Laboratorium 3

Cel ćwiczenia:

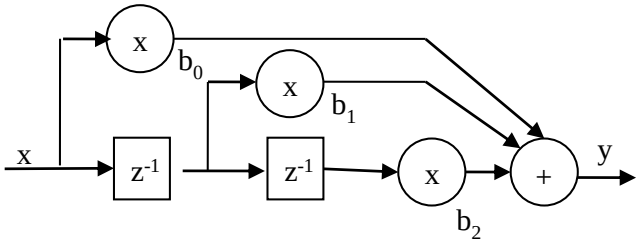
- opanowanie projektowania filtrów za pomocą prostych funkcji oraz aplikacji okienkowej;
- zrozumienie zagadnienia dekompozycji filtrów NOI na sekcje bikwadratowe (ang. SOS);
- zrozumienie metody podziału na bloki przy filtracji sygnałów bardzo długich (np. w czasie rzeczywistym

1. Za pomocą funkcji `fir1` metoda prób i błędów zaprojektuj dolnoprzepustowy filtr SOI o minimalnym rzędzie spełniający zadane wymagania:
   - pasmo przepustowe od `0` do `0.025*Fs/2`
   - pasmo zaporowe od `0.050*Fs/2` do końca
   - zafalowania w paśmie przepustowym: `1` dB
   - tłumienie w paśmie zaporowym: `60` dB

   Wyświetl charakterystykę amplitudową i fazową (`freqz`) oraz opóźnienie grupowe (`grpdelay`). Sprawdź czy wymagania zostały spełnione. Wyświetl zera i bieguny na okręgu jednostkowym (`zplane`).
   Uwaga 1: We wszystkich funkcjach dotyczących projektowania filtrów w Matlabie parametry częstotliwościowe są unormowane do częstotliwości Nyquista czyli `w=1.0` odpowiada `Fs/2`.
   Uwaga 2: Wymagania są wysokie – proszę się nie zdziwić, że rząd filtru wychodzi trzycyfrowy!
2. Za pomocą narzędzia `fdatool`(albo `filterDesigner`)zaprojektuj filtr jak w 1 wyświetl i porównaj charakterystyki z zadań 1 oraz 2 a następnie oceń o ile lepiej/gorzej wyszedł filtr.
   EXTRA: Interesujące zagadnienie badawcze: co w tych wymaganiach na filtr najsilniej wpływa na rząd filtru?
3. Za pomocą pary funkcji `buttord`/`butter` zaprojektuj filtr o NOI o minimalnym rzędzie spełniający wymagania punktu 1. Wyświetl charakterystykę amplitudową i fazową oraz opóźnienie grupowe. Sprawdź czy wymagania zostały spełnione. Wyświetl zera i bieguny na okręgu jednostkowym. Powtórz zadanie dla filtrów Czebyszewa i eliptycznego – tu już użyj narzędzia `fdatool/filterDesigner`.
   Uwaga: `fdatool/filterDesigner` można użyć do wyświetlenia charakterystyk wszystkich filtrów na jednym wykresie (ikonka), wynik projektowania z zadaniem 1 należy wtedy zaimportować do filterDesigner'a i być może zapisać w aplikacji "Filter Manager".
4. Zaprojektuj (`fdatool`()) filtr NOI rzędu 10 lub wyższego (samodzielnie dobierz jakieś wymagania). Porównaj wartość największego i najmniejszego co do modułu współczynnika. Przetestuj jego realizację w strukturze prostej - `filter()` przy pomocy: impulsu jednostkowego, skoku jednostkowego, sinusoidy w paśmie przepustowym i zaporowym.
   Zdekomponuj filtr na sekcje bikwadratowe i przetestuj ponownie – przyda się funkcja `sosfilt()`, po drodze warto zrozumieć co to jest ta macierz Lx6 :-).
   Uwaga 1: `filterDesigner` może zaprojektować współczynniki w strukturze prostej lub kaskadowej – przełącza się to w menu „Edit".
   Uwaga 2: współczynniki można wyeksportować do zmiennych Matlaba (menu „File=>Export")
5. Zarejestruj lub wygeneruj bardzo długi sygnał (np. 60 sekund przy `Fs=48` kHz). Podziel na krótkie bloki i przefiltruj bloki po kolei filtrem SOI (dowolnie zaprojektowanym). Zadbaj o zapamiętanie stanu filtru pomiędzy blokami, aby na początku kolejnego bloku nie powstał stan przejściowy.
6. Zrealizuj to samo za pomocą `fftfilt()` (uwaga, tu samodzielnie musisz zadbać o sklejanie na granicach bloków!)
   W obu przypadkach odsłuchanie sygnału może wykazać, że sklejanie się nie udało :-).

ZADANIA EXTRA:

7. Stwórz własną funkcję `mysoi(B, x)` realizującą filtrację SOI dowolnego rzędu (w uproszczeniu 2 rzędu). Porównaj działanie funkcji z wbudowaną funkcją `filter` dla wektora `B = [1 2 1]` (tzn. sprawdź, czy działa tak samo i jak duża jest różnica wynikająca np. z innego zaokrąglania wyników).



8. Stwórz własną funkcję `mynoi(A,x)` realizującą filtrację NOI dowolnego rzędu (w uproszczeniu 2 rzędu) bez zer w filtrze. Współczynniki filtru podawane są w wektorze A. Porównaj działanie funkcji z wbudowaną funkcją `filter` dla wektora `A = [1.00 -1.78 0.80]`.