

# Opis systemów dyskretnych, filtry FIR i IIR

Jan Kowalski

13 listopada 2020

## Spis treści

<b>1</b>	<b>Generacja sygnału</b>	<b>3</b>
1.1	Zadanie . . . . .	3
1.2	Rozwiązanie . . . . .	3
<b>2</b>	<b>Decymacja sygnału</b>	<b>4</b>
2.1	Zadanie . . . . .	4
2.2	Rozwiązanie . . . . .	4
<b>3</b>	<b>Projektowanie filtru</b>	<b>5</b>
3.1	Zadanie . . . . .	5
3.2	Rozwiązanie . . . . .	6
<b>4</b>	<b>Decymacja przefiltrowanego sygnału</b>	<b>6</b>
4.1	Rozwiązanie . . . . .	7
<b>5</b>	<b>Rekonstrukcja sygnału</b>	<b>7</b>
5.1	Rozwiązanie . . . . .	8
<b>6</b>	<b>Filtr sinc</b>	<b>9</b>

## Spis rysunków

1	Sygnał oryginalny. . . . .	4
2	Sygnał po decymacji. . . . .	4
3	Porównanie sygnału oryginalnego i po decymacji. . . . .	5
4	Charakterystyki amplitudowe filtrów. . . . .	6
5	Opóźnienie filtru Butterwortha. . . . .	6
6	Sygnał przefiltrowany przez filtr $H_1$ . . . . .	7
7	Porównanie sygnału oryginalnego i po filtracji przez filtr $H_1$ . . . . .	7
8	Przefiltrowany sygnał po decymacji. . . . .	8
9	Sygnał zwiększeniu częstotliwości próbkowania . . . . .	8
10	Sygnał zwiększeniu częstotliwości próbkowania i filtracji. . . . .	9
11	Porównanie sygnałów przed i po rekonstrukcji . . . . .	9
12	Charakterystyki amplitudowe filtrów. . . . .	10
13	Sygnał przefiltrowany przez filtr $H_2$ . . . . .	10

14	Przefiltrowany sygnał po decymacji . . . . .	11
15	Sygnał zwiększeniu częstotliwości próbkowania. . . . .	11
16	Sygnał zwiększeniu częstotliwości próbkowania i filtracji. . . . .	12
17	Porównanie sygnałów przed i po rekonstrukcji . . . . .	12

## Jak ogarnąć środowisko w jakim pracujemy

Całość rozwiązania ma znajdować się w pliku ‘`filtry.m`’. Edytor należy włączyć za pomocą:

```
$ make edit
```

dla sekcji preferujących edytor vim lub:

```
$ make edit-kate
```

jeśli preferuje się edytor kate (graficzny).

W pliku tym piszemy równocześnie trzy rzeczy:

- kod programu interpretowany przez GNU Octave (wolna implementacja Matlab),
- komentarze do kodu, za pomocą linii zaczynających się od ‘%’ (jak standardowy komentarz w Octave/Matlab),
- wnioski umieszczane w raporcie, za pomocą linii zaczynających się od ‘%’.

Raport kompilujemy za pomocą:

```
$ make
```

Raport można wyświetlić za pomocą:


```
$ make show
```

Cały raport generowany jest automatycznie. Zarówno kod programu jak i wnioski dodawane są automatycznie do raportu. Wykresy robione są automatycznie, ale program **musi** umieścić dane do wykresów w odpowiednich plikach. Nazwy plików znajdują się w opisach zadań, np. ‘`x_1`’. Dane należy zapisywać za pomocą funkcji `writedata()`. Funkcja ma następujący prototyp:

```
writedata(name, x, y)
```

*name* to nazwa pliku do którego mają być zapisane dane, *x* to wektor danych na osi *x*, *y* to wektor danych na osi *y*. Dla wykresów czasowych wektor *x* ma reprezentować czas w sekundach. Dla wykresów gęstości widmowej mocy (GWM) wektor *x* ma reprezentować częstotliwość w hercach, a wektor *y* wartości GWM. Przykład:

```
t = [0; 0.1; 0.2; 0.3];
x = [1; 2; 3; 4];
writedata('x_1', t, x);
```

 Fragmenty oznaczone takim znakiem są na dodatkowe punkty, bez nich można uzyskać 10 punktów.

# Deklaracje wstępne

```
function writedata(name, x, y)
    f = fopen(name, "w");
    y(find(isinf(y))) = -1e6;
    fprintf(f, "%f %f\n", [vec(x) vec(y)]');
    fclose(f);
end
```

```
function [y, f] = gwm(x, f_s)
    x = x(length(x) / 2 + 1:end);
    n = length(x);
    y = fft(x) / n;
    y = [y(1); 2 * y(2:n / 2); y(n / 2 + 1)];
    y = 20 * log10(abs(y));
    f = [0:n/2]' / n * f_s;
end
```

