

Sprawozdanie z laboratorium 4. z przedmiotu TRA prowadzonego w semestrze 24Z

Piotr Sienkiewicz 324 887

1 Zadanie 1

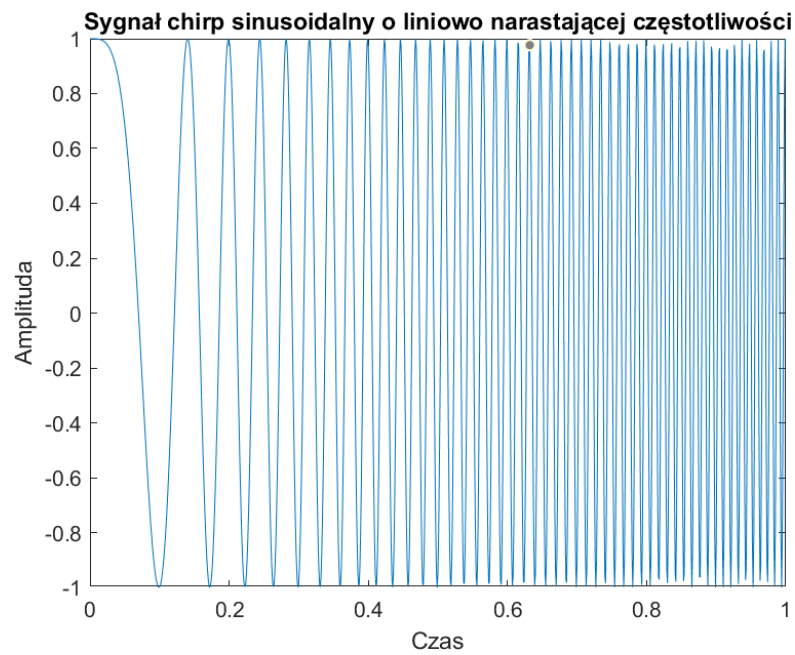
W pierwszym zadaniu, aby wygenerować sygnał sinusoidalny o liniowo rosnącej częstotliwości, wykorzystano funkcję `chirp` w programie MATLAB:

Listing 1: Skrypt MATLAB do zadania 1.

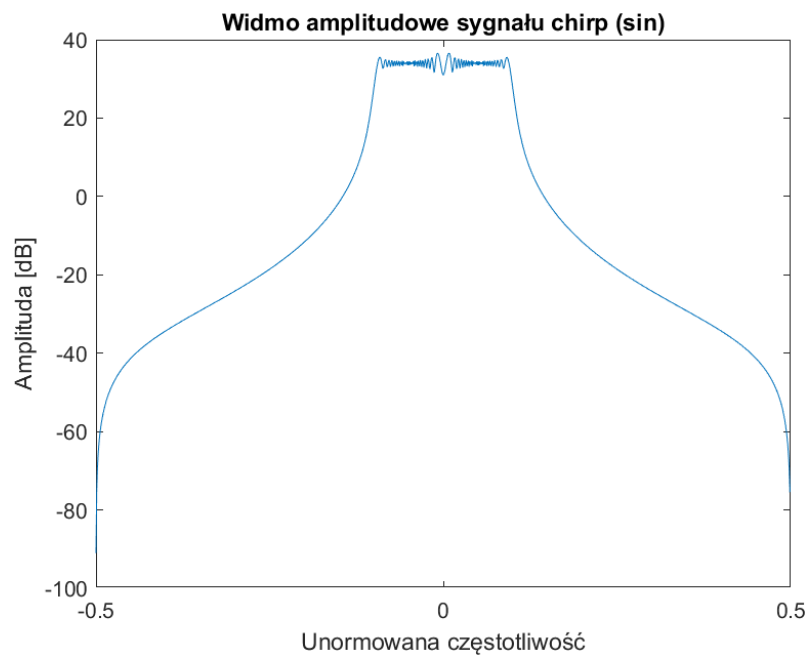
```
1 clear all
2
3 % Parametry sygnału
4 N = 1000;
5 f0_norm = 0;
6 f1_norm = 0.1;
7 t = (0:N-1) / N;
8
9 x1 = chirp(1:N, f0_norm, N, f1_norm);
10
11 % Wyświetlanie sygnału
12 figure;
13 plot(t, x1);
14 title('Sygnał chirp sinusoidalny o liniowo narastającej częstotliwości');
15 xlabel('Czas');
16 ylabel('Amplituda');
17
18 % Obliczanie widma sygnału
19 % z użyciem plotspec() - skala logarytmiczna
20 figure
21 plotspec(x1);
22 title('Widmo amplitudowe sygnału chirp (sin)');
23 xlabel('Unormowana częstotliwość');
24 ylabel('Amplituda [dB]');
25
26 % z użyciem fft() - skala liniowa
27 y1_fft = fft(x1);
28 frequencies = (-N/2:N/2 - 1) / N;
29 figure
30 plot(frequencies, abs(fftshift(y1_fft)));
31 title('Widmo amplitudowe sygnału chirp (sin)');
32 xlabel('Unormowana częstotliwość');
33 ylabel('Amplituda [lin]');
```

Funkcja `chirp` przyjmuje jako argumenty wektor zawierający kolejne numery próbek, częstotliwość początkową (unormowaną), oraz próbkę `N`, przy której sygnał ma już osiągnąć częstotliwość `f1_norm`. Następnie program rysuje przebieg sinusoidy w czasie, dodatkowo normalizując czas, aby przyjmował wartości od 0 do 1. Wyznaczane jest też widmo z użyciem funkcji `plotspec(x1)` oraz z użyciem funkcji `fft()`. W tym drugim przypadku należy dodatkowo dokonać przesunięcia widma, aby uzyskać widmo zespolone odpowiednio pokrywające się na osi OX wykresu. Dodatkową różnicą jest to, że `fft()` zwraca wartości w skali liniowej, a nie w decybelowej.

