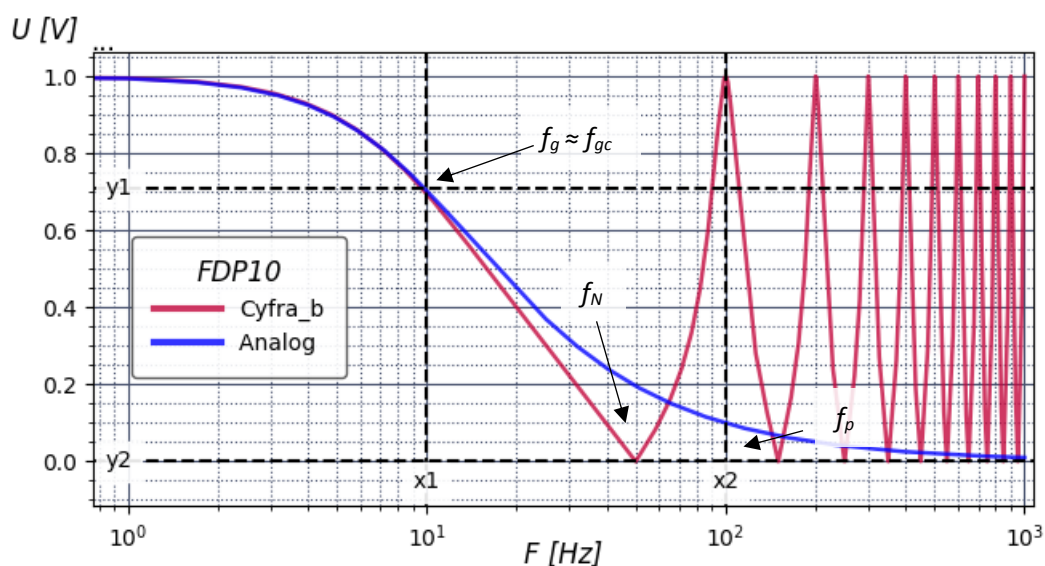
$$H_T[z] = \frac{Y[z]}{X[z]} = \frac{B_0 + B_1 \cdot z^{-1}}{1 + A_1 \cdot z^{-1}} \quad (21)$$

Taka realizacja filtru, w najprostszym przypadku daje następujące równanie różnicowe:

$$y[n] = B_0 \cdot x[n] + B_1 \cdot x[n-1] - A_1 \cdot y[n-1] \quad (22)$$

$$\left\{ Z^*[z^{-k}] \rightarrow x[n-k] \right\}$$

W porównaniu do (10) wymaga dodatkowych operacji arytmetycznych i zapamiętania poprzedniej wartości sygnału wejściowego $x[n-1]$.



Rys.4. Pasmo filtru dolnoprzepustowego: $f_g=10$ Hz, $f_p=100$ Hz
Aproksymacja całki metodą trapezów

W tym przypadku częstotliwość graniczna filtru cyfrowego pokrywa się z częstotliwością graniczną filtru analogowego, a w zakresie większych częstotliwości ma nawet lepsze tłumienia.

Z przedstawionych charakterystyk a/f filtru cyfrowego wynika, że w przeciwieństwie do filtru analogowego jego pasmo powyżej częstotliwości Nyquista nie zanika do zera tylko jest nieskończenie okresowe. Jego odpowiedź impulsowa jest również nieskończona. Stąd nazwa takiego filtru – filtr o nieskończonej odpowiedzi impulsowej (ang. Infinite Impulse Response - IIR).

Określenie „dolnoprzepustowy” odnosi się w tym przypadku do parametrów filtru analogowego, który był punktem odniesienia dla filtru cyfrowego i takie dobranie częstotliwości próbkowania aby pasmo sygnału wejściowego było mniejsze niż częstotliwość Nyquista. Jeżeli dla danej częstotliwości próbkowania nie ograniczymy w ten sposób pasma sygnału jeszcze przed przetwornikiem ADC to filtr cyfrowy też go nie ograniczy. Przepuści zgodnie ze swoim pasmem. Ograniczenie pasma może wynikać ze źródła sygnału np. pomiar wielkości fizycznych o paśmie ograniczonym ze względu na fizyczną naturę obiektu lub musi zostać dokonane za pomocą rzeczywistego filtru analogowego.

Niniejszy dokument zawiera prostą analizę przydatną przy projektowaniu cyfrowych filtrów dolnoprzepustowych przydatnych w mikroprocesorowych układach akwizycji danych.

Obliczenia wykonano przy wykorzystaniu modułu *signal* z biblioteki *SciPy* języka Python, a wykresy wygenerowano za pomocą biblioteki *Matplotlib*.

