

Opis sygnału w dziedzinie czasu i częstotliwości, kwantyzacja sygnału

Jan Kowalski

5 grudnia 2019

Spis treści

1	Opis symulowanych sygnałów w dziedzinie czasu i częstotliwości	3
1.1	Teoria	3
1.2	Zadanie	3
1.2.1	Sygnał sinusoidalny	3
1.2.2	Sygnał prostokątny	3
1.2.3	Sygnał losowy	3
1.3	Rozwiązanie	3
2	Opis złożonego sygnału w dziedzinie czasu i częstotliwości	5
2.1	Zadanie	5
2.1.1	Sygnał 4	5
2.1.2	Sygnał 5	5
2.2	Rozwiązanie	5
3	Kwantyzacja	6
3.1	Teoria	6
3.2	Zadanie	6
3.2.1	Kwantyzacja bez ditheringu	6
3.2.2	Kwantyzacja z ditheringiem	7
3.3	Rozwiązanie	7

Spis rysunków

1	Sygnał sinusoidalny.	4
2	Sygnał prostokątny.	4
3	Sygnał losowy.	4
4	Sygnał 4.	5
5	Sygnał 5.	6
6	Kwantyzacja.	7

Jak ogarnąć środowisko w jakim pracujemy

Całość rozwiązania ma znajdować się w pliku 'sygnaly.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);
```

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
```

1 Opis symulowanych sygnałów w dziedzinie czasu i częstotliwości

1.1 Teoria

Gęstość widmową mocy (GWM) w decybelach obliczamy jako:

$$X_{dB}(j2\pi f) = X_{dB}\left(j2\pi f_s \frac{k}{N}\right) = 20 \log_{10} \left| \frac{\text{FFT}(x)}{N} \right|, \quad (1)$$

gdzie $f = f_s \frac{k}{N}$ to częstotliwość, f_s to częstotliwość próbkowania, k to numer binu (numer elementu wektora w wyniku, licząc od 0), N to liczba próbek sygnału x .

1.2 Zadanie

1.2.1 Sygnał sinusoidalny

1. Wygeneruj sygnał sinusoidalny o zadanej przez prowadzącego częstotliwości.
2. Zapisz wygenerowany sygnał do pliku 'x_1' (`writedata()`).
3. Wyznacz GWM sygnału, oraz zapisz ją do pliku 'X_1'.
4. Zapisz wnioski (%). Co widać na widmie, co być powinno.

1.2.2 Sygnał prostokątny

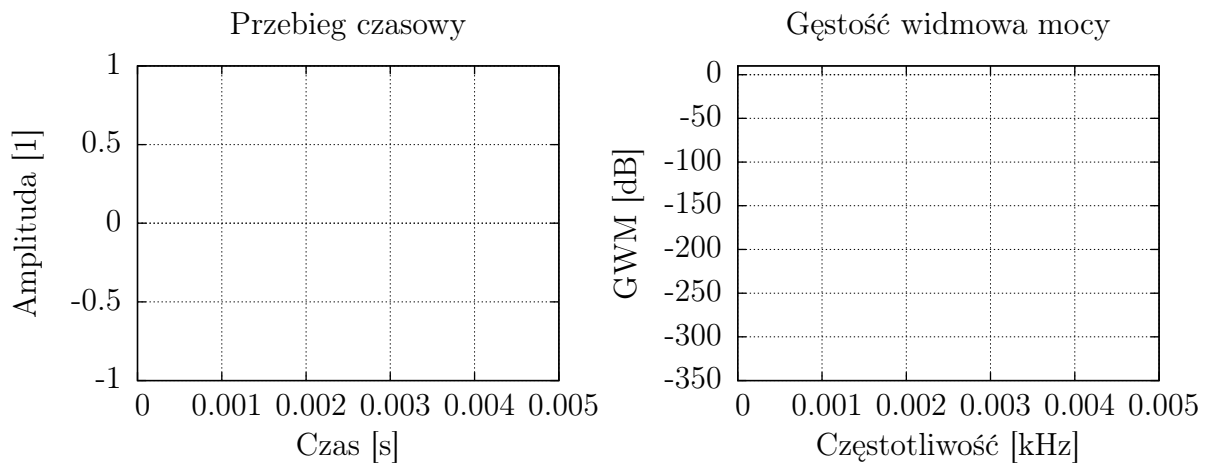
1. Wygeneruj sygnał prostokątny o zadanej przez prowadzącego częstotliwości.
2. Zapisz wygenerowany sygnał do pliku 'x_2'.
3. Wyznacz GWM sygnału, oraz zapisz ją do pliku 'X_2'.
4. Zapisz wnioski. Czy jest jakiś związek pomiędzy sygnałem sinusoidalnym i prostokątnym o tej samej częstotliwości? Czym widma tych sygnałów się różnią.