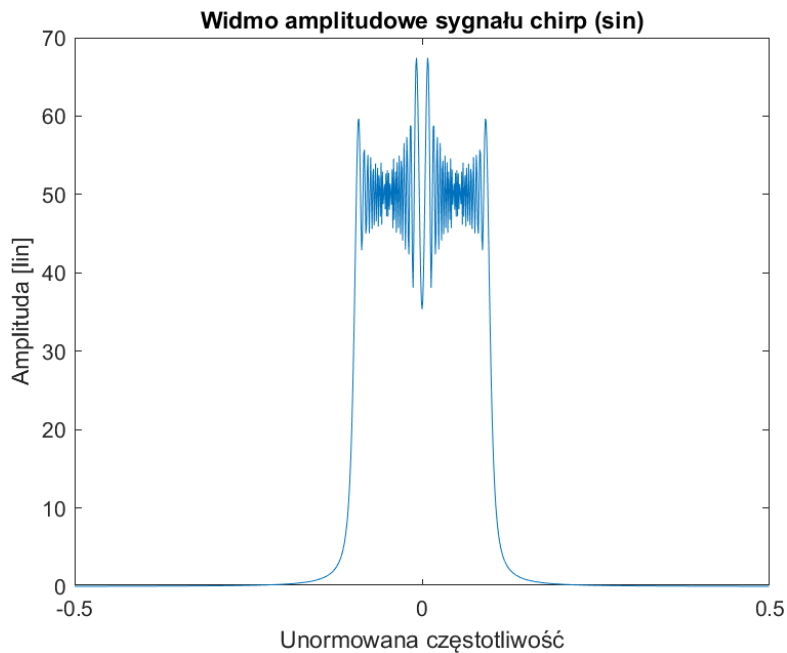
Za pomocą powyższego programu otrzymano następujące wykresy:



Rysunek 1: Przebieg sygnału w czasie



Rysunek 2: Widmo sygnału uzyskane za pomocą funkcji `plotspec()`



Rysunek 3: Widmo sygnału uzyskane za pomocą funkcji `fft()`

Jak widać, widmo takiego sygnału nieokresowego składającego się z przebiegu sinusoidalnego o rosnącej częstotliwości zawiera składowe częstotliwości takie, jakie przyjmował sygnał podstawowy. Dla wyższych częstotliwości widmo bardzo szybko maleje.

1.1 Zadanie 2

Tym razem także wygenerowano sygnał sinusoidalny o liniowo rosnącej częstotliwości, ale tym razem wedle zaleceń prowadzącego wygenerowano sygnał o wartościach zespolonych, używając tej samej funkcji co ostatnio, ale tym razem z dodatkowym parametrem:

Listing 2: Skrypt MATLAB do zadania 2.

```

1 clear all
2
3 % Parametry sygnału
4 N = 1000;
5 f0_norm = 0;
6 f1_norm = 0.1;
7 t = (0:N-1) / N;
8
9 x1 = chirp(1:N, f0_norm, N, f1_norm, 'linear', 'complex');
10
11 % Wyświetlanie sygnału
12 figure;
13 plot(t, real(x1));
14 title('Sygnał chirp sinusoidalny o liniowo narastającej częstotliwości');
15 xlabel('Czas');
16 ylabel('Amplituda');
17
18 % Obliczanie widma sygnału

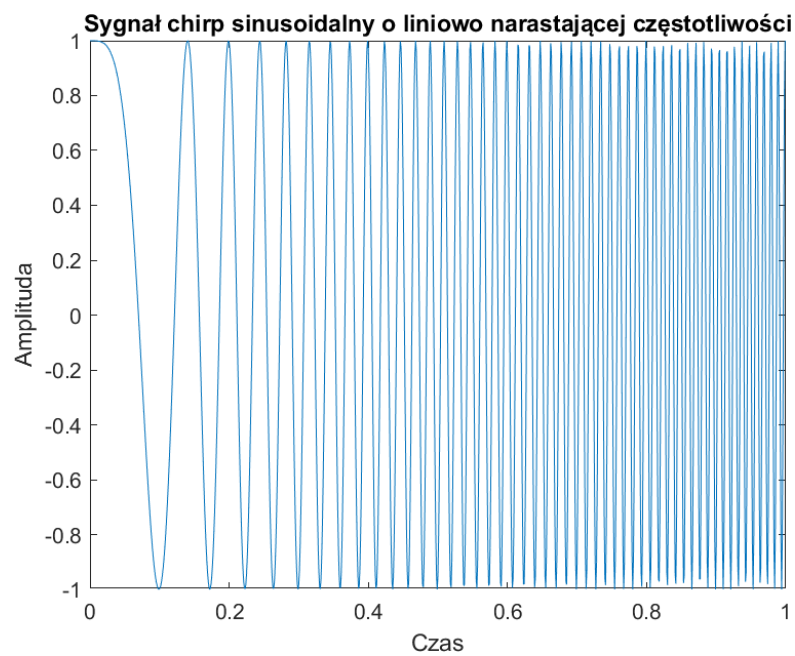
```

```

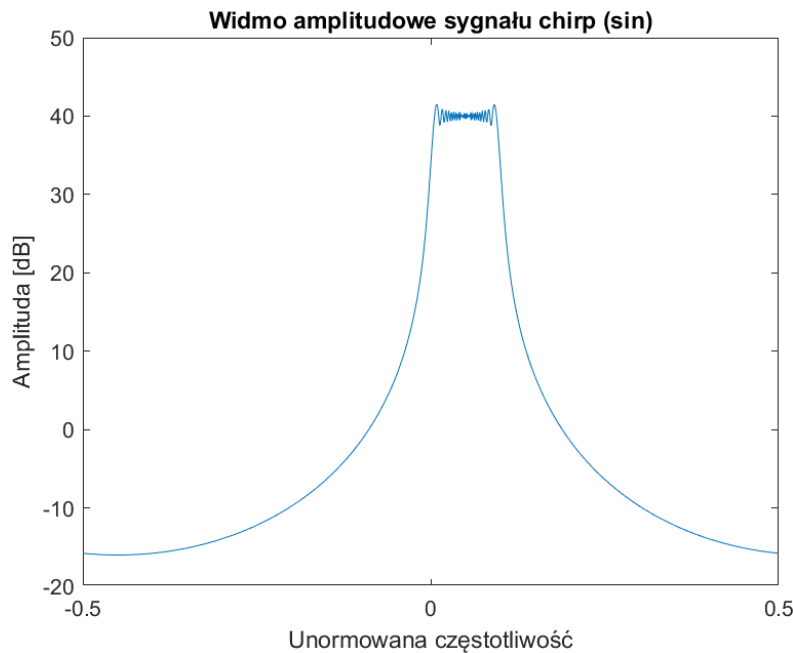
19 figure
20 plotspec(x1);
21 title('Widmo amplitudowe sygnału chirp (sin)');
22 xlabel('Unormowana czestotliwosc');
23 ylabel('Amplituda [dB]');

