

## 1. Wstęp:

Celem ćwiczenia jest zapoznanie się studentów z podstawowymi sygnałami i ich właściwościami. Większość z nich używane jest w przetwarzaniu sygnałów, kompresji danych, ich filtracji oraz składowaniu. Do wykonania ćwiczenia niezbędny jest program Matlab oraz podstawowe umiejętności z jego korzystania.

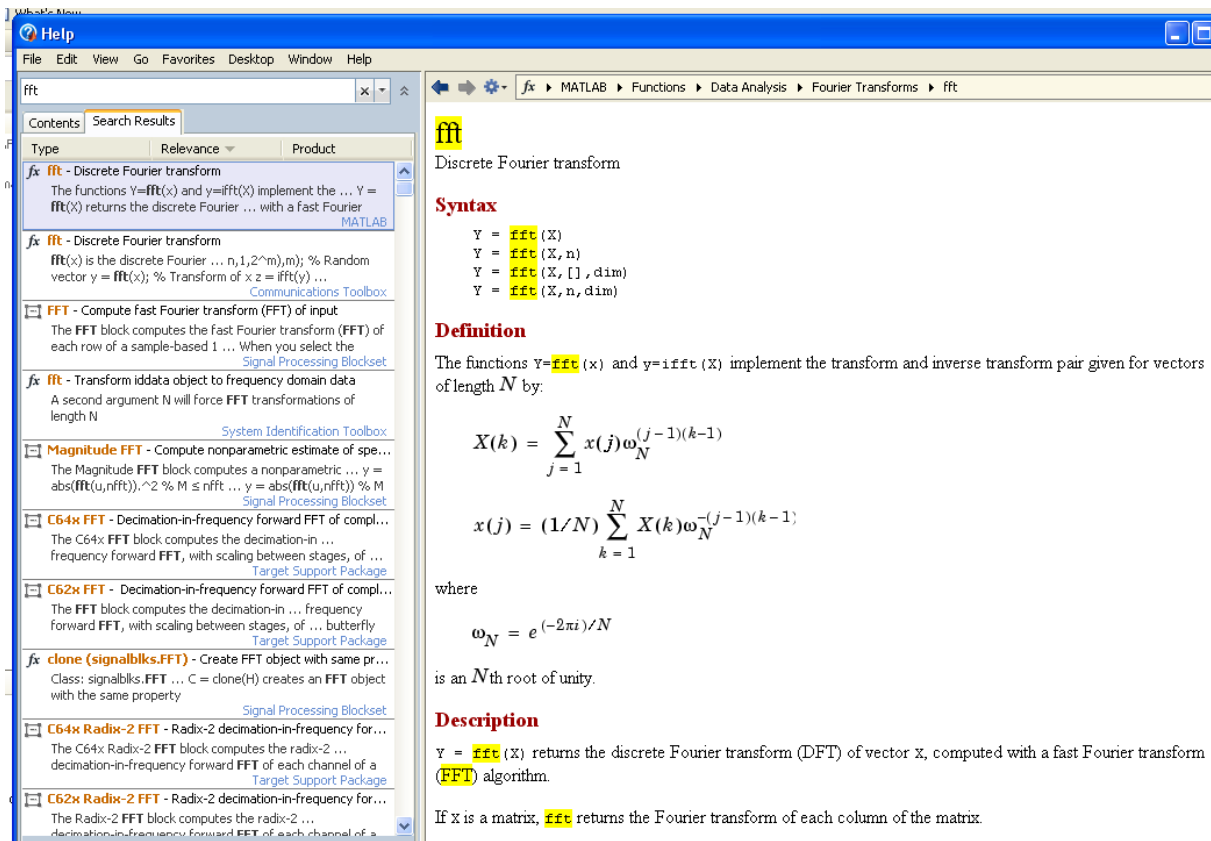
## 2. Szczegóły techniczne:

Środowisko Matlab jest bardzo wygodnym narzędziem do wykonywania skomplikowanych obliczeń. Wbudowano w niego wiele gotowych funkcji matematycznych, których zastosowanie znacząco skraca czas rozwiązania zadania. Student posiadający podstawowe umiejętności z zakresu programowania w języku, C, C++ oraz Java bardzo szybko zgłębi specyfikę środowiska Matlab.