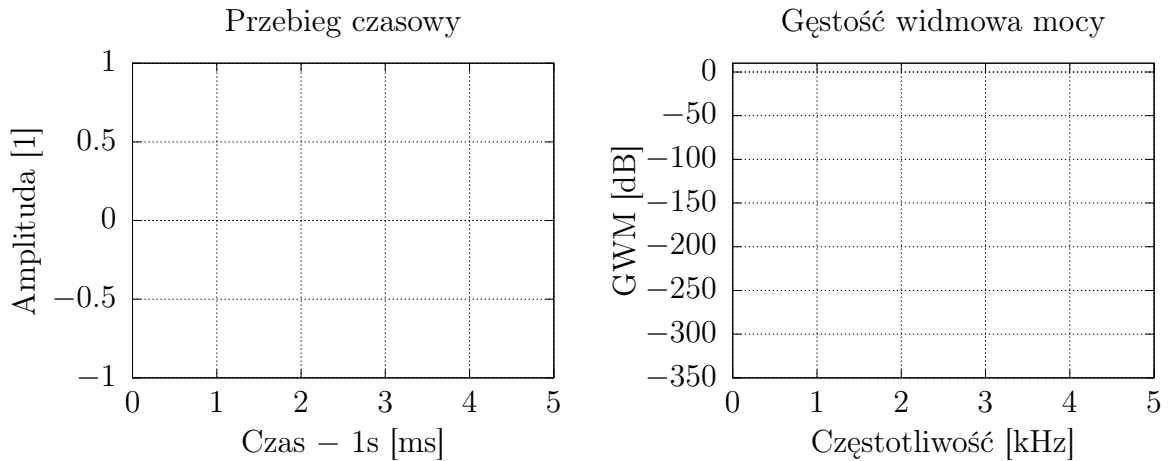
## 1 Generacja sygnału

### 1.1 Zadanie

1. Wygeneruj 16000 próbek sygnału dyskretnego  $x_1$  będącego sumą trzech składowych sinusoidalnych:  $f_1$ ,  $f_2$  oraz  $f_3$ . Amplituda każdej ze składowych to 0.25. Częstotliwość próbkowania wynosi  $f_s = 8000$  Hz.
2. Zapisz wygenerowany sygnał do pliku 'x\_1'.
3. Wyznacz GWM z drugiej połowy sygnału (GWM zawsze będziemy liczyć z tylko drugiej połowy sygnału, by ominąć ewentualne stany nieustalone związane z filtracją na początku, sugerowane jest napisanie sobie odpowiedniej funkcji to wszystko realizującej), oraz zapisz ją do pliku 'X\_1'.
4. Zapisz wnioski.

### 1.2 Rozwiązanie

Przebieg czasowy oraz GWM sygnału zostały umieszczone na rysunku 1.



Rysunek 1: Sygnał oryginalny.

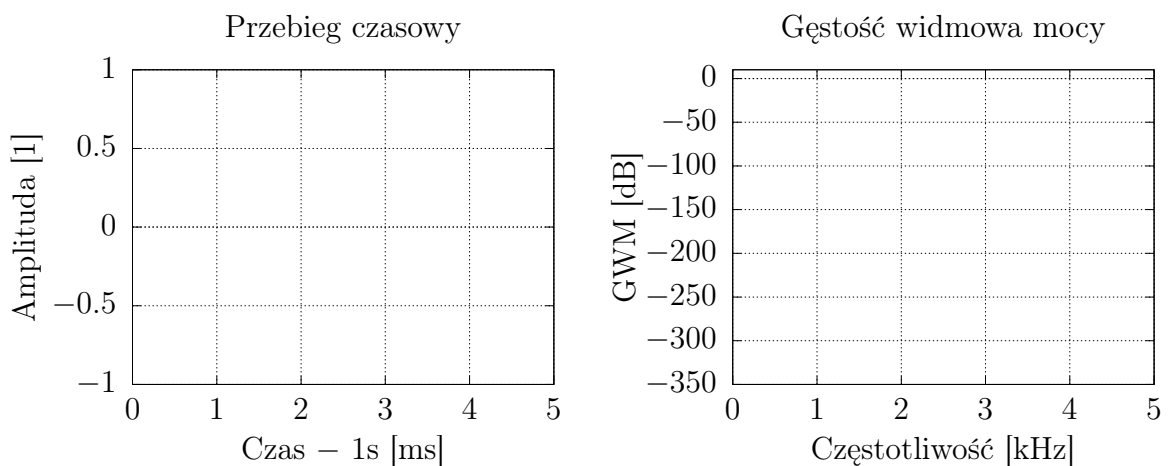
## 2 Decymacja sygnału

### 2.1 Zadanie

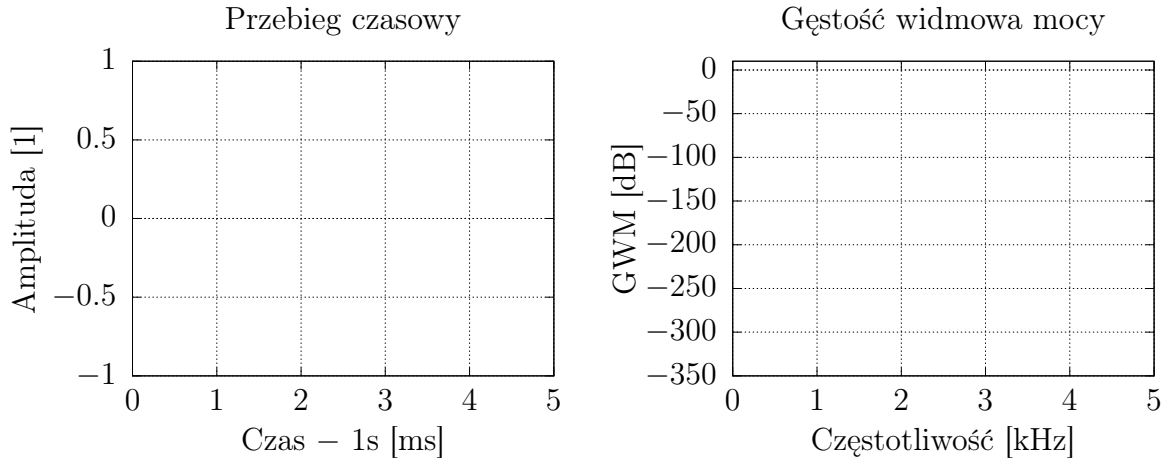
1. Przeprowadź decymację sygnału poprzez wybór co  $D$ -tej próbki z sygnału  $x_1$ , tworząc sygnał  $x_2$ . Decymację można przeprowadzić w języku Matlab za pomocą:  $x_2 = x_1(1:D:end);$ .
2. Zapisz wygenerowany sygnał do pliku 'x\_2'.
3. Wyznacz GWM z drugiej połowy sygnału oraz zapisz ją do pliku 'X\_2'.
4. Zapisz wnioski.

