Jacek Misiurewicz
Krzysztof Kulpa
Piotr Samczyński
Mateusz Malanowski
Piotr Krysik
Łukasz Maślikowski
Damian Gromek
Artur Gromek
Marcin K. Bączyk

Zakład Teorii Obwodów i Sygnałów
Instytut Systemów Elektronicznych
Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Laboratorium Cyfrowego Przetwarzania Sygnałów

Wersja do wydruku - bez części teoretycznej

Warszawa 2020

Ćwiczenie 4

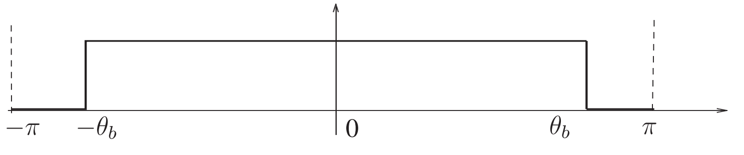
Filtry – właściwości, projektowanie, przetwarzanie sygnałów

Część teoretyczną w tej wersji opuszczono.

4.2. Zadania do pracy własnej studenta

Podobne zadania mogą znaleźć się na wejściówce. Nie dotyczy to zadań oznaczonych tu jako „trudne”.

1) Idealny filtr LP.



Oblicz odpowiedź impulsową idealnego filtru dolnoprzepustowego o charakterystyce amplitudowej jak na rysunku. Przyjmij $\theta_b = \pi/4$ i charakterystykę fazową równą zeru. **Wskazówka:** Przecałkowanie funkcji wykładniczej w zadanych granicach nie jest trudne.

2) Obcinając wynik poprzedniego zadania oblicz współczynniki przyczynowego filtru SOI rzędu 8 (czyli mającego odpowiedź impulsową o długości 9 próbek). Naszkicuj odpowiedź impulsową, zastanów się jakie będzie on miał opóźnienie grupowe. **Wskazówka:** Zajrzyj do opisu metody projektowania filtrów „poprzez obcięcie szeregu Fouriera”.

3) Myśląc kategoriami „mnożenie w czasie = splot w częstotliwości” spróbuj naszkicować charakterystykę uzyskanego filtru przyczynowego i oszacować szerokość pasma przejściowego.

4) Jak zmieniłaby się ww. charakterystyka, gdyby zamiast obcięcia użyć okna (nieprostokątnego)?

5) Oblicz i naszkicuj charakterystykę amplitudową filtru SOI o współczynnikach:

a) $[1,\ -1]$,

b) $[1,\ 1,\ 1]$,

c) $[1,\ -2,\ 1]$.

6) Przeanalizuj filtr NOI o jednym biegunie rzeczywistym w punkcie a. Wybierz $a\colon |a| < 1$, bo nie chcemy filtru niestabilnego.

a) Zapisz jego transmitancję w dziedzinie $\mathcal{Z}$.

b) Znajdź charakterystykę filtru, spróbuj ją naszkicować (do tego celu wylicz wartości w charakterystycznych punktach – max, min itd. …).

c) Znajdź odpowiedź impulsową.

7) Przeanalizuj filtr NOI o dwóch biegunach sprzężonych w punktach $r\mathrm{e}^{\pm \mathrm{j}\theta_p}$. Sam wybierz parametry r i θ_p, ale dopilnuj aby filtr był stabilny.

4.3. Dostępny sprzęt i oprogramowanie

4.3.1. Standardowe funkcje Matlaba

Filtracja cyfrowa w Matlabie została zaimplementowana w standardowej funkcji `filter`.

Parametrami polecenia `filter(B,A,x)` są:

`B` – wektor współczynników licznika (części SOI),

`A` – wektor współczynników mianownika (części NOI); jeśli realizujemy filtr SOI, wstawiamy tu 1,

`x` – sygnał do przefiltrowania (wynik będzie miał taki sam rozmiar jak `x` – zazwyczaj podajemy tu wektor sygnału, ale `x` może też być macierzą, i wtedy każda jej kolumna zostanie przefiltrowana oddzielnie).