Podobnie jak w przypadku programowania jedną z najważniejszych umiejętności jest skutecznie korzystanie z dokumentacji technicznej. W środowisku Matlab jest ona dostępna pod ikoną znaku zapytania (rys. 1.). Po jej otwarciu pojawia się wyszukiwarka funkcji i zagadnień (rys. 2.). Korzystając z niej można bardzo łatwo znaleźć szukaną funkcję, jej wywołania, podobne hasła oraz opis teoretyczny zagadnienia. W razie braków w znajomości środowiska roboczego można skorzystać z jednego z wielu dostępnych w sieci samouczków programu Matlab. Na przykład: <http://www.tutorialspoint.com/matlab/>



Rys. 1. Ikona pomocy technicznej w oknie programu Matlab zaznaczona na czerwono.



Rys. 2. Okno wyszukiwarka pomocy pakietu Matlab.

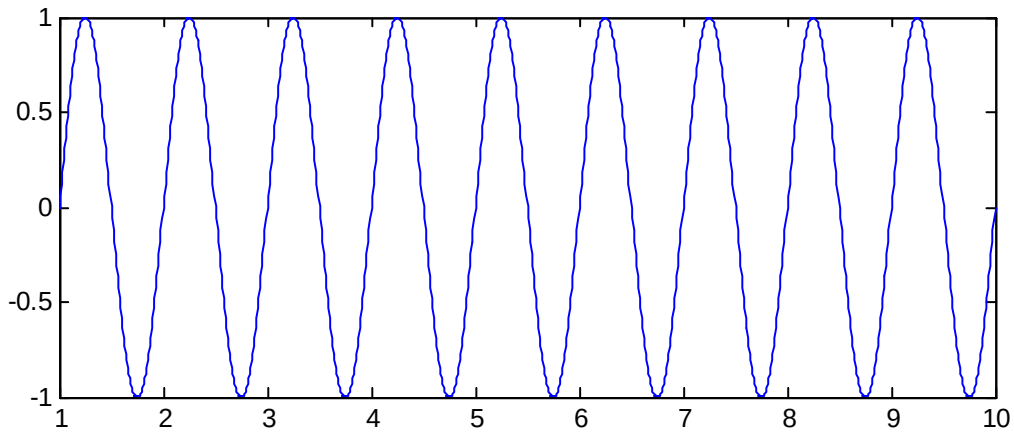
### 3. Zadania:

3.1. Korzystając z dostępnych funkcji wygeneruj w dziedzinie czasu oraz wyświetl na wykresach następujące sygnały:

- funkcję sinus o częstotliwości 100 Hz
- deltę Diraca w chwili czasu 3 s
- uskok jednostkowy w chwili czasu 3s
- sygnał prostokątny rozpoczynający się w chwili czasu 3s
- funkcję cosinus o częstotliwości 100 Hz
- dowolny, kilkusekundowy fragment nagrania dźwiękowego (załaduj go do workspace poleceniem wavread - plik \*.wav)

Podczas pracy pamiętaj o tym, że wynikiem zadania będą sygnały dyskretne. Musisz zadbać o spełnienie podstawowych praw takich jak twierdzenie o próbkowaniu, zapewnienie odpowiedniej rozdzielczości czasowej sygnałów oraz odpowiedni dobór liczby próbek sygnału. Nie bój się eksperymentować aby znaleźć optymalne parametry funkcji. Zapisz wyniki swoich działań oraz nasuwające się wnioski.

```
>> x=1:0.01:10;
>> y= sin(2.*pi.*1.*x);
>> plot(x,y)
>>
```



Przykładowy kod generujący oraz wyświetlający funkcję sinus o częstotliwości 1Hz, dla przedziału czasu od 1 s do 10 s, z próbkami co 0.01 s.