### 2.2 Rozwiązanie

Przebieg czasowy oraz GWM sygnału zostały umieszczone na rysunku 2. Porównanie przebiegów zostało umieszczone na rysunku 3.



Rysunek 2: Sygnał po decymacji.



Rysunek 3: Porównanie sygnału oryginalnego i po decymacji.

### 3 Projektowanie filtru

#### 3.1 Zadanie

Wyjaśnij jakie zjawisko jest powodem pojawiania się nowych częstotliwości w sygnale. Zaproponuj filtr cyfrowy  $H_1$  eliminujący to zjawisko, a zarazem nie mający istotnego wpływu na sygnał poprawnie odtwarzany. Przyjmij następujące kryteria:

- częstotliwości mniejsze niż  $\alpha \frac{f_s}{2D}$  mają być przenoszone z błędem nie większym jak  $R_p$  decybeli
- częstotliwości nie mniejsze niż  $\frac{f_s}{2D}$  mają być tłumione co najmniej  $R_s$  decybeli

Parametry  $\alpha$ ,  $R_p$  oraz  $R_s$  są podane przez prowadzącego.

Zastosuj filtr Butterwortha. Parametry w postaci wielomianowej  $H_1(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})}$  filtru Butterwortha  $n$ -tego rzędu, o znormalizowanej częstotliwości odcięcia  $f_c$  (wartość 1 to częstotliwość Nyquista) można wygenerować za pomocą:

```
[B, A] = butter(n, f_c);
```

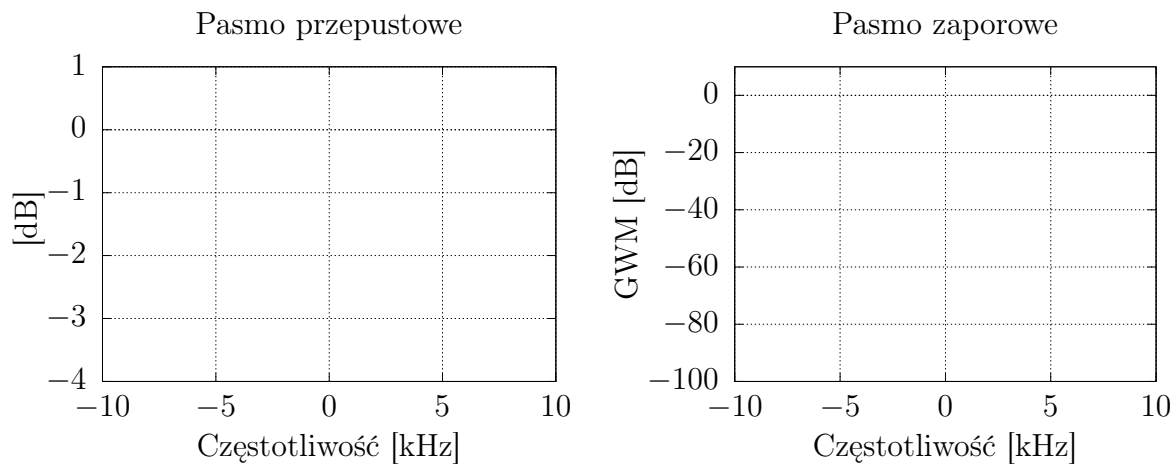
Do określenia minimalnego rzędu oraz częstotliwości odcięcia spełniające powyższe wymagania można się posłużyć funkcją `buttord()`.

```
[n, f_c] = buttord(alpha / D, 1 / D, R_p, R_s);
```

1. Wyznacz charakterystykę amplitudową filtru  $H_1$ , zapisz do pliku 'H\_1'.
2. Wyznacz opóźnienie w próbkach filtru  $H_1$ , zapisz do pliku 'Hd\_1'.
3. Przefiltruj sygnał  $x_1$  przez filtr  $H_1$ , tworząc sygnał  $x_3$ . Zapisz wygenerowany sygnał do pliku 'x\_3'. Wyznacz GWM z drugiej połowy sygnału oraz zapisz ją do pliku 'X\_3'.
4. Zapisz wnioski.

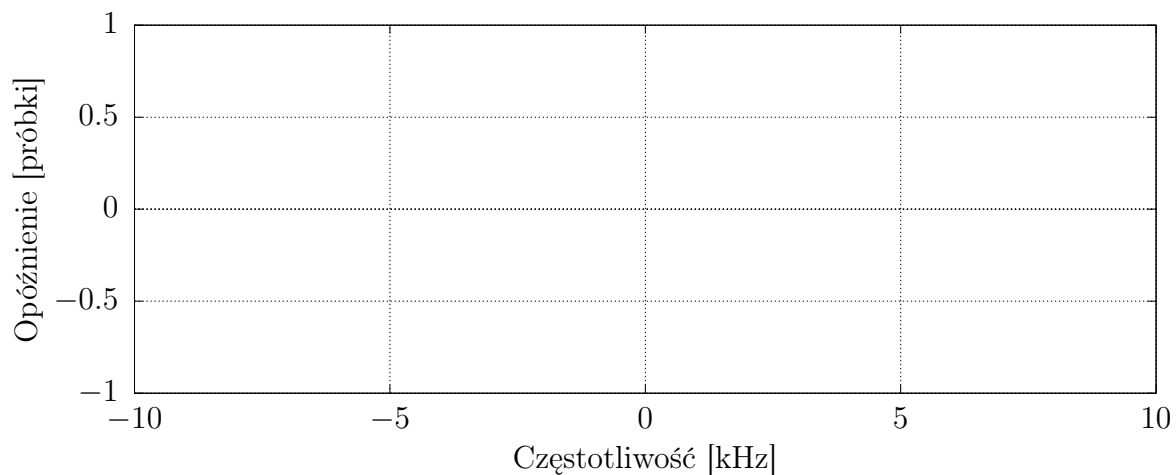
### 3.2 Rozwiązanie

Charakterystyki amplitudowe filtrów zostały umieszczone na rysunku 12.



