

Imię Nazwisko  
Nr .....

Gdańsk 15.01.2020

Specjalność:

## SYMULACJA KOMPUTEROWA SYSTEMÓW PROJEKT

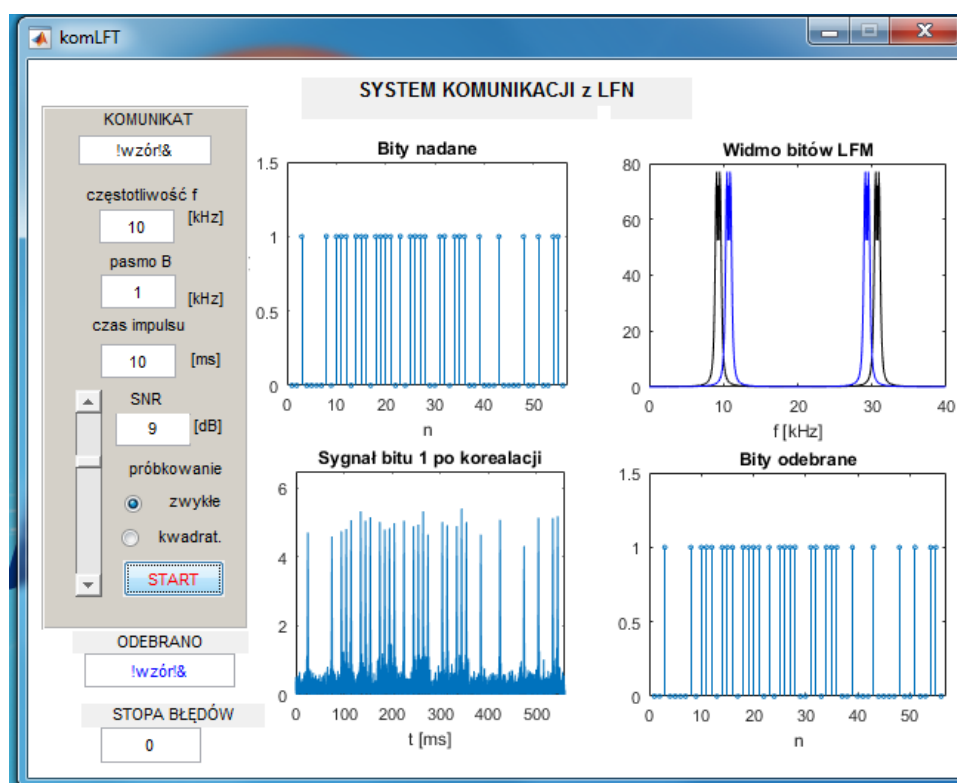
Temat nr 30: System komunikacyjny wykorzystujący liniową  
modulację częstotliwości LFM.

### Opis systemu

System przeznaczony jest do przesyłania informacji tekstowej o dowolnej długości, składającej się ze znaków znajdujących się w zbiorze ASCII. Charakterystyczną cechą systemu jest liniowa modulacja częstotliwości LFM wąskopasmowych sygnałów bitów. Aby wykorzystać zalety sygnałów LFM, w systemie dokonywany jest ich odbiór korelacyjny. Zastosowany sposób modulacji jest przykładem akademickim o wątpliwym zastosowaniu praktycznym.

W odbiorniku systemu zastosowano próbkowanie zwykłe i kwadraturowe.

Wprowadzanie informacji i nastaw oraz prezentacja wyników odbywa się w graficznym interfejsie użytkownika GUI. Interfejs został utworzony w programie „guide” i jest przedstawiony na rys. 1.



Rys. 1. Interfejs użytkownika

W okienkach znajdujących się w ciemnym polu wpisuje się przesyłany komunikat (tekst), częstotliwość, szerokość pasma LFM zajmowanego przez bit i czas trwania impulsu bitu. Stosunek sygnału do szumu w paśmie odbiornika ustawia się suwakiem, a wynik widoczny jest w okienku SNR. Rodzaj próbkowania wybiera się przyciskami, przy czym przy uruchomieniu systemu należy koniecznie wcisnąć pusty przycisk, a potem ewentualnie wcisnąć przycisk właściwego wyboru.