*Efekt działania wyżej pokazanego kodu*

3.2. Kolejnym zadaniem będzie wygenerowanie wyżej wymienionych funkcji w na płaszczyźnie 2D. Rozpocznij od najprostszych:

- a) deltę Diraca o współrzędnych (10, 100)
- b) uskok jednostkowy o współrzędnych (10, y)
- c) sygnał prostokątny o współrzędnych (10, y) i wymiarach 50x50 punktów obrazu
- d) funkcję sinus o częstotliwości (1/50, 1/200)
- e) funkcję cosinus o częstotliwości (10, 10)
- f) dowolne zdjęcie/obraz (załaduj go do workspace poleceniem `imread` oraz przekonwertuj z przestrzeni kolorów RGB do skali szarości poleceniem `rgb2gray`)

„Przejście” z dziedziny czasu do dziedziny płaszczyzny obrazu wymaga trochę wyobraźni oraz chwili zastanowienia, nie zniechęcaj się więc na samym początku. Nie bój się eksperymentować aby znaleźć optymalne parametry funkcji. Zapisz wyniki swoich działań oraz nasuwające się wnioski.

```

clear;
x=1:900;
y=1:900;

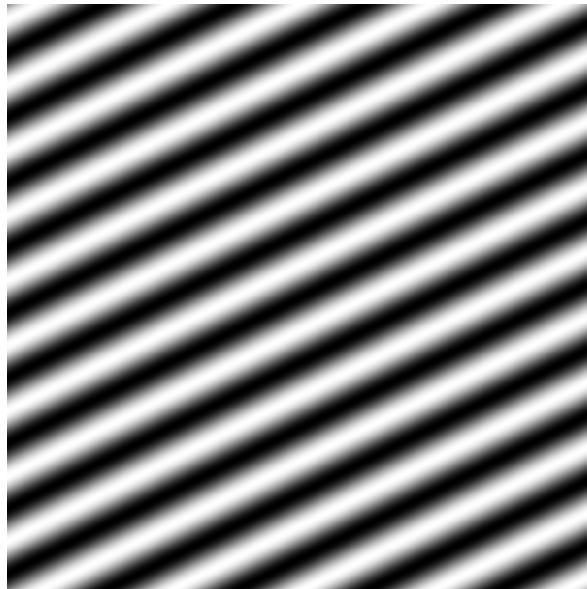
n = length(x);
m = length(y);

for ii = 1:n
    for jj = 1:m
        f(ii,jj) = sin(2.*pi.*x(ii)./100 + 2.*pi.*y(jj)./200);
    end;
end;

figure;
imshow(f, [min(f(:)) max(f(:))]);

```

Przykładowy skrypt (m-plik) generujący oraz wyświetlający dwuwymiarową funkcję sinus o częstotliwościach (1/100, 1/200), dla płaszczyzny 900x900 punktów