Rysunek 4: Charakterystyki amplitudowe filtrów.

Opóźnienie filtru Butterwortha zostało umieszczone na rysunku 5.



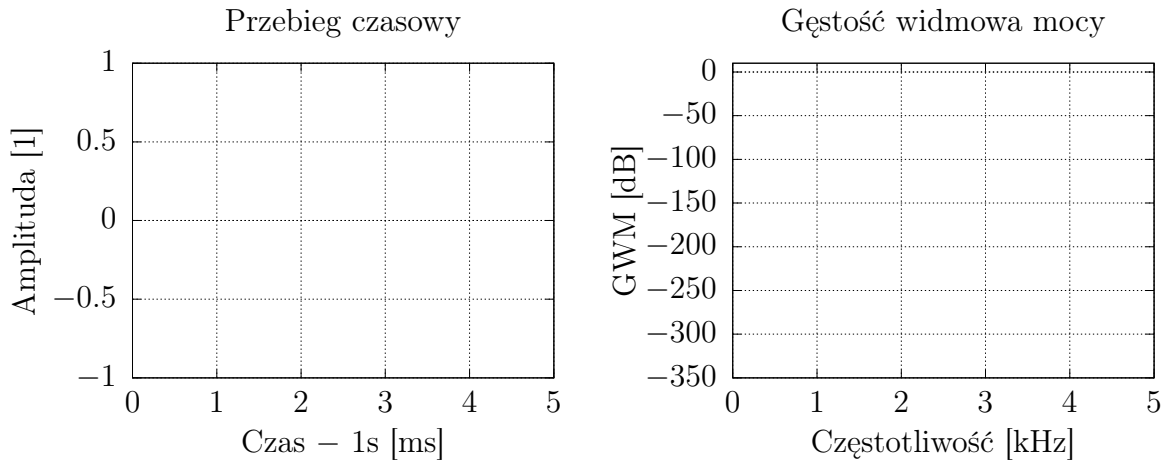
Rysunek 5: Opóźnienie filtru Butterwortha.

Przebieg czasowy oraz GWM sygnału zostały umieszczone na rysunku 6.

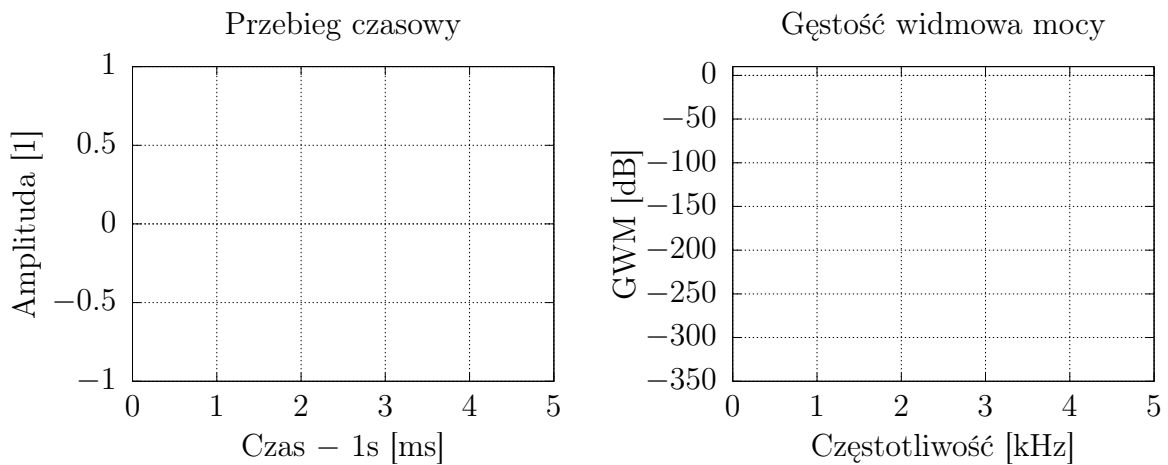
Porównanie przebiegów czasowych oraz GWM, sygnału oryginalnego i po filtracji, zostało umieszczone na rysunku 7.

## 4 Decymacja przefiltrowanego sygnału

1. Przeprowadź decymację sygnału poprzez wybór co  $D$ -tej próbki z sygnału  $x_3$ , tworząc sygnał  $x_4$ . Zapisz wygenerowany sygnał do pliku 'x\_4'. Wyznacz GWM z drugiej połowy sygnału oraz zapisz ją do pliku 'X\_4'.
2. Zapisz wnioski.



Rysunek 6: Sygnał przefiltrowany przez filtr  $H_1$ .



Rysunek 7: Porównanie sygnału oryginalnego i po filtracji przez filtr  $H_1$ .