```

Otrzymano następujące wykresy:



Rysunek 4: Przebieg sygnału w czasie



Rysunek 5: Widmo sygnału uzyskane za pomocą funkcji `plotspec()`

Przebieg sygnału w czasie nie zmienił się względem poprzedniego zadania, ponieważ narysowano tylko rzeczywistą część sygnału. W tym przypadku, natomiast, zmieniło się widmo, gdyż opisuje ono tylko część rzeczywistą sygnału. Jest ono węższe oraz niesymetryczne względem 0 (dla częstotliwości unormowanej równej 0 dopiero zaczyna się pik wartości, podczas gdy w poprzednim przykładzie był on symetryczny względem zera). Jednak gdy zrobić sprzężenie widma i dodać je do otrzymanego w tym przykładzie, otrzymałoby się widmo sygnału rzeczywistego z zadania 1. Pokazuje to, jak w tym przypadku moc sygnału jest podzielona po równo na moc sygnału rzeczywistego i urojonego.

2 Zadanie 3

W tym zadaniu powielono sygnał z zadania 2. pięciokrotnie za pomocą funkcji `repmat()`, a następnie wygenerowano jego widmo:

Listing 3: Skrypt MATLAB do zadania 3.

```

1 clear all
2
3 % Parametry sygnału
4 N = 1000;
5 f0_norm = 0;
6 f1_norm = 0.1;
7
8 repetitions = 5;
9
10 t = (0:repetitions*N-1) / N;
11
12 x1 = chirp(1:N, f0_norm, N, f1_norm, 'linear', 'complex');
13 x2 = repmat(x1, 1, repetitions);
14
15 % Wyświetlanie sygnału

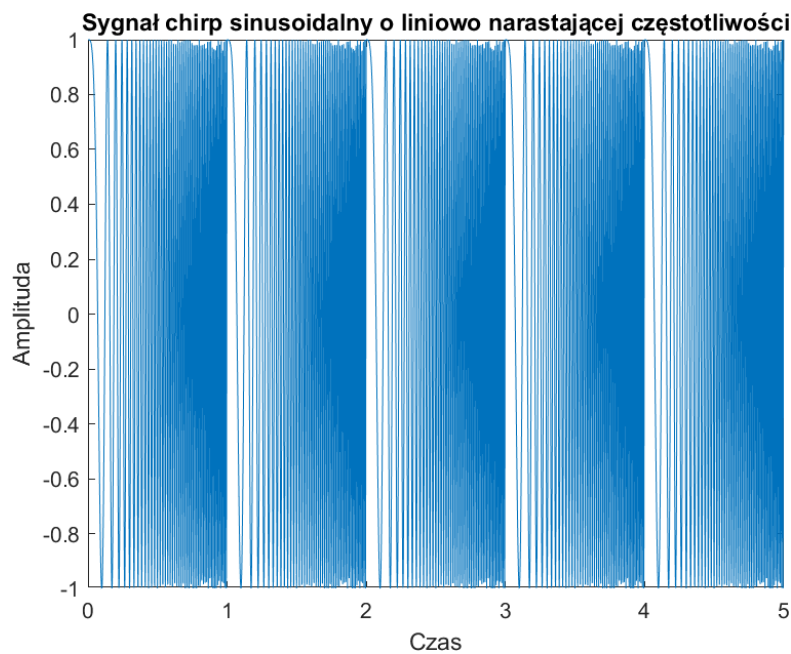
```

```

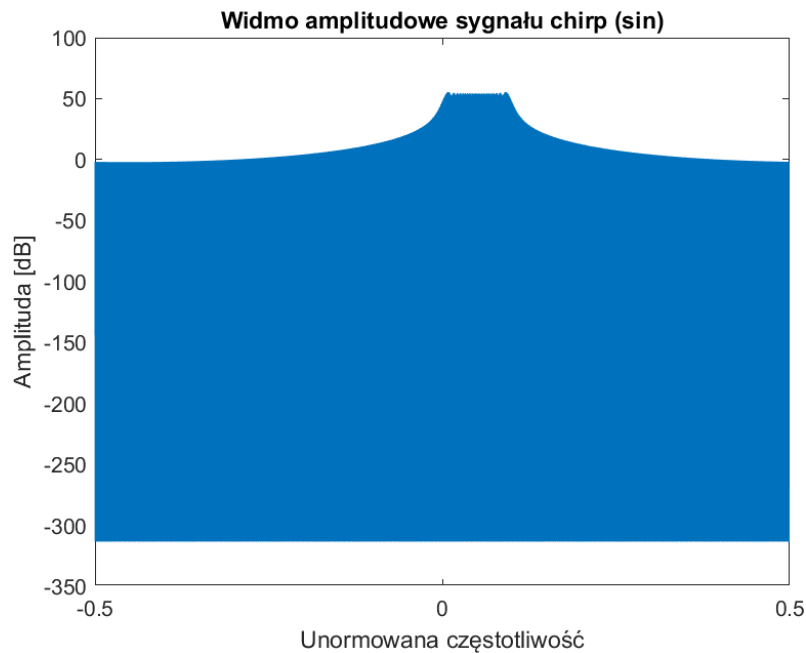
16 figure;
17 plot(t, real(x2));
18 title('Sygnał chirp sinusoidalny o liniowo narastającej częstotliwości');
19 xlabel('Czas');
20 ylabel('Amplituda');
21
22 % Obliczanie widma sygnału
23 figure
24 plotspec(x2);
25 title('Widmo amplitudowe sygnału chirp (sin)');
26 xlabel('Unormowana częstotliwość');
27 ylabel('Amplituda [dB]');

```

Otrzymano następujące wykresy:



Rysunek 6: Przebieg sygnału w czasie



Rysunek 7: Widmo sygnału uzyskane za pomocą funkcji `plotspec()`

Od razu widać, jak bardzo zmieniło się widmo sygnału. Przypomina ono widmo ciągłe, chociaż w powiększeniu widać, że składa się z bardzo gęsto ułożonych prążków. Wynik eksperymentu był do przewidzenia, gdyż widmo sygnału okresowego o zmiennej częstotliwości jest widmem ciągłym. W przypadku zwiększenia ilości powtórzeń sygnału, ilość prążków wzrosła, zwłaszcza pojawiły się prążki o bardzo małej wartości.

