

## Wprowadzenie do środowiska Matlab/GNU Octave

### 1 Cel ćwiczenia

1. Wprowadzenie do środowiska Matlab/GNU Octave.
2. Wczytanie i wyświetlenie sygnału cyfrowego.

### 2 Ważne informacje

Matlab

- <http://www.math.utah.edu/lab/ms/matlab/matlab.html>
- <http://www.math.mtu.edu/~msgocken/intro/intro.html>

GNU Octave

- <http://www.gnu.org/software/octave/doc/interpreter/>
- <http://www.youtube.com/watch?v=TWqYSOiSTEa&list=PLZ-E1VZLTeHMtUopmGx99KS775aHYP3e4&index=1>

### 3 Zadania

1. Matlab jako kalkulator: Używając **Command Window** wykonaj następujące obliczenia

- |                                                                     |                     |
|---------------------------------------------------------------------|---------------------|
| (a) $\exp\left(\frac{(x-2y)^2}{4}\right)$ ; dla $x = 2$ i $y = 3.5$ | (c) $2 \neq 5$      |
| (b) $\cos(2x) + \pi$ ; dla $x = \frac{\pi}{3}$                      | (d) $\sqrt[3]{156}$ |

Użyj funkcji *whos* aby uzyskać nazwy użytych zmiennych oraz informacje o nich.

2. Wygeneruj wektory:

Uwaga: Od tej chwili kolejne zadania zapisz w osobnych skryptach (m-plikach) *File* → *New* lub (*Ctrl* + *N*)

- |                                                                               |                                                                                                                        |
|-------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| (a) $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & \dots & 99 & 100 \end{bmatrix}$ | (c) $\mathbf{C} = \begin{bmatrix} 5 & 4.9 & 4.8 & \dots & 1.1 & 1 \end{bmatrix}$                                       |
| (b) $\mathbf{B} = \begin{bmatrix} -1 \\ 1/2 \\ 3 \\ 9 \end{bmatrix}$          | (d) $\mathbf{D} =$ [Do wektora $\mathbf{B}$ dodaj element 5-ty o wartości 1 oraz zmień wartość elementu 2-giego na 10] |

3. Wygeneruj macierz  $\mathbf{P} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 & 7 \\ 5 & 6 & 7 & 8 & 9 \end{bmatrix}$ . Która z poniższych odpowiedzi da macierz  $\begin{bmatrix} 1 & 3 & 4 \\ 3 & 5 & 6 \\ 5 & 7 & 8 \end{bmatrix}$ .

- |                            |                                |
|----------------------------|--------------------------------|
| (a) $\mathbf{P}(:, :)$     | (d) $\mathbf{P}(:, [1\ 3:4])$  |
| (b) $\mathbf{P}(:, 1:3:4)$ | (e) $\mathbf{P}(:, 1:2:5)$     |
| (c) $\mathbf{P}(1:3:4, :)$ | (f) $\mathbf{P}(:, [1\ 3\ 4])$ |

4. Wygeneruj macierze  $A=[1 \ 2; 3 \ 4]$  oraz  $B=[1 \ 2'; 3 \ 4']$ . Porównaj ich rozmiar za pomocą funkcji `size()` - z czego wynika różnica rozmiarów?

Użyj funkcji `str2num()` do zamiany zmiennych znakowych na liczbowe i sprawdź ponownie rozmiar macierzy  $B$  po tej zmianie.

5. Jeśli  $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ ,  $B = \text{eye}(3)$ ,  $C = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$ ,  $D = \text{zeros}(3,3)$ ,  $e = [1 \ 2 \ 4]$ ,  $F = \text{ones}(2) * 5$ ,

które z poniższych komend nie spowodują wystąpienia błędu?

- |              |              |                               |
|--------------|--------------|-------------------------------|
| (a) $A - C$  | (e) $B * D$  | (i) $(A * F) - B(1:2,1:2)$    |
| (b) $e * C$  | (f) $B. * D$ | (j) $F - e(1,2)$              |
| (c) $A * C$  | (g) $F * A$  | (k) $B(1,1)/D(2,3)$           |
| (d) $A * C'$ | (h) $F. * A$ | (l) $B(1,1) \setminus D(2,3)$ |

6. Wygeneruj wektor  $V = [-1 \ -4 \ 8 \ 10 \ 2 \ 15]$ .

Zmień wartości wektora  $V$  mniejsze od 5 na wartość 100. Napisz kod dwoma sposobami: (1) używając odpowiednich pętli (klasycznie), a następnie (2) wektorowo tzn.  $V(V < 5) = 100$ . Porównaj czas jaki zajęło wykonanie obu kodów (służą do tego funkcje `tic` i `toc`) - który kod działa szybciej?

7. Wygeneruj 10-sekundowy odcinek funkcji `sinc` spróbkowany z częstotliwością  $F_p = 1000$  Hz, a następnie samodzielnie oblicz i narysuj:

- (a) wartość średnią,  
(b) funkcję autokorelacji unormowaną przez długość sygnału  $N$ :

$$R_x[k] = \frac{1}{N} \sum_{n=0}^{N-1-k} x[n]x[n+k] \quad \text{dla } 0 \leq k \leq N-1. \quad (1)$$

- (c) Otrzymane wyniki porównaj stosując funkcję `mean` i `xcorr(, 'biased')`

```
Fp=1000;%czestotliwosc probkowania
dt=1/Fp;
t=0:dt:10-dt;%wektor czasu
y=sinc(t);
5 plot(t,y);%rysowanie funkcji sinc

N=size(t);
N=N(2);
sum=0.0;
10 R=zeros(1,N);
```