## 4.1 Rozwiązanie

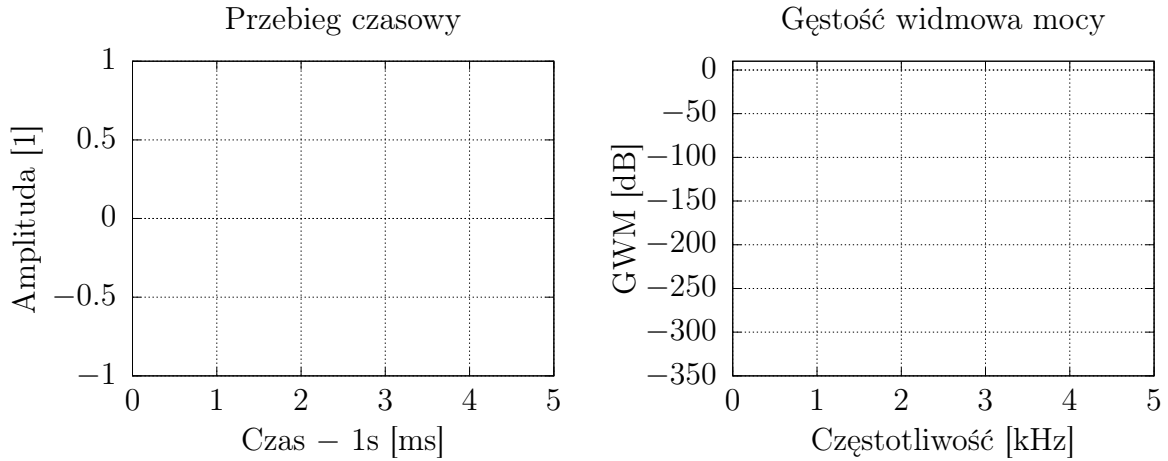
Przebieg czasowy oraz GWM sygnału zostały umieszczone na rysunku 8.

## 5 Rekonstrukcja sygnału

1. Zwiększ  $D$ -krotnie częstotliwość próbkowania sygnału  $x_4$  oraz zwiększ jego amplitudę  $D$ -krotnie uzyskując sygnał  $x_5$ .

Sygnał ten powinien mieć długość równą długości sygnału  $x_3$  (sygnał  $x_4$  przed decymacją) oraz powinien być uzupełniony zerami. Można to przeprowadzić w języku Matlab za pomocą:

```
x_5 = zeros(size(x_3));
x_5(1:N:end) = D * x_4;
```



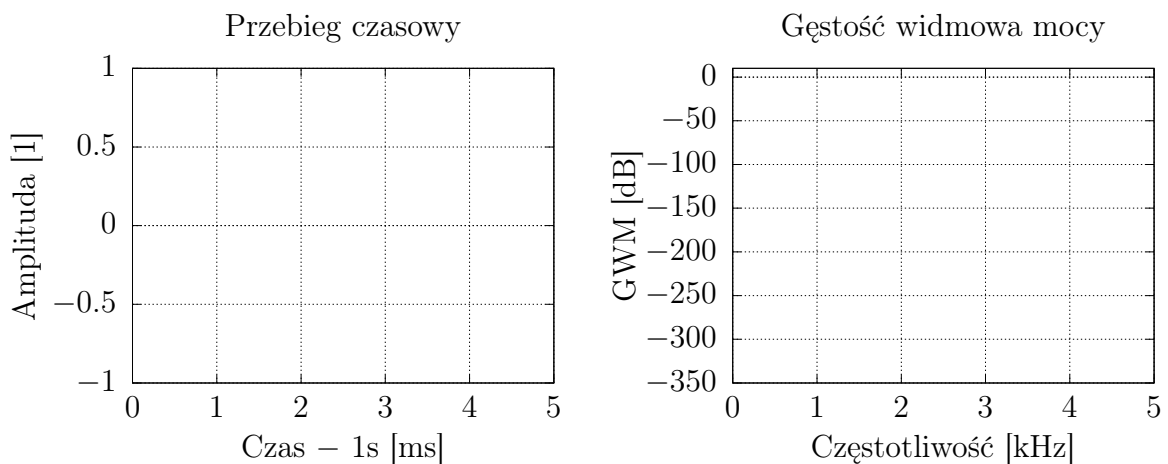
Rysunek 8: Przeszeregowany sygnał po decymacji.

2. Przeszereguj sygnał  $x_5$  przez filtr  $H_1$ , tworząc sygnał  $x_6$ .
3. Zapisz wygenerowany sygnał do pliku 'x\_6'.
4. Wyznacz GWM z drugiej połowy sygnału oraz zapisz ją do pliku 'X\_6'.
5. Zapisz wnioski.

🔗 Spróbuj skompensować opóźnienie generowane przez filtr rekonstrukcyjny.

## 5.1 Rozwiązanie

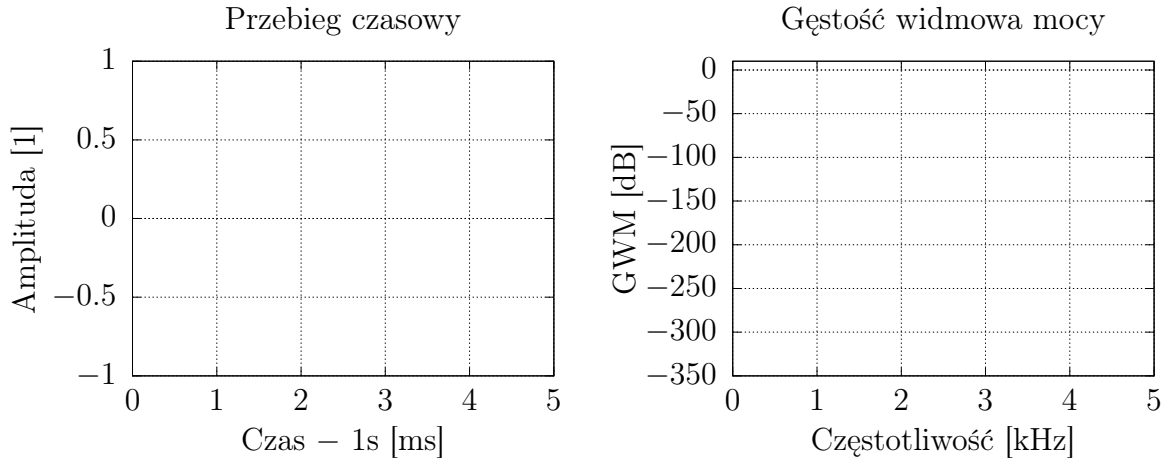
Przebieg czasowy oraz GWM sygnału  $x_5$  zostały umieszczone na rysunku 9. Przebieg czasowy oraz GWM sygnału  $x_6$  zostały umieszczone na rysunku 10.