Charakterystykę częstotliwościową filtru najsprawniej oblicza się za pomocą funkcji `[h w]=freqz(B,A,N)`, gdzie `B` i `A` to współczynniki licznika i mianownika, natomiast `N` – liczba punktów na osi częstotliwości od 0 do π (domyślnie 512).

Jeśli pominiemy parametry wyjściowe, otrzymamy wykres; w przeciwnym przypadku wynikiem jest wektor zespolonych wzmocnień `h` i wektor pulsacji unormowanych `w`, z których można samemu stworzyć wykres charakterystyki amplitudowej i fazowej.

Najprościej (chociaż to jest wersja „dla opornych”) filtry projektuje się za pomocą Matlabowego narzędzia `fdatool`. Jest to narzędzie z interfejsem graficznym, w którym można z menu zadać klasę filtru i częściowo graficznie podać wymagania częstotliwościowe. Potem można zobaczyć wyniki projektowania i wyeksportować współczynniki. Jak z każdym narzędziem uniwersalnym, czasem nie rozumiemy co robimy, a czasem zaplączemy się przy wydobywaniu z niego współczynników. W niniejszym ćwiczeniu można się tym narzędziem posłużyć przy niektórych zadaniach *extra*.

4.4. Eksperymenty do wykonania w laboratorium – projektowanie i wykorzystanie filtrów cyfrowych

4.4.1. Filtr SOI i NOI, związek zer i biegunów z charakterystyką

4.4.1.1. Usuwanie składowej stałej

- Wygeneruj sygnał sinusoidalny ze składową stałą. Użyj częstotliwości unormowanej równej $0{,}1 + \textit{numer stanowiska}/50$

```
x=2+sin(f*2*pi*[0:99]);
plot(x)
mean(x)%obliczenie wart. sredniej
```

Zanotuj wyrażenie użyte do generacji sygnału oraz obliczoną wartość średnią (składową stałą). Zanotuj

- Oblicz współczynniki wielomianu o podwójnym zerze w punkcie $\mathrm{e}^{\mathrm{j}\pi\cdot 0}$ (e do potęgi j pi razy zero), umieść je w zmiennej `B`.
  Wskazówka: Obliczenie współczynników wielomianu – `help poly`, ale w tym zadaniu szybciej jest pomyśleć samemu.
  Zanotuj obliczone współczynniki Zanotuj
- Przefiltruj sygnał filtrem, którego transmitancja jest znalezionym wielomianem, obejrzyj wynik

```
y=filter(B,1,x);
plot(0:99,x,0:99,y);
mean(y)
```

Zanotuj wartość składowej stałej w sygnale po filtracji Zanotuj

***Odpowiedz na pytanie*:** Jakiego typu (SOI/NOI) filtr użyto w tym eksperymencie? Zapisz wyrażeniem matematycznym jego transmitancję $H(z)$ i charakterystykę amplitudową $A(\theta)$. Odpowiedz ?

4.4.1.2. Prosty rezonator

- Przygotuj (oblicz współczynniki) filtru 2 rzędu o biegunach w punktach $0{,}9\mathrm{e}^{\pm \mathrm{j}0{,}2\pi}$ i z podwójnym zerem w $\mathrm{e}^{\mathrm{j}\pi} = -1$. Dla zgodności z `help`em do funkcji `filter` umieść współczynniki licznika w zmiennej `B`, a mianownika – w `A`.
  Wskazówka: Konieczne jest znalezienie współczynników wielomianu, a więc znów przyda się `poly`. W Matlabie nie ma stałej e, ale jest funkcja `exp()`.
  Zanotuj współczynniki na wszelki wypadek. Zanotuj
- Zbadaj przygotowany filtr według poniższego opisu.
  – Utwórz wykres odpowiedzi impulsowej filtru; poniższy kod pokazuje przykład, jak szybko skonstruować impuls jednostkowy.

```
dlt=zeros(1,64); dlt(1)=1;,
plot(filter(...)) %zastanów się co tu dalej wpisać
```