Uruchomienie programu następuje po wciśnięciu START. Wszystkie operacje wykonywane są w programach oblLFMP.m i korrelLFM.m. Po wykonaniu wstępnych obliczeń, których efekty pokazują się na górnych rysunkach program zawiesza się i oczekuje na wprowadzenie liczby transmisji użytych do wyznaczania stopy błędów. Jeżeli stopa błędów nie jest wyznaczana należy wpisać  $K=1$ . Program wykonuje wszystkie pozostałe operacje, pokazuje wyniki na dolnych rysunkach i podaje odebrany tekst. W okienku STOPA BŁĘDÓW podaje liczbę błędów w tej transmisji. Jednocześnie wykonuje kopię rysunków do umieszczenia w sprawozdaniu. Są one zamieszczone niżej.

Przy wpisaniu  $K>1$  program powtarza operacje jak dla  $K=1$  wykonując po cztery rysunki w każdej transmisji i wyznacza stopę błędów (suma liczby błędów w każdej transmisji podzielona przez  $K$ ). W celu uniknięcia dużej liczby zbędnych rysunków należy w programie oblLFM „zakomentować” instrukcje wykonujące rysunki.

#### Opis programu przy próbkowaniu zwykłym

1. Wpisujemy tekst (tu !wzór!&).
2. Program podaje numery znaków w kodzie ASCII.  
33 119 122 243 114 38
3. Znaki są zapisywane w kodzie binarnym

00100001

01110111

01111010

11110011

01110010

00100110

4. Tworzona jest macierz liczb binarnych

0 0 1 0 0 0 0 1

0 1 1 1 0 1 1 1

0 1 1 1 1 0 1 0

1 1 1 1 0 0 1 1

0 1 1 1 0 0 1 0

0 0 1 0 0 1 1 0

5. Tworzony jest wektor nadawanych liczb binarnych, pokazany na rys. 2.

Bity reprezentowane są przez wąskopasmowe sygnały LFM, których widma zamieszczono na rys.3.

Analogowy zapis sygnałów ma postać:

$$s(t) = \sin[2\pi(f - B/2 + Bt/2T)t],$$

gdzie  $f$  jest częstotliwością środkową,  $B$  – szerokością pasma, a  $T$  czasem trwania impulsu bitu.

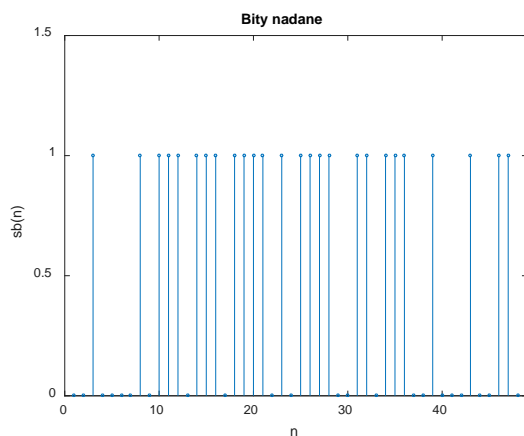
Cyfrową postać generowanych sygnałów LFM przedstawiają następujące instrukcje:

```
f1=f-0.7*B;    %Częstotliwość środkowa bitu 1
f0=f+0.7*B;    %Częstotliwość środkowa bitu 0
Bd=f-1.2*B;    %Częstotliwość dolna
Bg=f+1.2*B;    %Częstotliwość dolna
Bc=2.4*B;      %Szerokość pasma odbiornika
```

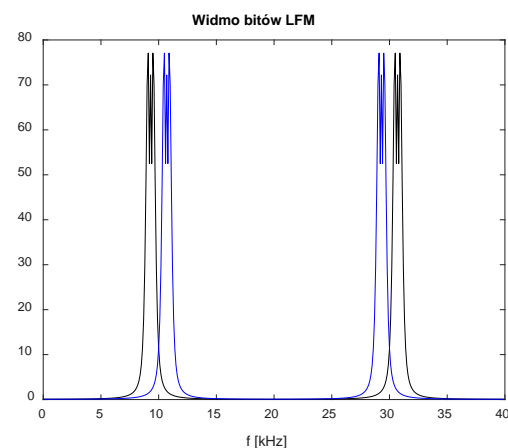
```
a1=2*pi*f1/fs;
a2=2*pi*f0/fs;
b=2*pi*B/(2*fs);
n = 0:Ni-1;
s1=sin((a1-b+b*n/Ni).*n);    %Sygnał bitu 1
s0=sin((a2-b+b*n/Ni).*n);    %Sygnał bitu 0
```

Częstotliwość próbkowania wynosi  $fs=4f$ .