3 Zadanie 4.

W zadaniu tym stworzono filtr dopasowany do sygnału `x1`, czyli sinusoidy o zmiennej częstotliwości takiej jak w zadaniu 2:

Listing 4: Skrypt MATLAB do zadania 4.

```

1 clear all
2
3 % Parametry sygnału
4 N = 1000;
5 f0_norm = 0;
6 f1_norm = 0.1;
7
8 repetitions = 5;
9
10 t = (0:repetitions*N-1) / N;
11
12 x1 = chirp(1:N, f0_norm, N, f1_norm, 'linear', 'complex');
13 x2 = repmat(x1, 1, repetitions);
14
15 % Filtr dopasowany
16 B = conj(x1(end:-1:1));
17 y = filter(B, 1, x2);
18

```

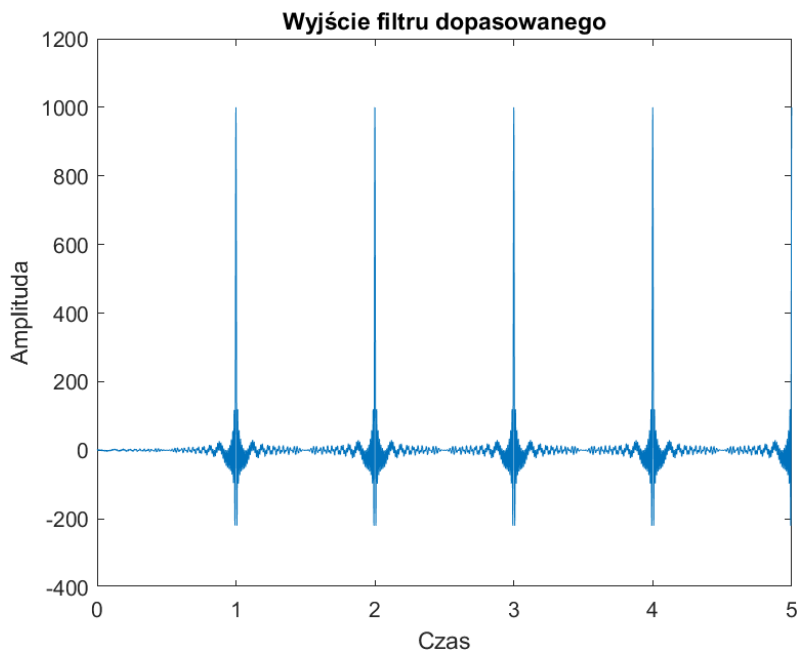


```

19 % Wyświetlanie wyjścia filtra dopasowanego
20 figure;
21 plot(t, real(y));
22 title('Wyjście filtra dopasowanego');
23 ylabel('Amplituda'); % równa energii sekwencji sygnału
24 xlabel('Czas');

```

Filtr dopasowany analizuje sygnał i wykrywa w nim wystąpienie sygnału, dla którego został zaprojektowany. Jego współczynniki to tak w zasadzie wektor sprzężeń z odwróconego wektora poszukiwanego sygnału, dlatego też w tym przypadku filtr ten ma aż 1000 współczynników. Sygnał na wyjściu tego filtra jest następujący:



Rysunek 8: Wyjście filtra dopasowanego

Powyższy sygnał pokazuje poprawne wykrywanie wystąpienia kolejnych sekwencji sygnału x_1 w sygnale powielonym x_2 . Gdy dana sekwencja została wykryta, na wyjściu filtra pojawia się pik o wartości równej energii sygnału szukanej sekwencji. W tym przypadku piki mają wartość 1000, co by się zgadzało, bo kwadrat amplitudy sygnału wejściowego wynosi 1, a liczba próbek danej sekwencji wynosi 1000.

4 Zadanie 5

W tym zadaniu dodano do sygnału szum:

Listing 5: Skrypt MATLAB do zadania 5.

```

1 clear all
2
3 % Parametry sygnału
4 N = 1000;
5 f0_norm = 0;
6 f1_norm = 0.1;