1.2.3 Sygnał losowy

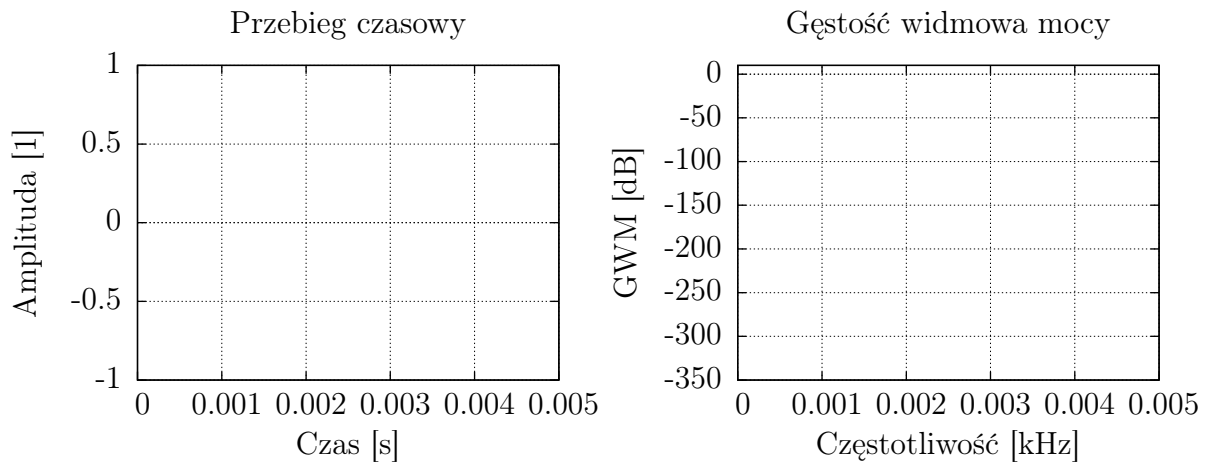
1. Wygeneruj sygnał losowy o zadanych przez prowadzącego właściwościach.
2. Zapisz wygenerowany sygnał do pliku 'x_3'.
3. Wyznacz GWM sygnału, oraz zapisz ją do pliku 'X_3'.
4. Zapisz wnioski.

1.3 Rozwiązanie

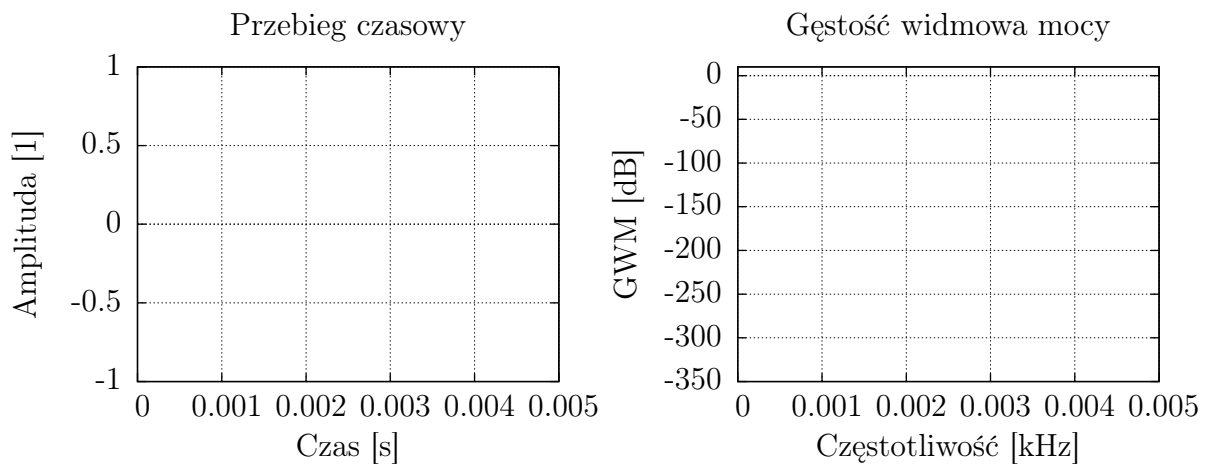
Przebieg czasowy oraz GWM sygnału sinusoidalnego zostały umieszczone na rysunku 1. Przebieg czasowy oraz GWM sygnału prostokątnego zostały umieszczone na rysunku 2. Przebieg czasowy oraz GWM szumu zostały umieszczone na rysunku 3.



