**Podstawy przetwarzania sygnałów – laboratorium 2**

Wszystkie sygnały w Matlabie są zdyskretyzowane, zatem przechowujemy wektor próbek sygnału. Należy też pamiętać, że wartości przyjmowane przez daną próbkę są skwantowane (jest ograniczona precyzja/dokładność). Zatem generując sygnał, generujemy zbiór próbek dla odpowiadających im chwil. Podobnie wygląda sytuacja w przypadku analizy sygnałów ciągłych – należy najpierw je spróbkować, czyli pobrać/odczytać i zapisać wartości sygnału w danych chwilach. Należy pamiętać o tym, że przetworniki A/C charakteryzują się rozdzielczością. Zatem możemy zapisać wartość sygnału z pewną ograniczoną dokładnością.

**Generowanie podstawowych sygnałów.**

**Sygnał sinusoidalny:**

1. Określamy, ile próbek (zmienna N) będziemy potrzebować lub ile sekund sygnału potrzebujemy.
2. Określamy z jaką częstotliwością próbkujemy sygnał (nazwijmy tę zmienną fp), czyli mamy informację, ile próbek sygnału podczas 1s trwania sygnału zgromadzimy.
3. Przykładowo: mając N=2000 dla fp=200 Hz mamy 10s sygnału, dla N=1000 i fp=200 Hz mamy 5 s sygnału
4. Generujemy wektor czasu t=0:1/fp:(N-1)/fp;  
   czyli od chwili 0 co 1/fp (np. 1/200) do (N-1)/fp (np. (1000-1)/200) – w ten sposób dostaniemy N wartości wektora czasu (od 0 do N-1)
5. Generujemy sygnał sinusoidalny o amplitudzie A1, częstotliwości f1 i kącie przesunięcia fazowego fi1 (w radianach): x=A1\*sin(2\*pi\*f1\*t+fi1)
6. Sygnał sinusoidalny ze składową stałą k0: x=k0+A1\*sin(2\*pi\*f1\*t+fi1)

---------------------------------------------------------------------------------------

Skrypt 1:

clear all

fp=100;

f1=10;  
fi1=0;

A1=2;

N = 200; %liczba probek sygna?u

t=0:1/fp:(N-1)/fp;

x = A1\*sin(2\*pi\*f1\*t+fi1);  
plot (t, x);

xlabel ('czas [s]');

ylabel ('sygnal');

title ('wykres sinusoidy');

-----------------------------------------------------

Zadania:

1. W powyższym skrypcie zamiast plot(t,x) użyj stem(t,x) – co uzyskałeś?
2. Spróbuj dodać składową stałą o k0=2. Co się zmieniło w sygnale?
3. Spróbuj wygenerować sygnał będący sumą dwóch składowych sinusoidalnych o częstotliwościach f1=10Hz, f2=20Hz, amplitudach A1=2, A2=6 oraz kątach przesunięcia fazowego równych 0. Przyjmij częstotliwość próbkowania fp=200Hz; N=1000;
4. Dodaj do sygnału z poprzedniego zadania trzecią składową f3=15Hz, A3=4. Co zauważyłeś charakterystycznego w przebiegu sygnału?
5. Spróbuj tak zmodyfikować skrypt, aby trzecia składowa trwała od 1 do 3 sekundy trwania sygnału.

**Sygnały losowe.**

1. Generowanie szumu o rozkładzie równomiernym: funkcja rand(1,N) – generowanie N próbek sygnału, wartości próbek należą do przedziału (0,1).
2. Generowanie szumu z rozkładu normalnego: funkcja randn(1,N)

Zadania:

1. Wygeneruj sygnał losowy z rozkładu równomiernego, N=1000 próbek, dodaj ten sygnał do sygnału ze skryptu 1. Wykreśl w 1 okienku (Figure) po kolei: sygnał, szum, sygnał+szum, skorzystaj z polecenia subplot.
2. Wygeneruj sygnał losowy z rozkładu równomiernego, tak aby wartości sygnału należały do przedziału [0.01; 0.05], dodaj ten sygnał do sygnału ze skryptu 1. Wykreśl w 1 okienku (Figure) po kolei: sygnał, szum, sygnał+szum, skorzystaj z polecenia subplot.
3. Wygeneruj sygnał losowy z rozkładu normalnego, N=1000 próbek, dodaj ten sygnał do sygnału ze skryptu 1. Wykreśl w 1 okienku (Figure) po kolei: sygnał, szum, sygnał+szum, skorzystaj z polecenia subplot.
4. Wygeneruj sygnał losowy z rozkładu normalnego, średnia sygnału 0.01, wariancja 0.5, dodaj ten sygnał do sygnału ze skryptu 2. Wykreśl w 1 okienku (Figure) po kolei: sygnał, szum, sygnał+szum, skorzystaj z polecenia subplot.

**Sygnał o narastającej/malejącej częstotliwości – chirp, sweep signal**

1. Generujemy wektor czasu jak dla sinusa.
2. Generujemy sygnał wywołując funkcję chirp(t, f0, t1, f1) z odpowiednimi argumentami, t – wektor czasu, f0 – częstotliwość początkowa, t1 – chwila, w której osiągana jest częstotliwość f1

Zadania:

1. Wygeneruj trwający 5s sygnał chirp, fp=20Hz, częstotliwość f0=1Hz, f1=5Hz, t1=5s;
2. Wygeneruj trwający 5s sygnał chirp, fp=20Hz, częstotliwość f0=1Hz, f1=5Hz, t1=2s;
3. Wygeneruj sygnał jak z zad.1, ale częstotliwość ma narastać logarytmicznie.

**Sygnał audio – audioread(), audiowrite(), sound()**

1. Wczytaj do wektora x sygnał audio
2. Narysuj przebieg tego sygnału w czasie
3. Odsłuchaj ten sygnał
4. Wytnij z sygnału próbki odpowiadające drugiej sekundzie tego sygnału i zapisz je do wektora x2, tak, aby dało się je odsłuchać.