  – Oblicz charakterystykę częstotliwościową.

```
[H, w]=freqz(...);
```

– Oblicz opóźnienie grupowe (przybliż $\mathrm{d}\phi/\mathrm{d}\theta$ skończonymi różnicami[1]).

```
grd=-diff(angle(H))./diff(w); % uwaga na minus!
```

– Wykreśl na wspólnym wykresie charakterystykę (amplitudową) i opóźnienie grupowe.

```
plot(w,abs(H),w(2:end),grd);
```

Zmień oś poziomą wykresu z pulsacji (częstości) na częstotliwość unormowaną[2] i ***zapisz w sprawozdaniu zmienioną linijkę kodu***.

Zanotuj

Uwaga: Jeżeli faza `angle(h)` wychodzi nieciągła z powodu przekroczenia π, możesz użyć `unwrap(angle(h))`; nieciągłością dla punktu $\theta = \pi$ można się nie przejmować – wzmocnienie jest tu zerowe[3].

Naszkicuj

Naszkicuj wykres opóźnienia grupowego i wzmocnienia filtru (charakterystyki częstotliwościowej).

Zanotuj

Zanotuj maksymalne opóźnienie i jego lokalizację na osi częstotliwości.

Przetestuj filtr w kilku punktach charakterystyki. W każdym punkcie wygeneruj sinusoidę o odpowiedniej częstotliwości (najlepiej o jednostkowej amplitudzie), przefiltruj, i sprawdź czy zmierzone (z wykresu sygnału wejściowego i wyjściowego) wzmocnienie jest zgodne z wykresem uzyskanym z `freqz()`. Wyniki ***zanotuj w tabelce, wpisując konkretne wartości częstotliwości***.

Zanotuj

Częstotliwość f_n	(zero)	(nieduża)	(w max. ch-ki)	(duża)
Amplituda we A_{in}				
Amplituda wy A_{out}				
Wzmocnienie z charakterystyki $\lvert H(f_n)\rvert$				
Wzmocnienie z testu A_{out}/A_{in}				

Odpowiedz

Odpowiedz na pytanie: Jak można w praktyce wyznaczyć przybliżoną charakterystykę amplitudową nieznanego filtru? Jakie mogą być źródła niedokładności?

4.4.1.3. $\boxed{\textbf{Zadanie extra}}$ Prosty rezonator II

Powtórz poprzednie zadanie dla biegunów w $0{,}9\mathrm{e}^{\pm \mathrm{j}0{,}8\pi}$.

---

[1] W Matlabie jest funkcja `grpdelay()`, która oblicza opóźnienie w bardziej wyrafinowany sposób.

[2] Student powinien już pamiętać, że pulsacja unormowana $\theta = \pi$ odpowiada częstotliwości unormowanej $f_n = 0{,}5$ albo częstotliwości fizycznej $f = 0{,}5 f_s$. Ten przypis powstał dlatego, że niektórzy nie pamiętają i zadają prowadzącym głupie pytania.

[3] Jeśli na wykresie widzisz nieciągłość gdzieś indziej, być może zrobiłeś jakiś błąd – nieciągłość *może* się pojawić w charakterystyce fazowej filtru, ale akurat *przy tym konkretnym filtrze* jej nie ma.

4.4.2. Bezpośrednie projektowanie filtrów SOI – metoda obcięcia szeregu Fouriera

4.4.2.1. Filtr o odpowiedzi $\mathrm{sinc}()$

Spróbujemy teraz zaprojektować idealny filtr dolnoprzepustowy (o prostokątnej charakterystyce częstotliwościowej i o zerowej fazie). Granicą pasma przepustowego filtru będzie $\theta_G = (2 + \textit{nr stanowiska})/20 \cdot \pi$.