Rysunek 1: Sygnał sinusoidalny.



Rysunek 2: Sygnał prostokątny.



Rysunek 3: Sygnał losowy.

2 Opis złożonego sygnału w dziedzinie czasu i częstotliwości

2.1 Zadanie

2.1.1 Sygnał 4

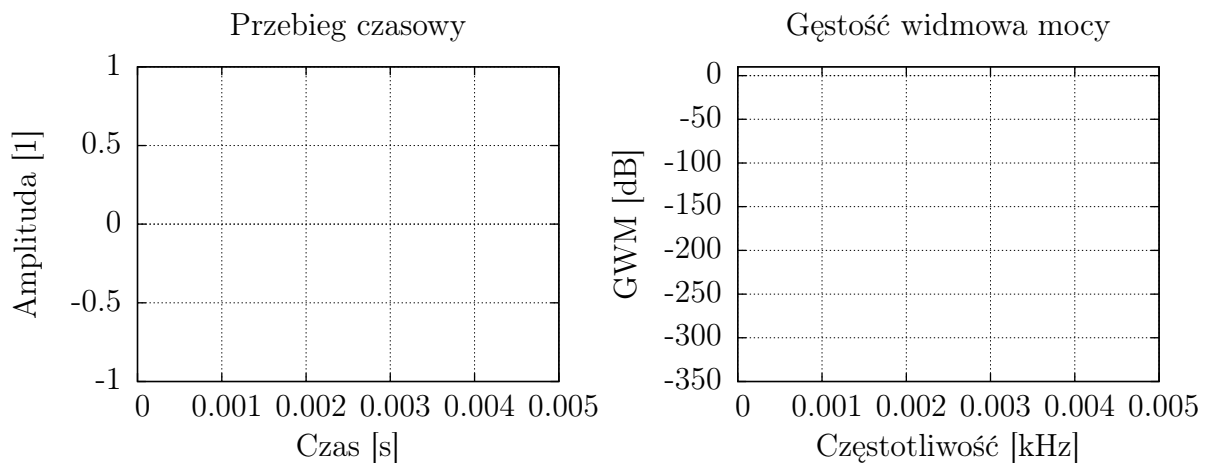
1. Odczytaj sygnał z pliku `dane/s4` oraz jego częstotliwość próbkowania z pliku `dane/s4.fs`.
2. Zapisz sygnał do pliku `'x_4'`.
3. Wyznacz GWM sygnału, oraz zapisz ją do pliku `'X_4'`.
4. Zapisz wnioski. Opisz z czego składa się sygnał.