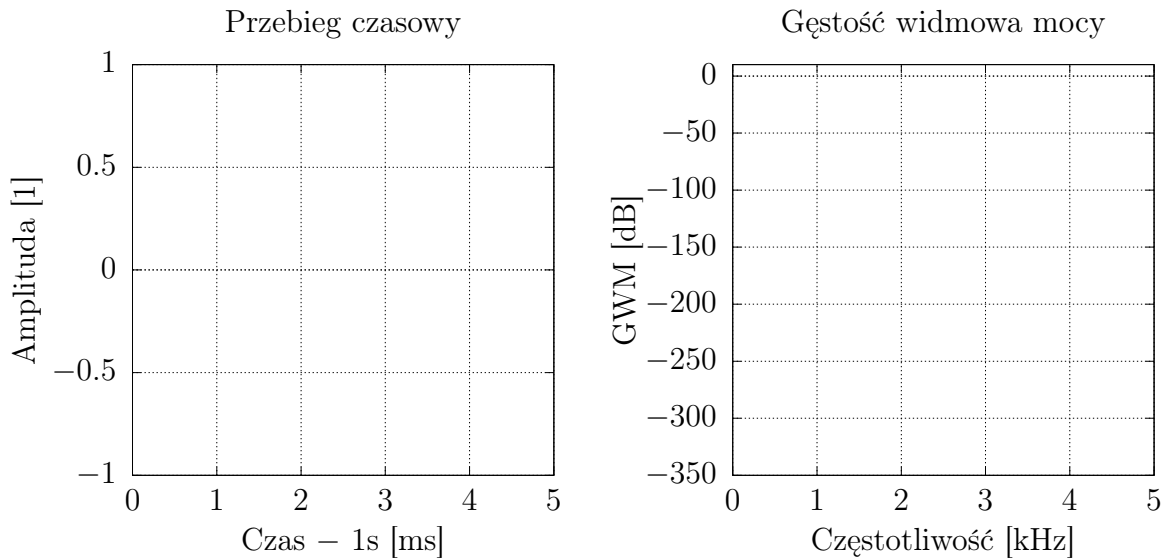
Rysunek 9: Sygnał zwiększeniu częstotliwości próbkowania

Porównanie sygnałów przed i po rekonstrukcji zostało przedstawione na rysunku 11.





Rysunek 10: Sygnał zwiększeniu częstotliwości próbkowania i filtracji.



Rysunek 11: Porównanie sygnałów przed i po rekonstrukcji

## 6 Filtr sinc

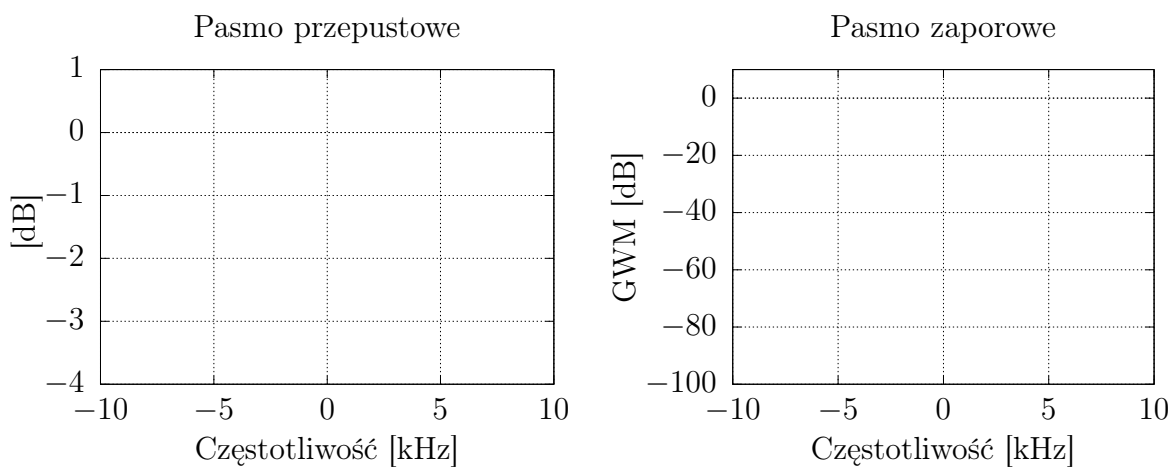
☞ Zastosuj alternatywnie filtr idealny („brick wall”, sinc). Parametry odpowiedzi impulsowej tego filtru można wyznaczyć za pomocą:  $i=(-M:M)'$ ;  $H = f_c * \text{sinc}(f_c * i)$ ; gdzie  $M$  to parametr określający liczbę parametrów filtru, filtr ma  $2M+1$  elementów odpowiedzi impulsowej.  $f_c$  to znormalizowana częstotliwość odcięcia filtru. Uzasadnij dobór  $f_c$  oraz  $M$ .

1. Wyznacz charakterystykę amplitudową filtru  $H_2$ , zapisz do pliku ‘H\_2’.
2. Wyznacz opóźnienie w próbkach filtru  $H_2$ , zapisz do pliku ‘Hd\_2’.
3. Przetnij sygnał  $x_1$  przez filtr  $H_2$ , tworząc sygnał  $x_3s$ . Zapisz wygenerowany sygnał do pliku ‘x\_3s’. Wyznacz GWM z drugiej połowy sygnału oraz zapisz ją do pliku ‘X\_3s’.
4. Przeprowadź decymację sygnału poprzez wybór co  $D$ -tej próbki z sygnału  $x_3s$ ,

tworząc sygnał  $x_{4s}$ . Zapisz wygenerowany sygnał do pliku ‘ $x_{4s}$ ’. Wyznacz GWM z drugiej połowy sygnału oraz zapisz ją do pliku ‘ $X_{4s}$ ’.