Zadanie extra Wygeneruj przebieg będący odpowiedzią impulsową idealnego filtru LP (dolnoprzepustowego) o żądanej pulsacji granicznej.

```
n=-1023:1024;% oś czasu
thetaG=(2+nrs)/20*pi
x=n*thetaG;
h=sin(x)./x;
h(n==0)
```

W ostatnim poleceniu wykorzystujemy tzw. indeksowanie logiczne. Operator `n==0` generuje wektor zer i jedynek (na indeksach, na których elementy wektora `n` /nie spełniają/spełniają/ wyrażenia logicznego); następnie tym wektorem indeksujemy wektor `h` – to oznacza wybór z wektora `h` elementów, dla których wektor indeksów ma wartość prawdy (czyli 1).

***Odpowiedz na pytanie*:** Jaką wartość powinna mieć funkcja $h(n)$ dla $n = 0$? Przypomnij sobie pojęcia granicy i ciagłości funkcji. Zastąp w obliczonym wektorze podejrzany element[4] wartością, odpowiadającą zerowej próbce uciąglonej funkcji $\sin(x)/x$.

Odpowiedz

```
h(n==0)=.....% tu samodzielna decyzja studenta ;-)
h=h*thetaG/pi;
%normalizacja, aby wzmocnienie w pasmie przepustowym było równe 1
plot(n,h,'-*');
```

Zanotuj (analityczne) obliczenia granicy.

Zanotuj

Jeśli nie robiłeś zadania *extra*, oblicz odpowiedź impulsową, korzystając z funkcji `sinc` (zdefiniowanej jako $\frac{\sin \pi x}{\pi x}$).

```
n=-1023:1024;% oś czasu
thetaG=(2+nrs)/20*pi
x=n*thetaG;
h=thetaG/pi*sinc(x/pi);
plot(n,h,'-*');
```

Powiększ wykres i naszkicuj go dla $-10 < n < 10$.

Naszkicuj

Wyświetl transformatę Fouriera odpowiedzi impulsowej (obliczoną przez FFT z dopełnieniem zerami do 2^{16}).

```
A=(abs(fft(h,2^16)));
```

---

[4] Jeśli czytasz ten tekst nie wykonując ćwiczenia, domyśl się, co mógł obliczyć Matlab jako $\sin(0)/0$.

```
plot(A)
```

Odpowiedz

***Odpowiedz na pytania*:**

- Jaka jest (według Twojej wiedzy teoretycznej) zależność między obliczoną transformatą a charakterystyką częstotliwościową filtru?
- Czy obliczona odpowiedź impulsowa była rzeczywiście idealna? (Przyjrzyj się rysunkowi). **Wskazówka:** Jaka jest dziedzina funkcji sinc (dziedzina funkcji $f(x)$ – zbiór wartości x, dla których $f(x)$ jest określona)?
- Jaka jest amplituda zafalowań pasma przepustowego w okolicy skoku charakterystyki? Wykonaj pomiar kursorem na rysunku. W tabeli wg wzoru poniżej zanotuj wartość jako % skoku (zauważ, że obcięliśmy odpowiedź filtru idealnego do zakresu $-1023 : 1023$, czyli zastosowaliśmy okno prostokątne!).

W tabeli poniżej na razie wypełniamy krok po kroku wiersz dotyczący okna prostokątnego, oknem Hamminga zajmiemy się za chwilę.

Użyte okno	zafalowania w %	I listek w dB	szer. pasma przejściowego filtru	szerokość listka głównego okna
prostokątne				
Hamminga				

Wyświetl ten sam wykres w skali logarytmicznej (w decybelach).

```
plot(20*log10(A))
axis([0 2^16 -50 30])
```

Zanotuj

Zanotuj w tabeli poziom pierwszego listka zafalowań w pasmie zaporowym wyrażony w dB.

Wykreśl transformatę Fouriera funkcji okna prostokątnego.

```
G=abs(fft(ones(size(h)),2^16));
figure
plot(G);
```

Zanotuj

Zanotuj w tabeli szerokość pasma przejściowego filtru i porównaj z szerokością listka głównego transformaty okna.