2.1.2 Sygnał 5

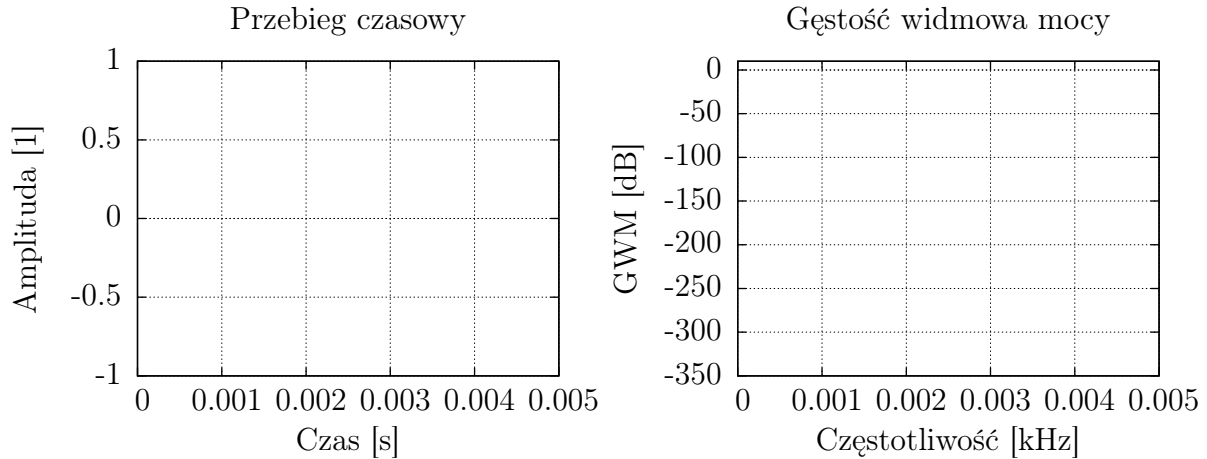
1. Odczytaj sygnał z pliku `dane/s5` oraz jego częstotliwość próbkowania z pliku `dane/s5.fs`.
2. Zapisz sygnał do pliku `'x_5'`.
3. Wyznacz GWM sygnału, oraz zapisz ją do pliku `'X_5'`.
4. Zapisz wnioski. Opisz z czego składa się sygnał.

2.2 Rozwiązanie

Przebieg czasowy oraz GWM sygnału 4 zostały umieszczone na rysunku 4. Przebieg czasowy oraz GWM sygnału 5 zostały umieszczone na rysunku 5.



Rysunek 4: Sygnał 4.



Rysunek 5: Sygnał 5.

3 Kwantyzacja

3.1 Teoria

Kwant przetwornika, q , to:

$$q = \frac{R}{2^b} \quad (2)$$

gdzie R to zakres przetwornika, a b to liczba bitów przetwornika.

Poprawna kwantyzacja przybliża wartości sygnału do najbliższego kwantu:

$$x_q = \text{round} \left(\frac{x}{q} \right) q \quad (3)$$

gdzie x_q to wartość skwantowana, a round to funkcja przybliżająca liczbę do najbliższej całkowitej. Kwantyzacja taka jest jednak w pełni deterministyczna, błąd kwantyzacji zależy od x i można go łatwo wyliczyć. By uniezależnić go od x stosuje się dithering (non-subtractive dithering):

$$x_q = \text{round} \left(\frac{x}{q} + d \right) q \quad (4)$$

gdzie d to dither. W przetwarzaniu sygnałów standardowy dither to dither losowy o rozkładzie trójkątnym w zakresie $[-1, 1]$. Zmienną losową o rozkładzie trójkątnym w takim zakresie można wygenerować odejmując dwie zmienne losowe o rozkładzie prostokątnym w zakresie $[0, 1]$.