6. Dokonywana jest modulacja polegająca na przypisywaniu bitom dodatnim sygnału  $s1$ , a bitom zerowym – sygnału  $s0$ . Ciąg sygnałów  $s1$  i  $s0$  jest sygnałem nadanym.



Rys.2. Tekst zakodowany w bitach.



Rys.3. Widma impulsów LFM.

7. Na wejściu odbiornika do sygnału nadawanego dodawany jest szum gaussowski *noise* o odchyleniu standardowym  $\sigma$  obliczanym jak następuje:

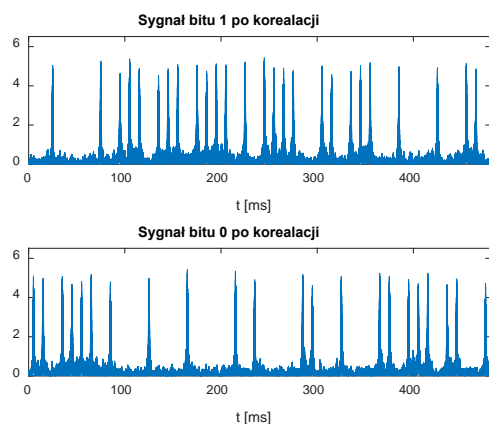
```
%Szum
vr=0.5*10^(-snr/10);
sigma=sqrt(0.5*vr*fs/Bc)
noise=sigma*randn(1,N);
```

8. Sumaryczny sygnał podawany jest do wejść dwóch korelatorów (program `korrelLFM.m`), w których jest obliczana korelacja sygnału odebranego z sygnałami wzorcowymi  $s1$  i  $s0$ . Sygnały na wyjściach korelatorów mają przebiegi pokazane na rys. 4.
9. Sygnały z wyjść korelatorów poddawane są detekcji progowej, a wyniki są sumowane w przedziałach czasu trwania bitu  $T$ .
10. Wyniki sumowania poddawane są drugiej detekcji, której wynik pokazano na rys. 5.

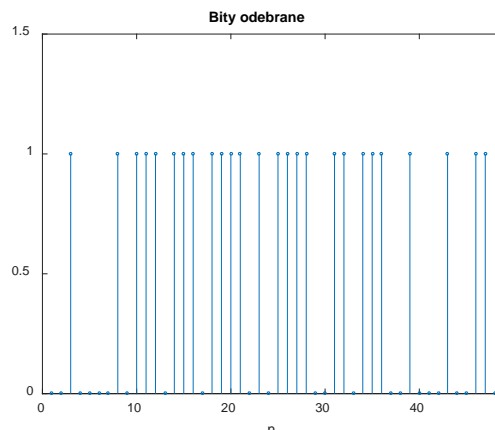
11. Sprawdzana jest zgodność bitów 1 i 0 z obu wyjść korelatorów. Błąd jest sygnalizowany w Command Window jako BŁĄD.

12. Wyznaczony sygnał jest dekodowany i wpisywany w oknie ODEBRANO.

Uwaga: stopę błędów można spisać z ekranu dla różnych wartości SNR i zamieścić w formie wykresu.



Rys.4. Sygnały po korelacji.



Rys. 5. Odebrane bity.

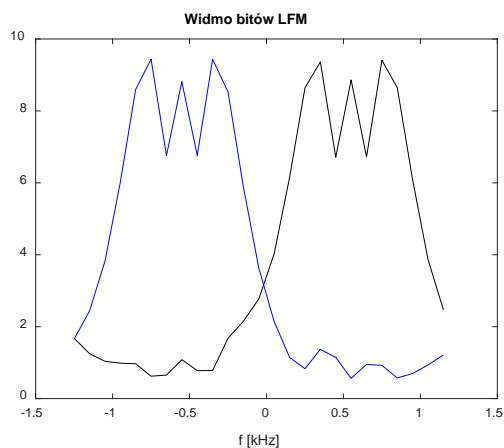
W pokazanym wyżej przykładzie transmisja była bezbłędna, co widać porównując rys. 2 z rys. 5.