Przykładowy program w języku Python do wyznaczania charakterystyk amplitudowo-fazowo-częstotliwościowych zawarty jest w osobnym dokumencie.

DODATEK

Wykorzystanie współczynników filtru w realizacji programu dla mikroprocesora.

W nomenklaturze technicznej przyjęło się przedstawiać transmitancję filtrów analogowych za pomocą transformaty Laplace'a. Ogólne równanie transmitancji ma postać następującą:

$$H[s] = \frac{b_M \cdot s^0 + b_{M-1} \cdot s^1 + b_{M-2} \cdot s^2 + \dots + b_0 \cdot s^M}{1 + a_{N-1} \cdot s^1 + a_{N-2} \cdot s^2 + \dots + a_0 \cdot s^N} \quad (23)$$

Rząd filtru definiowany jest tu przez wartość największego wykładnika stojącego przy zmiennej s w mianowniku czyli N . Jest to postać unormowana gdzie $a_0 = 1$ i $M \leq N$. Praktycznie opis każdego filtru można sprowadzić do tej postaci.

Dla filtru analogowego opisanego równaniami (1) współczynniki a i b mają następujące wartości:

$$b_0 = 1, \quad a_0 = T, \quad a_1 = 1$$

Przedstawiona wcześniej metoda analitycznego wyznaczania równania filtru cyfrowego, w przypadku bardziej skomplikowanych filtrów jest czasochłonna. Programy komputerowe takie jak *Matlab* czy *SciPy* (Python) wykorzystują współczynniki a i b do wyznaczenia współczynników A i B transformaty z filtru cyfrowego.

Ogólna postać transmitancji filtru cyfrowego przedstawiona za pomocą transformaty z ma postać:

$$\frac{Y[z]}{X[z]} = \frac{B_0 + B_1 \cdot z^{-1} + B_2 \cdot z^{-2} + \dots + B_M \cdot z^{-M}}{1 + A_1 \cdot z^{-1} + A_2 \cdot z^{-2} + \dots + A_N \cdot z^{-N}} \quad (24)$$

Dla filtrów cyfrowych opisanych równaniami (10, 15) oraz (21, 22), dla $f_g = 10$ Hz, $f_p = 100$ Hz, współczynniki są następujące:

metoda Eulera:

$$B_0 = \alpha f_a = 0.385869545 \quad B_1 = 0$$

$$A_0 = 1 \quad A_1 = -(1 - \alpha f_a) = -0.614130455$$

przekształcenie biliniowe:

$$B_0 = 0.23905722, \quad B_1 = 0.23905722$$

$$A_0 = 1 \quad A_1 = -0.52188555$$

W ogólnym przypadku równanie różnicowe do realizacji w programie mikroprocesora wyznaczone na podstawie obliczonych współczynników A i B , dla filtru rzędu II, będzie miało postać następującą (odwrotna transformata z równania (24)):

$$y[n] = B0 \cdot x[n] + B1 \cdot x[n-1] + B2 \cdot x[n-2] +$$

$$-A1 \cdot y[n-1] - A2 \cdot y[n-2]$$

$$\{ Z^*[z^{-k}] \rightarrow x[n-k] \}$$
(25)

Zawsze $A0 = 1$ i nie występuje już w obliczeniach. Taka postać równania wyjściowego nosi nazwę *formy bezpośredniej 1* (ang. *Direct Form 1 – DFI*).

Realizacja programowa równania (25) wymaga przechowywania dwóch poprzednich próbek sygnału wejściowego $x[n-1]$ i $x[n-2]$ oraz poprzednich wyników wyliczonej wartości sygnału wyjściowego filtru $y[n-1]$ i $y[n-2]$. Nie jest to forma optymalna ze względu na szybkość obliczeń.

Równanie (24) można przekształcić w sposób następujący:

$$\frac{Y[z]}{X[z]} = \frac{N[z]}{D[z]}$$
(26)

$$Y[z] = \frac{N[z]}{D[z]} \cdot X[z] = \frac{X[z]}{D[z]} \cdot N[z] = W[z] \cdot N[z]$$
(27)

Wtedy równanie różnicowe będzie miało postać następującą:

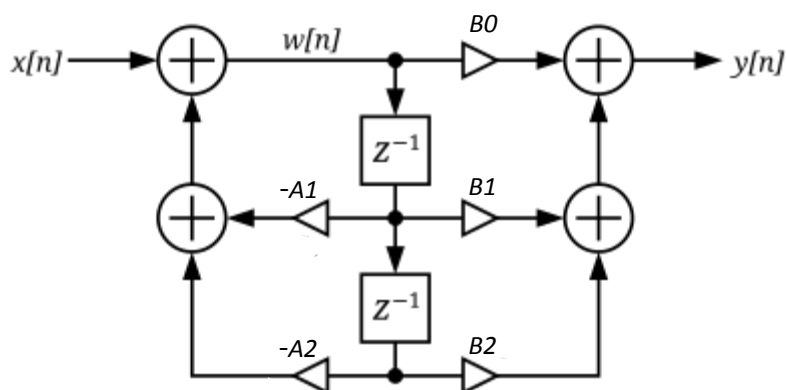
$$w[n] = x[n] - A1 \cdot w[n-1] - A2 \cdot w[n-2]$$

$$y[n] = B0 \cdot w[n] + B1 \cdot w[n-1] + B2 \cdot w[n-2]$$
(28)

Taka realizacja nosi nazwę *formy bezpośredniej 2* (ang. *Direct Form 2 – DF2*). Nie wymaga już przechowywania poprzednich próbek sygnału wejściowego i wyjściowego, a tylko dwóch poprzednich zmiennej pomocniczej w . W analogii do teorii sterowania można ją traktować jako zmienną stanu filtra.

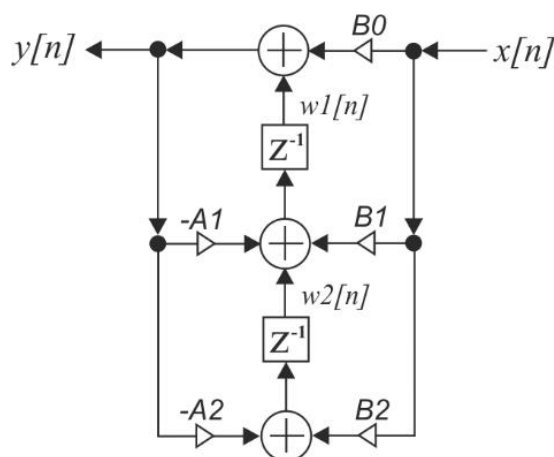
Kolejną postać równań różnicowych można uzyskać zarówno z formuły *DF1* jak i *DF2* zamieniając wejście filtra z jego wyjściem i odpowiednio przekształcając przepływ sygnału. Taka zamiana nie zmienia transmitancji filtra. Najlepiej pokazać to na przykładzie transformacji grafu przepływu sygnałów w filtrze.

Na przykładzie formy *DF2*:



Rys.5. Graf przepływu sygnałów w implementacji *DF2* cyfrowego filtra IIR.
Komponent z^{-1} na schemacie oznacza opóźnienie o 1 okres.

Zamieniając wejście z wyjściem należy przekształcić powyższy schemat zgodnie z zasadą: odwraca się kierunki przepływu wszystkich ścieżek sygnałowych, rozgałęzienia zamienia się w sumy, a sumy w rozgałęzienia. W ten sposób otrzymuje się formę bezpośrednią 2 transponowaną (ang. *Direct Form 2 Transposed – DF2T*).



Rys.6. Graf przepływu sygnałów w implementacji *DF2T* cyfrowego filtra IIR.

Na podstawie powyższego schematu funkcjonalnego układ równań można rozpisać następująco:

$$\begin{aligned} y[n] &= w1[n] + B0 \cdot x[n] \\ w1[n+1] &= w2[n] + B1 \cdot x[n] - A1 \cdot y[n] \\ w2[n+1] &= B2 \cdot x[n] - A2 \cdot y[n] \end{aligned} \quad (29)$$

Obliczenia powinny być wykonywane według kolejności wskazanej powyżej. W tej realizacji także należy przechowywać dwie zmienne $w1$ i $w2$. Indeks $[n+1]$ oznacza, że wyliczone wartości w kroku $[n]$ będą użyte dopiero w kroku następnym $[n+1]$.

Teoretycznie wszystkie powyższe realizacje są równoważne. Jednak w praktyce wybór zależy od konkretnej implementacji, rodzaju filtru, jego charakterystyk częstotliwościowych, wyboru reprezentacji i precyzji liczb, a także docelowego elementu w którym ten filtr będzie realizowany: mikroprocesor wyliczający filtr w sposób czysto programowy czy FPGA posiadająca specjalizowane komponenty dla implementacji takiego filtru. Forma *DFI* z reguły jest używana w przypadku prostych filtrów jak w podanym na początku przykładzie. W bardziej skomplikowanych filtrach używa się pozostałych form. Najprostszym sposobem weryfikacji prawidłowego działania filtru cyfrowego jest symulacja numeryczna uwzględniająca wszystkie aspekty pracy filtru w warunkach docelowych. Dokładna analiza właściwego wyboru formy realizacji filtru cyfrowego wykracza już poza zakres niniejszego opracowania.