*Efekt działania przykładowego kodu*

### 3.3 Sygnały 1D w dziedzinie częstotliwości:

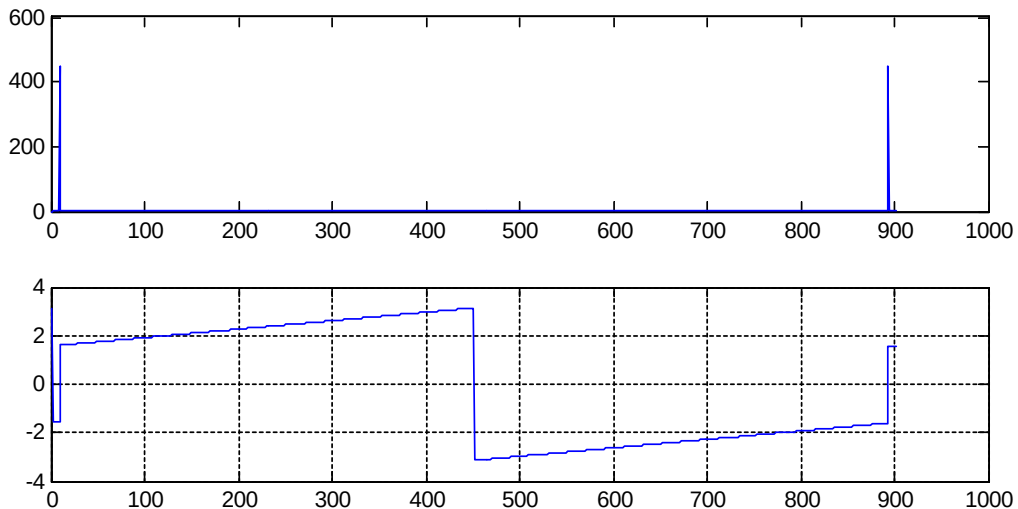
Korzystając z funkcji `fft` oblicz widmo transformaty FFT sygnałów z ćwiczenia 3.1. Otrzymane widmo podziel na widmo fazy oraz amplitudy oraz przedstaw je na oddzielnych wykresach. Wyszukaj widma odpowiadających sygnałów ciągłych i porównaj je z otrzymanymi. Jakie widzisz różnice i z czego one wynikają. Skomentuj otrzymane wyniki.

```

>> x=1:0.01:10;
>> y= sin(2.*pi.*1.*x);
>> widmo = fft(y);
>> amplituda = abs(widmo);
>> faza = angle(widmo);
>> subplot(2,1,1);
>> plot(amplituda)
>> subplot(2,1,2);
>> plot(faza)
>> grid on;
>> |

```

*Kod obliczający widmo FFT funkcji sinus oraz wyświetlający oddzielnie amplitudę oraz fazę widma*



*Efekt działania wyżej przytoczonego kodu*

### 3.4 Sygnały 2D w dziedzinie częstotliwości:

Korzystając z funkcji `fft2` oblicz widmo transformaty FFT sygnałów z ćwiczenia 3.2. Otrzymane widmo podziel na widmo fazy oraz amplitudy oraz przedstaw je na oddzielnych wykresach. Porównaj z wynikami otrzymanymi w punkcie 3.3. Jakie widzisz różnice i podobieństwa oraz z czego one wynikają. Czy jesteś już w stanie zrozumieć na czym polega rozszerzenie dziedziny sygnału do płaszczyzny?

```

obraz = imread('mathworks.jpg');
obraz = rgb2gray(obraz);

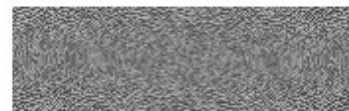
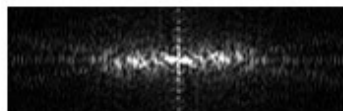
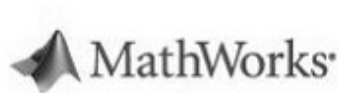
figure;
imshow(obraz, [0 255]);

widmo = fft2(obraz);
widmo = fftshift(widmo);
amplituda = abs(widmo);
faza = angle(widmo);

figure;
imshow(amplituda, [min(amplituda(:)) max(amplituda(:))./100]);
figure;
imshow(faza, [-7 7]);

```

*Przykładowy kod obliczający oraz wyświetlający widmo FFT loga MathWorks*



*Efekt działania kodu*

#### 4. Opracowanie wyników:

4.1 Otrzymane wykresy przedstaw zgodnie z poleceniami w punktach 3.1 do 3.4. Wraz z każdym przypadkiem zamieść odpowiednie fragmenty kodu.

4.2. Podaj przykłady sygnałów 1D oraz 2D stosowanych w świecie multimedialnym.

*Funkcje środowiska Matlab, które mogą być dla Ciebie przydatne w wykonywaniu ćwiczenia:*

*fft, fft2, ifft, ifft2, conv, conv2, deconv, plot, hold on; grid on; image, imshow, figure, sin, cos, abs, angle, wavread, imread, rgb2gray i inne.*