3.2 Zadanie

3.2.1 Kwantyzacja bez ditheringu

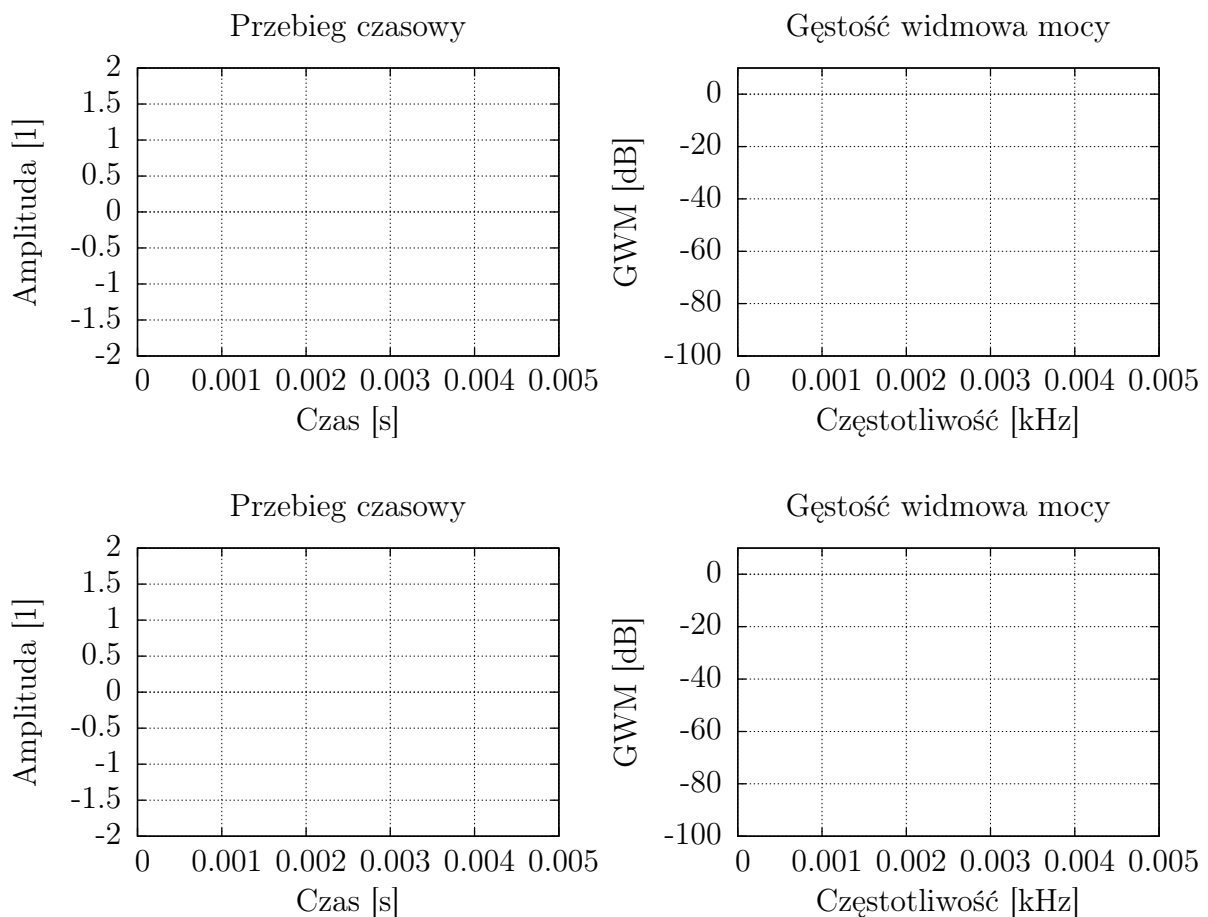
1. Przeprowadź kwantyzację sygnału sinusoidalnego z zadania 1 na 2, 4 i 8 bitach.
2. Zapisz sygnały kolejno do plików: 'x_1_2' (2 bit), 'x_1_4' (4 bit), 'x_1_8' (8 bit).
3. Wyznacz GWM sygnałów, oraz zapisz je kolejno do plików: 'X_1_2' (2 bit), 'X_1_4' (4 bit), 'X_1_8' (8 bit).
4. Zapisz wnioski.

3.2.2 Kwantyzacja z ditheringiem

1. Przeprowadź kwantyzację z ditheringiem sygnału sinusoidalnego z zadania 1 na 2, 4 i 8 bitach.
2. Zapisz sygnały kolejno do plików: 'x_1_2d' (2 bit), 'x_1_4d' (4 bit), 'x_1_8d' (8 bit).
3. Wyznacz GWM sygnałów, oraz zapisz je kolejno do plików: 'X_1_2d' (2 bit), 'X_1_4d' (4 bit), 'X_1_8d' (8 bit).
4. Zapisz wnioski.

3.3 Rozwiązanie

Przebiegi czasowe oraz GWM sygnałów po kwantyzacji zostały umieszczone na rysunku 6.



Rysunek 6: Kwantyzacja.