```

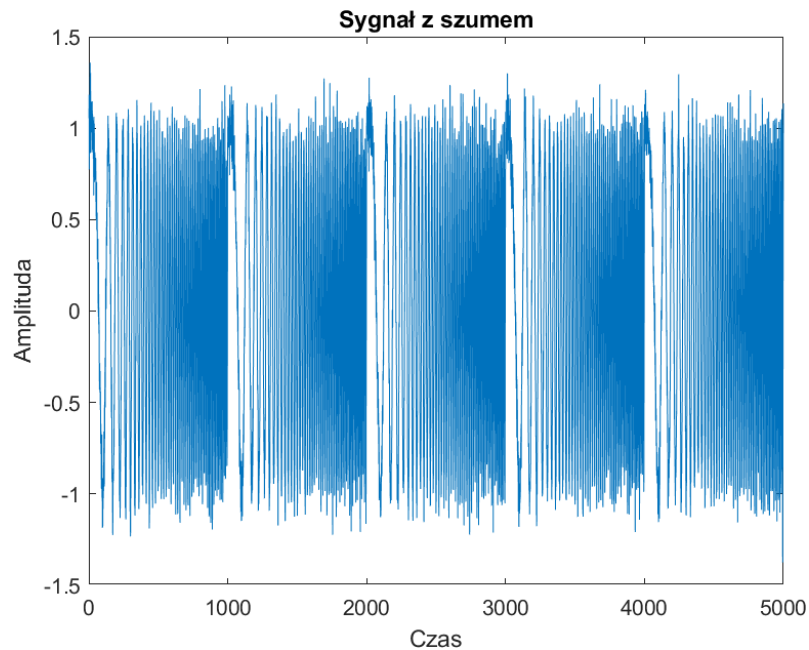
```

7 repetitions = 5;
8 noise_amplitude = 8; %0.1, 1, 5, 8
9
10 t = (0:repetitions*N-1) / N;
11
12 x1 = chirp(1:N, f0_norm, N, f1_norm, 'linear', 'complex');
13 x2 = repmat(x1, 1, repetitions);
14 noise = complex(randn(size(x2)), randn(size(x2)));
15 x2_with_noise = x2 + noise_amplitude*noise;
16
17 % Filtr dopasowany
18 B = conj(x1(end:-1:1));
19 y = filter(B, 1, x2_with_noise);
20
21 % Wyświetlanie wyjścia filtru dopasowanego
22 figure;
23 plot(t, real(y));
24 title('Wyjście filtru dopasowanego');
25 ylabel('Amplituda');
26 xlabel('Czas');

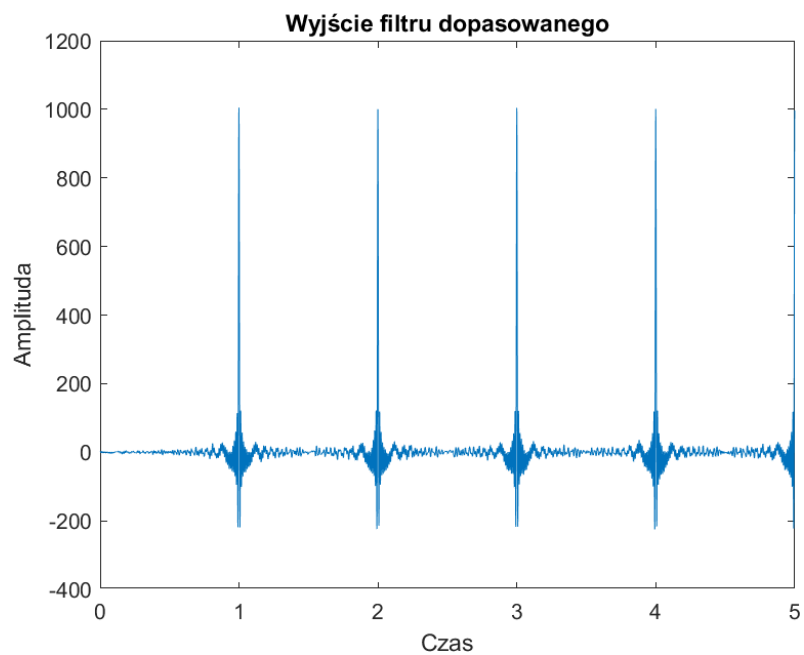
```

Podczas eksperymentu zwiększano amplitudę szumu przy zachowanej amplitudzie sygnału wynoszącej 1:

Amplituda szumu 0,1

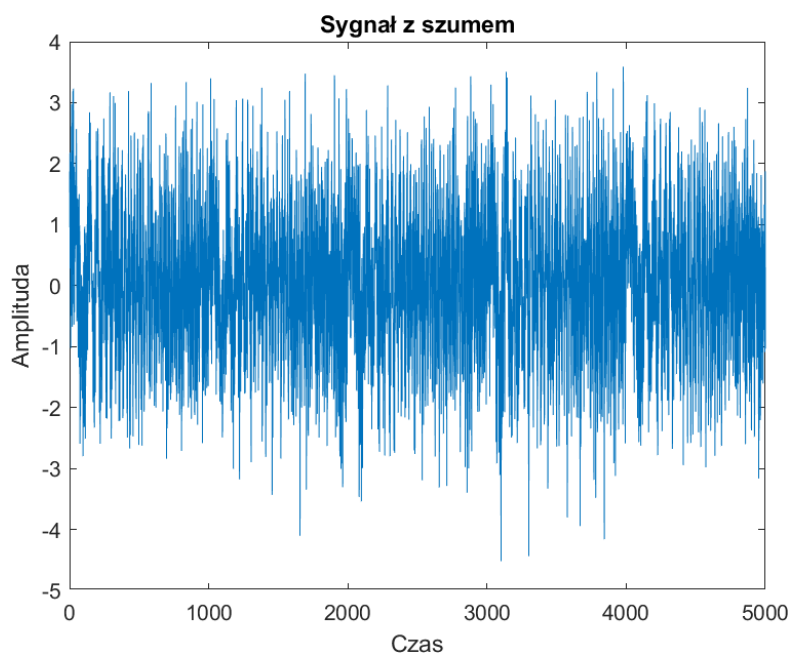


Rysunek 9: Przebieg sygnału z szumem o amplitudzie 0,1

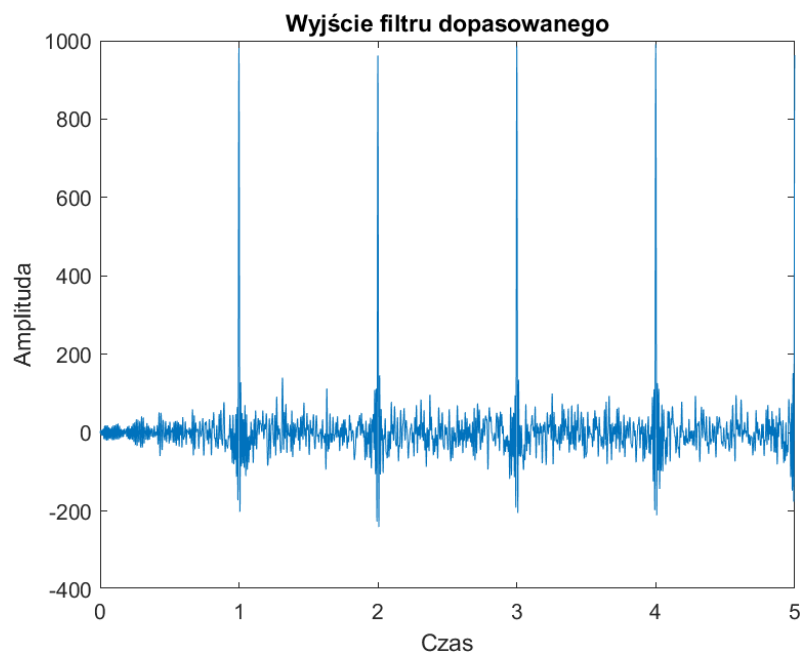


Rysunek 10: Wyjście filtru dopasowanego dla sygnału z szumem o amplitudzie 0,1

Amplituda szumu 1

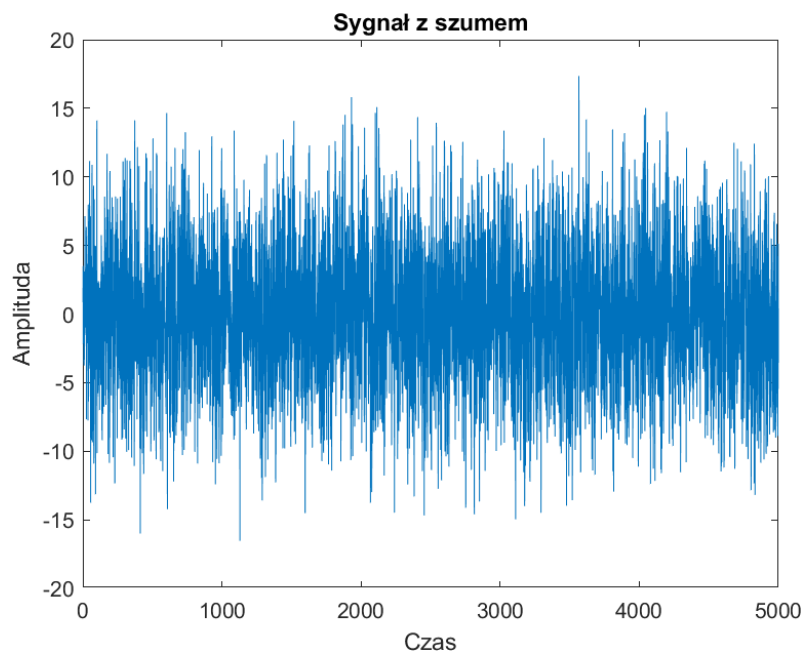


Rysunek 11: Przebieg sygnału z szumem o amplitudzie 1

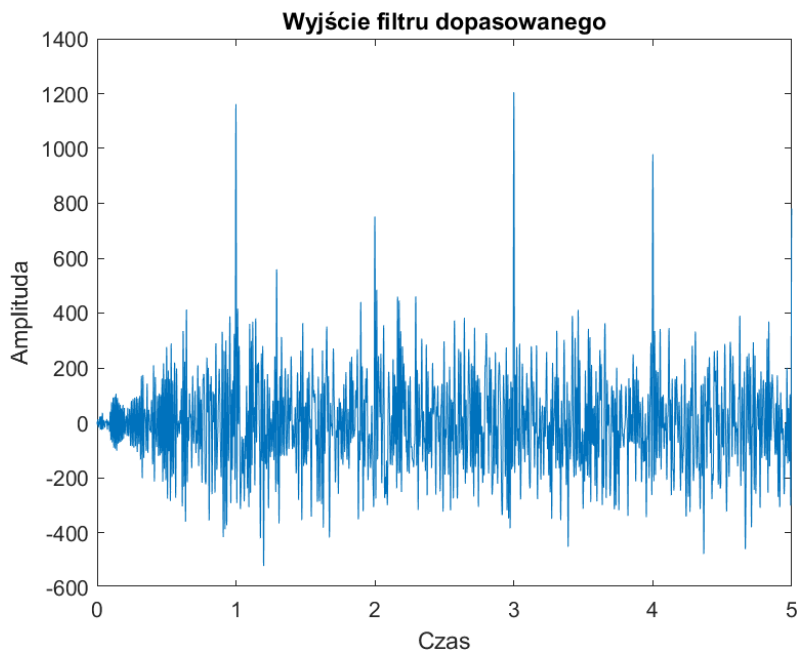


Rysunek 12: Wyjście filtru dopasowanego dla sygnału z szumem o amplitudzie 1

Amplituda szumu 5

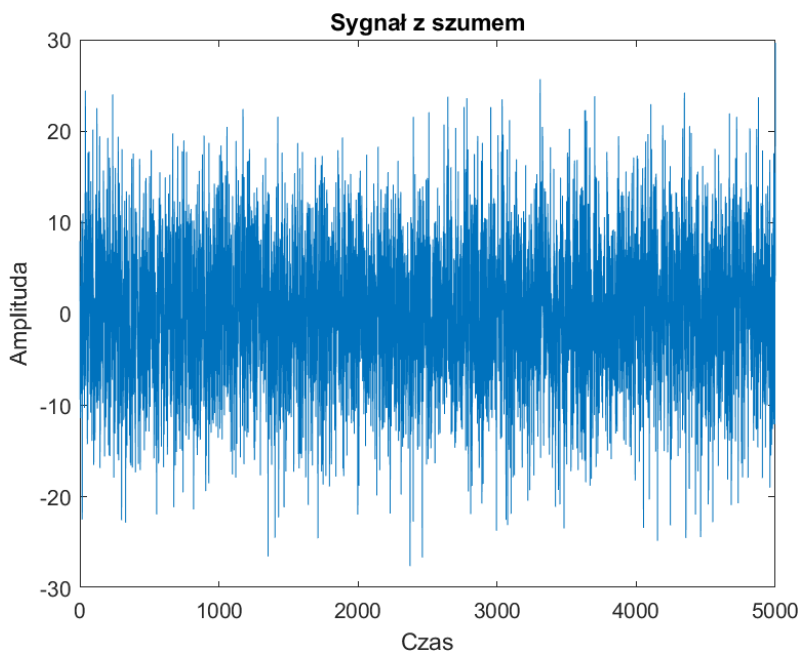


Rysunek 13: Przebieg sygnału z szumem o amplitudzie 5

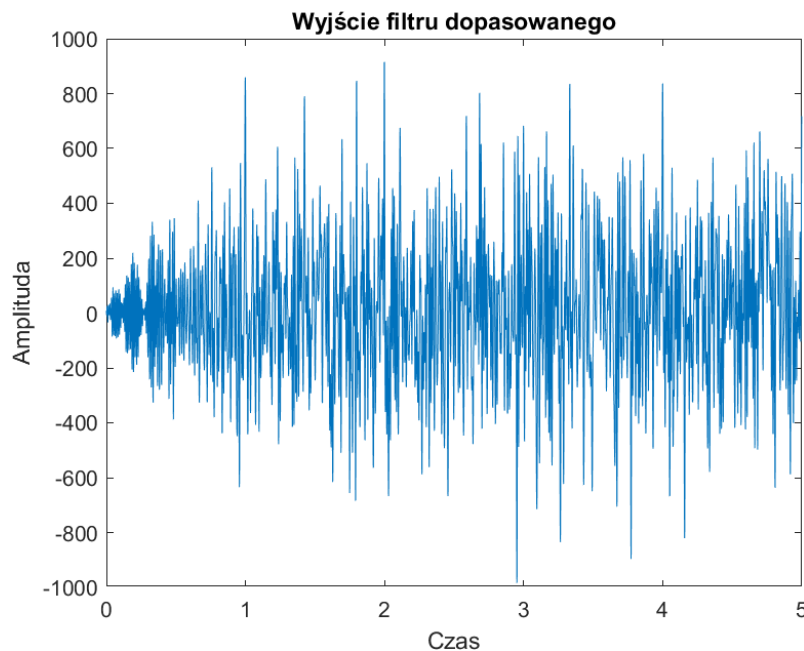


Rysunek 14: Wyjście filtru dopasowanego dla sygnału z szumem o amplitudzie 5

Amplituda szumu 8



Rysunek 15: Przebieg sygnału z szumem o amplitudzie 8

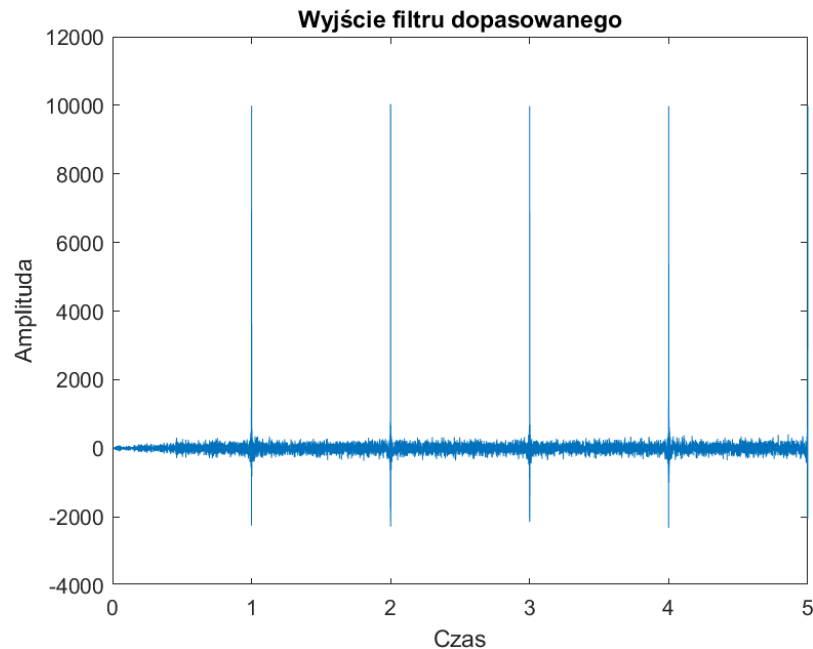


Rysunek 16: Wyjście filtru dopasowanego dla sygnału z szumem o amplitudzie 8

Powyższe wykresy pokazują, jak wspaniale działa filtr dopasowany. Nawet przy amplitudzie szumów 5 większej niż amplituda sygnału pozwala on na poprawne wykrycie danej sekwencji w sygnale. Dopiero przy amplitudzie szumu wynoszącej 8 ta detekcja nie jest jednoznaczna. Warto spojrzeć też na przebiegi sygnałów w czasie. Jeszcze dla wartości szumów 0,1 oraz 1 da się w jakiś sposób oszacować, gdzie mogła wystąpić dana sekwencja, natomiast na późniejszych przebiegach widać jedynie szum. To pokazuje, jak użyteczne są filtry dopasowane. Używa się ich często w telekomunikacji do wykrywania sygnałów taktujących lub sekwencji startowych i końcowych transmitowanych sygnałów.