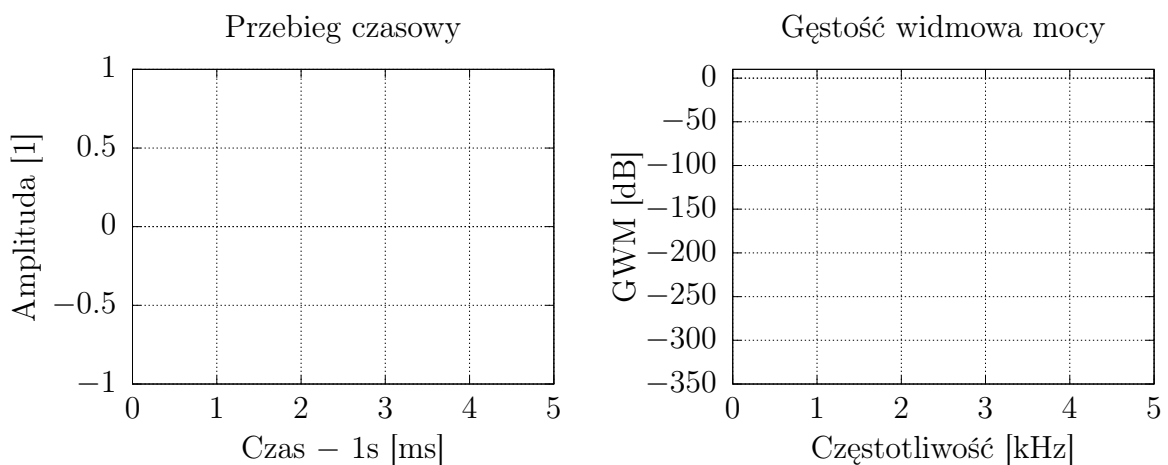
5. Zwiększ  $D$ -krotnie częstotliwość próbkowania sygnału  $x_{4s}$  oraz zwiększ jego amplitudę  $D$ -krotnie uzyskując sygnał  $x_{5s}$ .
6. Przefiltruj sygnał  $x_{5s}$  przez filtr  $H_2$ , tworząc sygnał  $x_{6s}$ .
7. Zapisz wygenerowany sygnał do pliku ‘ $x_{6s}$ ’.
8. Wyznacz GWM z drugiej połowy sygnału oraz zapisz ją do pliku ‘ $X_{6s}$ ’.
9. Spróbuj skompensować opóźnienie generowane przez filtr rekonstrukcyjny.

Charakterystyki amplitudowe filtrów zostały umieszczone na rysunku 12.



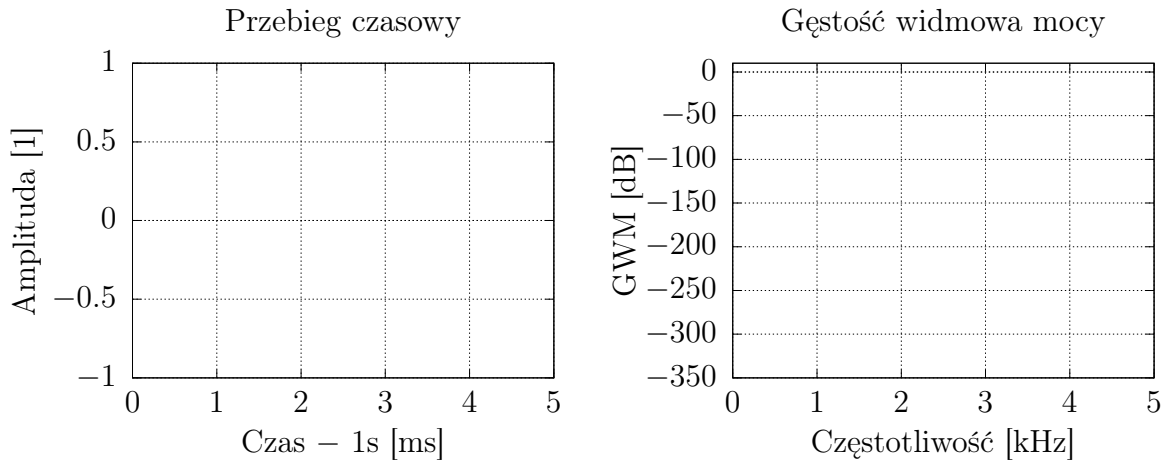
Rysunek 12: Charakterystyki amplitudowe filtrów.

Przebieg czasowy oraz GWM sygnału po filtracji zostały umieszczone na rysunku 13.