8. Wczytaj sygnał "`sygnal.bin`" zawierający dane spróbkowane z częstotliwością 1200 Hz i zapisane binarnie na 13 kanałach (`[13 inf]`), z precyzją '`float32`'. Narysuj fragment wczytanego sygnału trwający od 9 do 10 sekundy, dla pierwszych pięciu kanałów (każdy kanał na osobnym wykresie - użyj komendy `subplot`).

```
clear all
close all
%.....
```

```
5 | s=fopen('sygnał.bin','rb');  
   | A=fread(s,[13 inf],'float32');  
   | fclose(s);
```

## 9. Ściąga

```

disp('Ściąga z Matlab') %wypisanie na ekran
%tak można wstawić komentarz
%Ważne skroty: F1-pomoc, F5-save file and run, Ctrl+R -zakomentowanie
                %zaznaczonego fragmentu, Ctrl+T -odkomentowanie

5   a=2;
    %Usunięcie zmiennej z pamięci:
    clear a    % usuwa zmienna a;
    clear all  % usuwa wszystkie zmienne znajdujące się w pamięci

10  %Zapisanie zmiennych na dysku:
    save nazwapliku %domyslnie przyjmowane jest rozszerzenie .mat

    %Wczytanie danych z pliku:
    load nazwapliku

15  %Korzystanie z podręcznej pomocy podającej opis funkcji:
    help sin

    % Działania: + - * / ^

20  %Wybrane funkcje:
    b=5;
    n=3;

25  a=abs(b)          %wartosc bezwzględna, modul liczby zespolonej
    a=cos(b)          %cosinus (przedrostek "a" oznacza funkcję arcus zas litera
                        %"h" na końcu oznacza hiperboliczny

    a=sin(b)          %sinus
    a=tan(b)          %tanges
30  a=cot(b)          %cotangens
    a=angle(b)        %kat fazowy liczby zespolonej w radianach
    a=conj(b)         %liczba sprzężona
    a=exp(b)          %funkcja eksponencjalna
    a=imag(b)         %czesc urojona liczby zespolonej
35  a=log(b)          %logarytm naturalny
    a=log10(b)        %logarytm o podstawie 10
    a=real(b)         %część rzeczywista liczby zespolonej
    a=round(b)        %zaokraglenie
    a=sign(b)         %znak funkcji
40  a=sqrt(b)         %pierwiastek kwadratowy
    a=nthroot(b,n)    %pierwiastek n-tego stopnia
    a=eye(b)          %tworzenie macierzy jednostkowej
    a=ones(b)         %macierz jedynekowa
    a=zeros(b)        %macierz zerowa
45  a=rand(b)         %macierz losowa o rozkładzie równomiernym

    %Wybrane zmienne i stałe:
    ans              %standardowa zmienna przypisywana do wyrażenia nieskojarzonego
50  pi              %liczba pi
    i, j            %jednostka urojona
    date            %aktualna data

```

```
55 clock %aktualny czas [year month day hour minute seconds]
    Inf %nieskończonosc
    NaN %Not-a-Number

%Operatory logiczne w języku MATLAB:
    a=1;b=2;

60 % a == b równe
    % a ~= b różne
    % a < b mniejsze
    % a > b większe
65 % a <= b mniejsze lub równe
    % a >= b większe lub równe
    % a & b i
    % a | b lub

70 %Wektory i macierze
    A=[1,2,3,4] %wektor poziomy
    B=[1;2;3;4] %wektor pionowy
    C=1:10 %generowanie wektora [1 2 3 4 5 6 7 8 9 10]
75 D=1:2:10 %generowanie wektora [1 3 5 7 9]
    E=[1,2,3;4,5,6;7,8,9] %macierz 3x3
    F=E' %macierz transponowana
    L=length(A) %podaje dlugosc wektora lub rozmiar najdluzszego wiersza
    %lub kolumny macierzy
80 S=size(A) %podaje rozmiar wektora lub macierzy

%funkcje tworzenia macierzy:
%[rand,ones,eye,randi,randn,zeros,toeplitz,diag,magic,hilb]

85 %Grafika dwuwymiarowa
    x=[1 2 1 1 0 3 4]
    plot(x)

90 %subplot(m,n,p)
    %m - liczba wykresów w pionie
    %n - liczba wykresów w poziomie
    %p - kolejny numer wykresu

95 %axis([xmin xmax ymin ymax]); zmiana skali wykresu

%opis rysunku:
    title('tytuł rysunku');
    xlabel('opis osi x');
100 ylabel('opis osi y');
    legend('legenda');

105
```

```
%Obsługa plików
% fopen - otwarcie pliku
% plik = fopen('nazwa.txt','rt'); - otwarcie pliku txt do czytania
% plik = fopen('nazwa.txt','rt'); - otwarcie pliku txt do pisania
110 % plik = fopen('nazwa.txt','rb'); - otwarcie pliku binarnego do czytania
% plik = fopen('nazwa.txt','wb'); - otwarcie pliku binarnego do pisania
% fread - czytanie danych z pliku binarnego
% fwrite - zapis danych do pliku binarnego
% fprintf - zapis danych do pliku tekstowego
115 % fscanf - czytanie danych z pliku tekstowego
% fclose(plik)- zamknięcie pliku

%Przykład odczytu
FileName='dane1.bin';
120 fid1=fopen(FileName,'rb');
A=fread(fid,[Channels inf],'float32');
fclose(fid1);

%Przykład zapisu
125 fid2 = fopen('dane2.bin','w');
B = reshape([1:12],4,3);
fwrite(fid2,B);
fclose(fid2);

130 %Petla FOR
for i = 1:N
    ciag instrukcji;
end

135 %Petla WHILE
while (wyrażenie warunkowe)
    ciag instrukcji;
end

140 %Petla IF
if (wyrażenie warunkowe)
    ciag instrukcji;
end
```