5 Zadanie 6

W tym zadaniu zwiększono liczbę próbek do 10000 przy amplitudzie sygnału i szumu wynoszących 1. Otrzymano w ten sposób taką odpowiedź filtru:



Rysunek 17: Wyjście filtru dopasowanego dla sygnału z szumem o amplitudzie 1 przy ilości próbek 10000

Jak widać, tym razem jakość detekcji się poprawiła w porównaniu z analogicznym przypadkiem o mniejszej ilości próbek. Wynika to z tego, że piki sygnału wyjściowego zależą od energii sygnału zpróbkowanego, a jego energia zależy od ilości próbek. Stąd więc amplituda piku w tym przypadku jest większa, natomiast zaszumione tło pozostało mniej więcej takie same. Dlatego też w tym przypadku łatwiej jest rozróżnić pik sygnału wyjściowego od szumu.

6 Zadanie 7

W tym zadaniu napisano skrypt wyliczający SNR na wejściu i wyjściu filtru. SNR wejściowy to stosunek średniej mocy sygnału do średniej mocy szumów, a więc w tym przypadku to średnia z kwadratów poszczególnych próbek sygnału `x2` do średniej z kwadratów poszczególnych próbek `noise`. Natomiast SNR wyjściowy liczy się trochę inaczej, gdyż sam filtr dopasowany zwraca informacje o energii sygnału, a więc nie należy już podnosić amplitudy do kwadratu. Tutaj SNR jest stosunkiem amplitudy pików wyjściowych do średniej wartości próbek wyjściowych po usunięciu pików.