Wskazówka: Najprościej będzie notować szerokość pasma, licząc ile próbek FFT mieści się w nim – jest to sensowne, jeśli rozmiary FFT są takie same. Bądź jednak gotowy/a na pytanie prowadzącego „a ile to będzie w skali częstotliwości unormowanej?”.

Popraw charakterystykę filtru, używając okna Hamminga.

```
gg=hamming(length(h));
hh=h(:).*gg;
```

Wykreśl na ekranie charakterystykę poprawionego filtru oraz transformatę Fouriera funkcji okna Hamminga.

Zanotuj

Zanotuj w tabeli szerokość pasma przejściowego filtru i porównaj z szerokością listka głównego transformaty okna. Spróbuj sformułować wniosek – co poprawiliśmy i jakim kosztem.

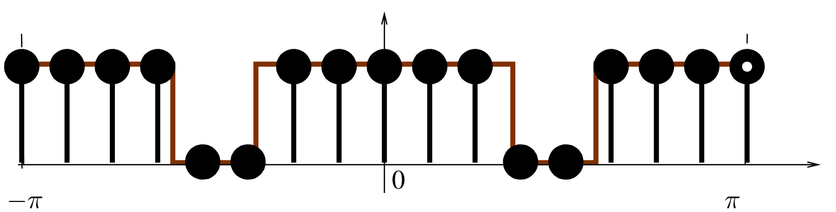
4.4.2.2. Filtr o zadanej charakterystyce częstotliwościowej

Zaprojektuj pasmowozaporowy (ang. *bandstop*) filtr SOI usuwający z sygnału zakłócenie o częstotliwości $F_c = (1 + \textit{numer stanowiska}/10)$ kHz. W dalszej części ćwiczenia będziesz usuwać z rzeczywistego sygnału takie zakłócenie wygenerowane generatorem laboratoryjnym. Do projektu załóż, że sygnał będzie próbkowany z częstotliwością $F_s = 48$ kHz.

```
Fs=48e3;
```

Spróbujemy wykonać ten projekt „ręcznie” – krok po kroku, oglądając wyniki pośrednie. Droga ta jest długa i żmudna, ale bardzo zalecamy przejście jej samodzielnie, gdyż pokazuje ona rozmaite niespodziewane efekty o dużej wartości dydaktycznej. Tylko w razie poważnego braku czasu można udać się „na skróty”, tj. użyć gotowej procedury Matlaba `fir2`, albo (o zgrozo) Matlabowego narzędzia z interfejsem graficznym czyli `fdatool`.

Ćwiczona tu metoda projektowania polega na obliczeniu odpowiedzi impulsowej filtru poprzez odwrotną transformatę Fouriera żądanej charakterystyki. Można to zrobić analitycznie, ale my posłużymy się komputerem (oraz odwrotną DTF). W tym celu żądaną charakterystykę filtru zadamy w postaci spróbkowanej (rys. 4.1).



Rysunek 4.1. Sposób zadania charakterystyki filtru

- Sprawdź, z jaką dokładnością można ustawić częstotliwość na generatorze i przyjmij do projektu pasmo zaporowe 1,5 raza szersze[5]. ***Zanotuj przyjęte założenia projektowe, wyrażone w dziedzinie częstotliwości unormowanej.*** **Zanotuj**
- Załóż, że pasmo przejściowe filtru może mieć szerokość porównywalną z 1/4 pasma zaporowego. Zastanów się, jaki z tego wynika minimalny rząd filtru L. **Wskazówka:** Wykorzystaj wyniki poprzedniego zadania co do związku szerokości pasma przejściowego z listkiem głównym transformaty okna. Zastanów się jaka jest szerokość listka głównego transformaty impulsu prostokątnego o długości $L + 1$.
  Zanotuj obliczony minimalny rząd. **Zanotuj**
  Tu spójrz na zegarek i zdecyduj (ew. w porozumieniu z prowadzącym) czy idziesz “na skróty (texttttfdatool()) czy krok po kroku.
- Przyjmij, że charakterystykę filtru spróbkujesz w $L_h = 4 \cdot (L + 1)$ punktach (gdzie L jest wyznaczonym przed chwilą rzędem filtru, zaokrąglonym do liczby