### Opis programu przy próbkowaniu kwadraturowym

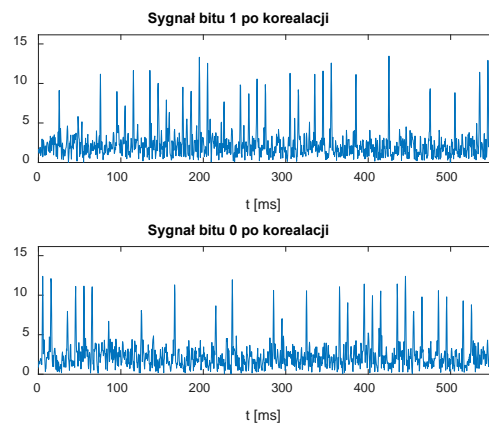
Przy próbkowaniu kwadraturowym w opisanym wyżej programie następują dwie istotne zmiany:

1. Sygnał odebrany jest próbkowany kwadraturowo według następujących instrukcji:  
( $d$  jest „okresem” próbkowania kwadraturowego)  
 $N_k = \text{round}(N/d)$  ;    %Liczba próbek kwadraturowych  
for  $n=1:N_k$   
   $s1k(n) = \text{sn}((n-1)*d+1)$ ; %Składowa kwadraturowa  
   $c1k(n) = \text{sn}((n-1)*d+2)$ ; %Składowa synfazowa  
end  
   $sk = c1k + 1i*s1k$ ;    %Sygnał zespolony
2. Sygnał zespolony jest korelowany z dwoma wzorcami sygnałów LFM bitów, które są wyznaczone z próbkowania kwadraturowego wzorców sygnałów próbkowanych w zwykły sposób. Widma takich sygnałów pokazano na rys. 6.

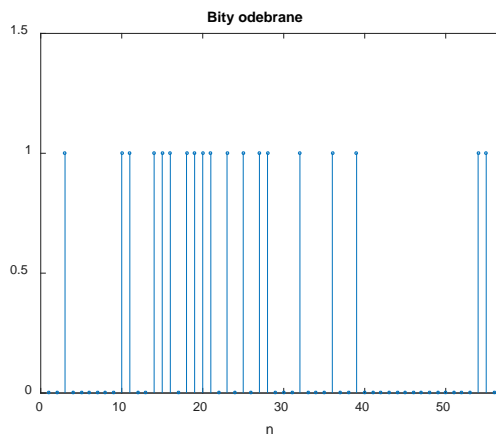
Dalsze operacje wykonywane są tak samo jak w programie z próbkowaniem zwykłym, lecz zmieniają się długości wektorów. Na rys. 7 pokazano sygnały po korelacji, a na rys. 8 – odebrane bity. Program z próbkowaniem kwadraturowym wykonany był przy tych samych ustawieniach co program z próbkowaniem zwykłym. Widoczne jest pogorszenie jakości transmisji – wystąpiło 6 błędów, a tylko jeden znak był poprawny. Nieznaczne zwiększenie SNR skutecznie eliminuje wszystkie błędy. Jakość transmisji można poprawić także zmieniając parametry próbkowania kwadraturowego, np. zwiększając czas trwania  $T$  impulsu bitu. Przykładowo, dwukrotne wydłużenie długości impulsu daje bezbłędną transmisję.



Rys. 6. Widmo sygnałów LFM po próbkowaniu kwadraturowym.



Rys. 7 Sygnały odebrane na wyjściach korelatorów.



Rys. 8 Odebrany ciąg bitów po próbkowaniu kwadraturowym.

### Uwagi i podsumowanie

Przeprowadzone testy programów wykonanych w ramach projektu wykazały jego poprawne funkcjonowanie dla różnych zbiorów parametrów, pod warunkiem, że ich dobór miał sens techniczny.

Program nie sygnalizuje wprowadzania złych nastaw, co wg autora jest jego jedynym ograniczeniem.