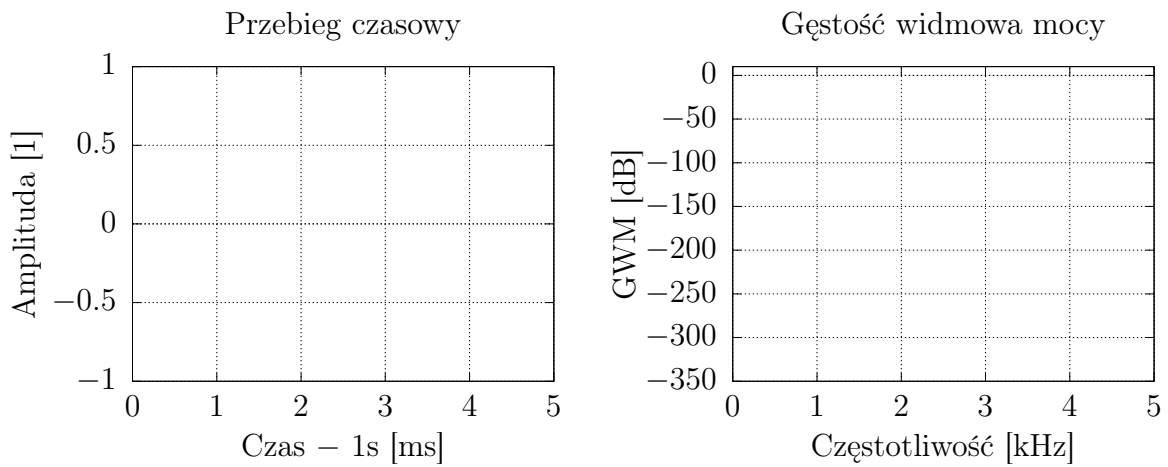


Rysunek 13: Sygnał przefiltrowany przez filtr  $H_2$ .

Przebieg czasowy oraz GWM sygnału po filtracji i decymacji zostały umieszczone na rysunku 14.



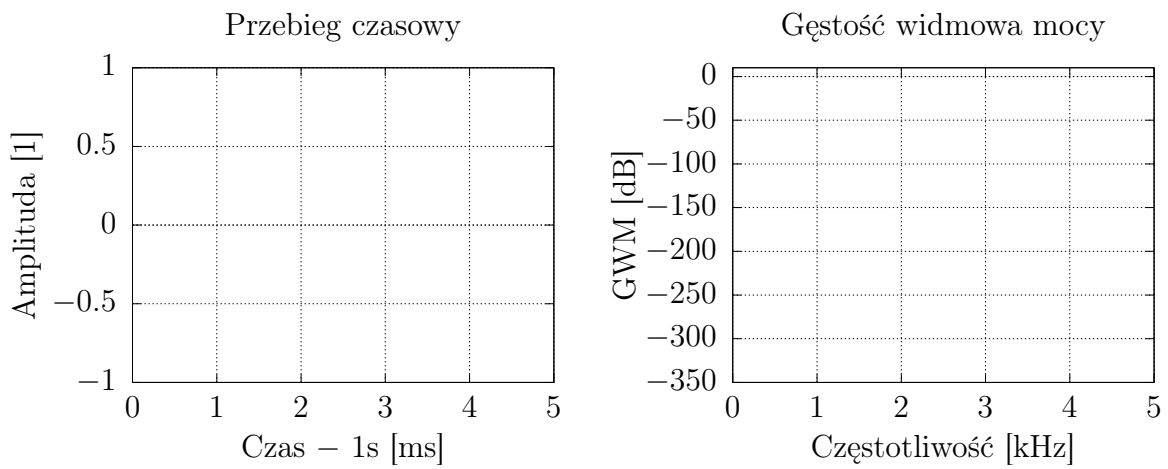
Rysunek 14: Przechwytywany sygnał po decymacji



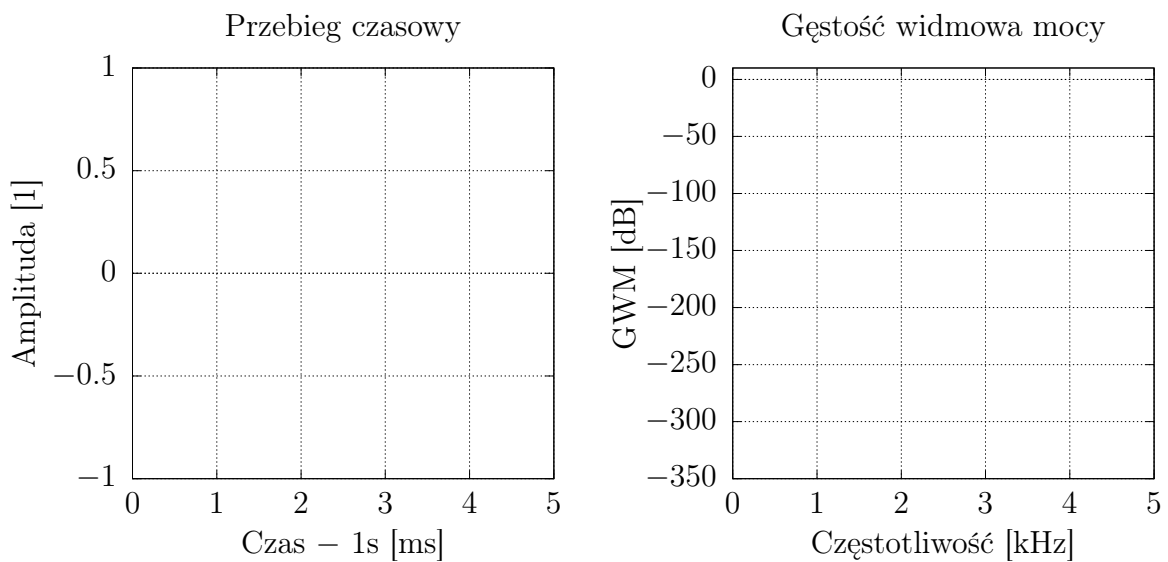
Rysunek 15: Sygnał zwiększeniu częstotliwości próbkowania.

Przebieg czasowy oraz GWM sygnału 'x\_5s' zostały umieszczone na rysunku 15. Przebieg czasowy oraz GWM sygnału 'x\_6s' zostały umieszczone na rysunku 16.

Porównanie sygnałów przed i po rekonstrukcji zostało przedstawione na rysunku 17.



Rysunek 16: Sygnał zwiększeniu częstotliwości próbkowania i filtracji.



Rysunek 17: Porównanie sygnałów przed i po rekonstrukcji