[5] Nawet jeśli masz bardzo pewną rękę, nie przyjmuj mniej niż ± 50 Hz.

nieparzystej).
Oblicz zakres indeksów punktów, w których charakterystyka ma być wyzerowana.**Wskazówka:** Najlepiej sobie narysować ręcznie taką charakterystykę i na niej zaznaczać.

```
Lz1=....;% indeks początkowy
Lz2=....;%indeks końcowy
```

Utwórz wektor `H` zawierający pożądaną charakterystykę filtru dla częstości od 0 do π, wypełniając ją jedynkami (a zerami w pasmie zaporowym).

```
H=ones(1,Lh/2);
H(Lz1:Lz2)=0;
```

Sprawdź, czy projektujesz właściwy filtr – wykreśl `H` w funkcji częstotliwości fizycznej.

```
plot([0:(Lh/2-1)]*Fs/Lh,H);
```

Teraz „dostaw” drugą połówkę charakterystyki (dla ujemnych częstotliwości, co przy cykliczności widma odpowiada częstotliwościom powyżej $F_s/2$).

```
H=[H H(end:-1:1)];
plot([0:(Lh-1)]*Fs/Lh,H);
```

Oblicz odwrotną transformatę Fouriera charakterystyki (za pomocą IFFT) – czyli odpowiedź impulsową filtru (operator `real()` usunie efekty błędów numerycznych).

```
h=real(ifft(H));
plot(h);
```

Jak widać, odpowiedź impulsowa jest cykliczna (wynika to z właściwości DFT) – przytnij ją do zakresu czasu od $n = -L/2$ do $n = +L/2$ i ustaw w odpowiedniej kolejności

```
h=[h((end-L/2):end) h(1:(L/2+1))];
plot(h);
```

Prawie gotowe – sprawdź co wyszło!!!

```
freqz(h,1);
```

Odpowiedz

?

***Odpowiedz na pytanie*:** Czy wyszedł Ci filtr, jakiego oczekiwałeś(-aś)? Spróbuj określić przyczyny niepowodzenia…

Zmodyfikuj współczynniki filtru z użyciem okna Hamminga.

```
hh=h(:).*hamming(length(h));
```

Zanotuj

Zweryfikuj charakterystykę filtru (np. z użyciem `freqz`). ***Zanotuj szerokości pasma przejściowego i zaporowego*** i porównaj z założeniami.

Jeśli filtr teraz spełnia Twoje oczekiwania, warto dla pewności zapisać współczynniki do pliku.

```
save nazwapliku.mat hh
```

4.4.3. Zadanie extra Projektowanie filtrów SOI metodą optymalizacyjną

Zaprojektuj filtr dolnoprzepustowy SOI metodą optymalizacyjną (`firpm`). Przyjmij „łatwe” parametry filtru – szerokie pasmo przejściowe (około 1/20 częstotliwości próbkowania).

Uwaga: Wszystkie procedury Matlaba przeznaczone do projektowania filtrów przyjmują specyfikację częstotliwości w postaci liczb od 0 do 1, gdzie 1 odpowiada $\theta = \pi$.

Przyjmij pasmo przepustowe do częstotliwości $(2 + \textit{nr stanowiska}) \cdot \pi/24$.

- Użyj rzędu 10, zwiększ jeśli potrzeba. ***Zanotuj wynikowe pasmo przejściowe i rząd filtru***. Zanotuj
- Utrudnij zadanie optymalizacji – przy tych samych pozostałych parametrach wymuś węższe pasmo przejściowe.
- Zwiększ rząd, aby zniwelować niekorzystne efekty zaostrzenia wymagań.