Listing 6: Skrypt MATLAB do zadania 7.

```

1 clear all
2
3 % Parametry sygnału
4 N = 1000;
5 f0_norm = 0;
6 f1_norm = 0.1;
7 repetitions = 5;
8 noise_amplitude = 8;
9
10 t = (0:repetitions*N-1) / N;
11
12 x1 = chirp(1:N, f0_norm, N, f1_norm, 'linear', 'complex');
13 x2 = repmat(x1, 1, repetitions);

```

```

14 noise = noise_amplitude*complex(randn(size(x2)), randn(size(x2)));
15 x2_with_noise = x2 + noise;
16
17 signal_power_in = mean(abs(x2).^2);
18 noise_power_in = mean(abs(noise).^2);
19 SNR_in = 10 * log10(signal_power_in / noise_power_in);
20
21 fprintf("noise_amplitude = %.2f \n", noise_amplitude);
22 fprintf("SNR_in = %f \n", SNR_in);
23
24 % Filtr dopasowany
25 B = conj(x1(end:-1:1));
26 y = filter(B, 1, x2_with_noise);
27
28 signal_peak_out = max(abs(y));
29 noise_out = abs(y);
30 noise_out(abs(y) > 150) = 0; % usuwanie próbek o wartości większej niż 150
31 SNR_out = 10 * log10(signal_peak_out / mean(noise_out));
32
33 fprintf("SNR_out = %f \n\n", SNR_out);

```

Wykonywano program dla sygnału o ilości próbek 1000 i amplitudzie 1 dla szumu o różnych wartościach amplitudy. Wyniki są następujące:

Listing 7: Wynik działania skryptu z zadania 7.

```

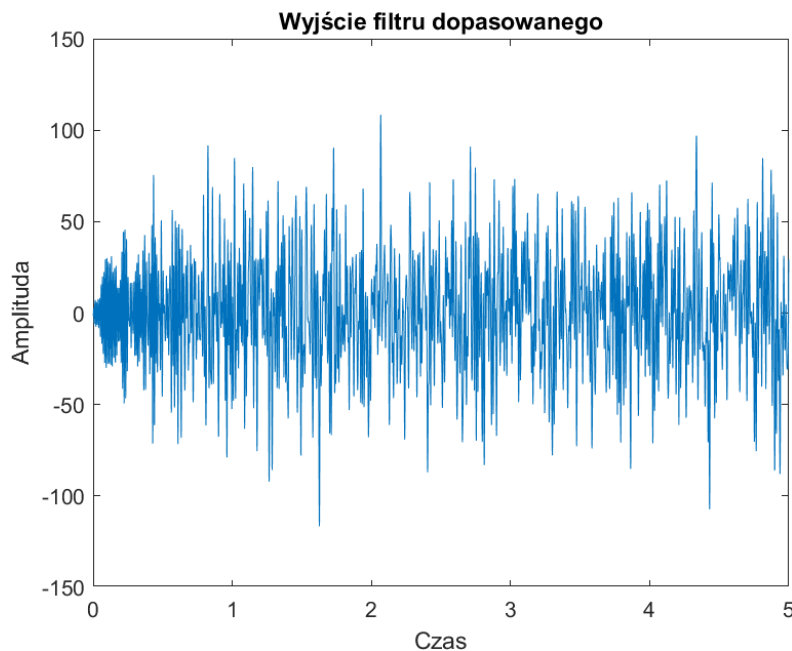
1 noise_amplitude = 0.10
2 SNR_in = 17.041049
3 SNR_out = 18.134336
4
5 noise_amplitude = 1.00
6 SNR_in = -2.906707
7 SNR_out = 14.083459
8
9 noise_amplitude = 5.00
10 SNR_in = -17.062758
11 SNR_out = 15.337730
12
13 noise_amplitude = 8.00
14 SNR_in = -21.122108
15 SNR_out = 18.676399

```

Jak widać, dla małych wartości amplitudy szumu SNR wejściowy jest prawie równy wyjściowemu, natomiast wraz ze zwiększaniem amplitudy szumu drastycznie maleje SNR_in, podczas gdy SNR_out jest prawie stały. Wynika to z tego, że w przypadku sygnału wejściowego SNR to po prostu stosunek mocy sygnału do szumu, a w późniejszych przypadkach amplituda szumu jest 8 razy większa niż sygnału, stąd ujemny SNR. Natomiast SNR wyjściowy jest prawie stały, co pokazuje, jak dobrze filtr dopasowany sprawuje się w roli detektora sygnału. Moc sygnału wyjściowego odpowiadająca wartościom pików jest prawie stała. Natomiast tło szumów wyjściowych zależy w małym stopniu od amplitudy szumów wejściowych.

7 Zadanie 8

Tym razem na wejściu filtru dopasowanego podano jedynie szum zespolony, co skutkowało otrzymaniem na wyjściu poniższego sygnału:



Rysunek 18: Wyjście filtru dopasowanego dla czystego szumu o amplitudzie 1

Jak widać, filtr nie wykrywa już żadnego sygnału, a na jego wyjściu występuje szum podobny do zaszumionego tła z poprzednich przykładów, gdy filtr działał poprawnie. Wynik eksperymentu jest oczywisty, filtr nie może nic wykrywać, gdy nie ma na wejściu odpowiedniego sygnału.

8 Zadanie 9

Co i w jaki sposób wpływa na to czy dany sygnał może być wykryty przez filtr dopasowany. Jakie parametry sygnału wejściowego mają znaczenie, a jakie nie?

Na wykrywalność sygnału wpływ ma kilka czynników. Znaczenie ma między innymi wartość SNR na wejściu filtru. Co prawda filtr świetnie sobie radzi z odnajdywaniem sekwencji poszukiwanego sygnału, nawet gdy wartość szumu jest o wiele większa niż sygnału, jednak ma to swoje granice. Drugim czynnikiem jest ilość próbek poszukiwanej sekwencji. Im jest ich więcej, tym większą energię ma ten fragment sygnału, a że filtr dopasowywany zwraca wartość odpowiadającą energii, to tym łatwiej jest wtedy w sygnale wyjściowym rozróżnić piki wyjściowe od szumu. Znaczenie ma też to, czy sygnał wejściowy jest zespolony, czy nie, ponieważ filtr został zaprojektowany do sygnału zespolonego, a więc może gorzej wykrywać sekwencje sygnału, jeżeli zawiera on tylko składową rzeczywistą.