***Odpowiedz na pytanie*:** Jakiej zmiany rzędu filtru (w przybliżeniu) wymaga dwukrotne zwężenie pasma przejściowego? Odpowiedz ?

4.4.4. Projektowanie filtrów NOI z prototypu analogowego

Prototyp analogowy będzie przekształcony w filtr cyfrowy przy wykorzystaniu transformacji biliniowej zmiennej s w zmienną z. W Matlabie zazwyczaj projektowanie jest dwuetapowe – najpierw obliczamy rząd filtru konieczny do spełnienia wymagań (i ew. parametry pomocnicze) funkcją `***ord` – np. `buttord()`, a następnie podajemy je do funkcji znajdującej współczynniki – np. `butter()`. Aby dokładnie zrozumieć, jakie parametry przyjmują te funkcje, zawsze warto użyć polecenia `help`.

Uwaga: Wszystkie procedury Matlaba przeznaczone do projektowania filtrów przyjmują specyfikację częstotliwości w postaci liczb od 0 do 1, gdzie 1 odpowiada $\theta = \pi$.

4.4.4.1. Projektowanie filtru

- Zaprojektuj filtr NOI dolnoprzepustowy (LP) o parametrach:
  – pasmo przepustowe (*passband*) do $0{,}2\pi$,
  – pasmo zaporowe (*stopband*) od $0{,}3\pi$,
  – zafalowania w paśmie przepustowym (*passband ripple*) 0,5 dB,
  – tłumienie w paśmie zaporowym (*stopband attenuation*) 80 dB
  (przy niektórych prototypach nie wszystkie parametry są wykorzystywane).
  Porównaj wyniki projektowania z użyciem różnych prototypów wybierz dwa z poniższych):
  1) Butterwortha (`buttord, butter`),
  2) Czebyszewa typ I i II (`cheb1ord, cheby1, cheb2ord, cheby2`),
  3) Cauera (eliptyczny: `ellipord, ellip`).

- W każdym przypadku wyświetl uzyskaną charakterystykę i sprawdź, czy spełnia wymagania.

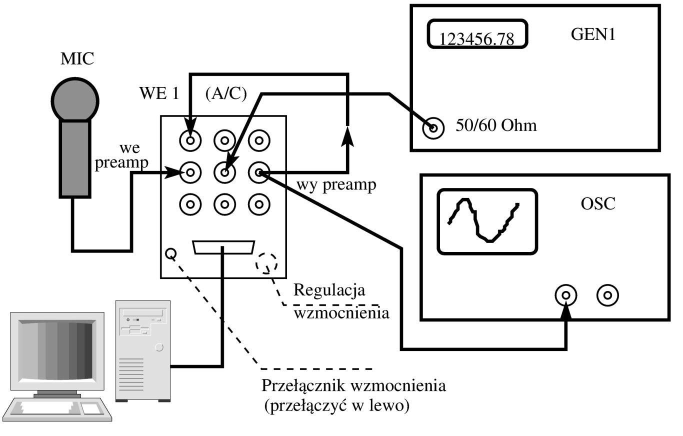
Zanotuj

Zanotuj rząd filtru w każdym przypadku i opisz zachowanie w paśmie zaporowym i przepustowym (zafalowania, sposób zanikania zafalowań z częstotliwością itp.).

4.4.5. Realizacja filtrów i zastosowania filtrów

4.4.5.1. Filtracja sygnału zarejestrowanego

- Zarejestruj sumę sygnału sinusoidalnego o częstotliwości $F_c = (1 + +\textit{numer stanowiska}/10)$ kHz i mowy z mikrofonu (rys. 4.2 – przedwzmacniacz pełni również rolę sumatora sygnałów). Do rejestracji użyj znanego Ci już polecenia `LCPS_getdata(Nprobek, Mblokow, Tprobkowania)`.
- Wykorzystaj filtr zaprojektowany w zadaniu 4.4.2.2 do usunięcia zakłócenia.
- Sprawdź widma sygnału wejściowego i wyjściowego filtru.
- Odsłuchaj wynik.



Rysunek 4.2. Połączenia do zakłócania mowy sygnałem z generatora

4.4.5.2. Zadanie extra Usuwanie przydźwięku 50 Hz

W tym zadaniu trzeba polegać na własnej fantazji.

- Pozyskaj sygnał z przydźwiękiem sieci (źródłem może być dźwięk zarejestrowany ze źle połączonym przewodem masowym, niestarannie pobrany sygnał EMG, mikrofon przystawiony do bzyczącego urządzenia – uwaga, tu może pojawić się raczej 100 Hz).
- Przeanalizuj sygnał (np. użyć FFT, albo spektrogramu), aby zrozumieć, jakie w nim występują zakłócenia,
- Zaprojektuj odpowiedni filtr, sprawdź go dokładnie i zastosuj do posiadanego sygnału.

Ciekawą alternatywą do projektowania filtru pasmowo-zaporowego jest użycie filtru wycinającego (ang. *notch filter*) o transmitancji zawierającej parę zer o promieniu 1 na częstotliwości zakłócenia i parę biegunów *na tej samej częstotliwości* lecz o promieniu nieznacznie mniejszym od 1.

4.4.5.3. Zadanie extra Kwantyzacja współczynników

Aplikacja `lab4receiver.vi` służy do analizy wpływu kwantyzacji na charakterystyki filtrów cyfrowych.

Współczynniki filtru, tak przed jak i po kwantyzacji, pokazane są w tabelach w zakładce **„Filter coefficients”**. Domyślnie wczytany filtr jest filtrem FIR rzędu 100 o częstotliwości odcięcia $0{,}075 \cdot f_s$. Możliwe jest manualne zdefiniowanie współczynników filtru (po kliknięciu przycisku **Reset** i wpisaniu własnych wartości) lub wczytanie współczynników filtru z pliku. Format pliku czytanego przez aplikację to dwukolumnowy plik ASCII, gdzie kolumny oddzielone są tabulatorem, a separatorem dziesiętnym jest kropka. Pierwsza kolumna pliku definiuje kolejne współczynniki licznika, poczynając od potęgi 0, druga mianownika.

Liczba bitów stosowanych po kwantyzacji do reprezentacji współczynników filtru jest wartością z kontrolki **„filter coeffs word length after quantization”**.

W zakładce **„frequency domain”** znajdują się wykresy widma amplitudowego sygnału odebranego przed i po filtracji cyfrowej.

Zakładka **„filter plots”** zawiera wykresy odpowiedzi impulsowej, zer i biegunów na płaszczyźnie S oraz charakterystyki amplitudowej wczytanego filtru przed i po kwantyzacji. Poszczególne wykresy można wyłączać/włączać checkboxami przy legendzie (jedynie, kiedy aplikacja nie jest w stanie odbioru!).

Wykorzystując `lab4receiver.vi` zbadaj wpływ kwantyzacji współczynników na charakterystykę filtru.

- Zaprojektuj w Matlabie filtr wysokiego rzędu i sprawdź jego charakterystykę (`freqz`). Propozycje filtrów poniżej, ale możesz użyć także własnych pomysłów.
  - Filtr SOI z zadania 4.4.2.2.
  - Filtr NOI z zadania 4.4.4.1.
- Zapamiętaj współczynniki filtru (`fwsp` musi być tablicą o 2 kolumnach – wektory `A` i `B` w razie potrzeby musisz uzupełnić zerami do równych długości).
  ```
  fwsp=[B(:),A(:)];
  save('fileName.txt', '-ascii','-double','-tabs','fwsp');.
  ```
- Załaduj współczynniki do `lab4receiver`.
- Sprawdź jak zmienia się charakterystyka filtru gdy zmieniasz liczbę bitów kwantyzacji.
- Sprawdź jak filtr filtruje sygnał z generatora, użyj różnych wartości amplitudy i różnych przebiegów (sin, chirp…).