

Politechnika Gdańska
Wydział Elektrotechniki i Automatyki
Katedra Inżynierii Systemów Sterowania

Teoria sterowania

MATLAB – funkcje zewnętrzne (m-pliki, funkcje)

Materiały pomocnicze do ćwiczeń laboratoryjnych 1 – Część VI

Opracowanie:

Michał Grochowski, dr inż.

Robert Piotrowski, dr inż.

Bartosz Jaworski, mgr inż.

Gdańsk

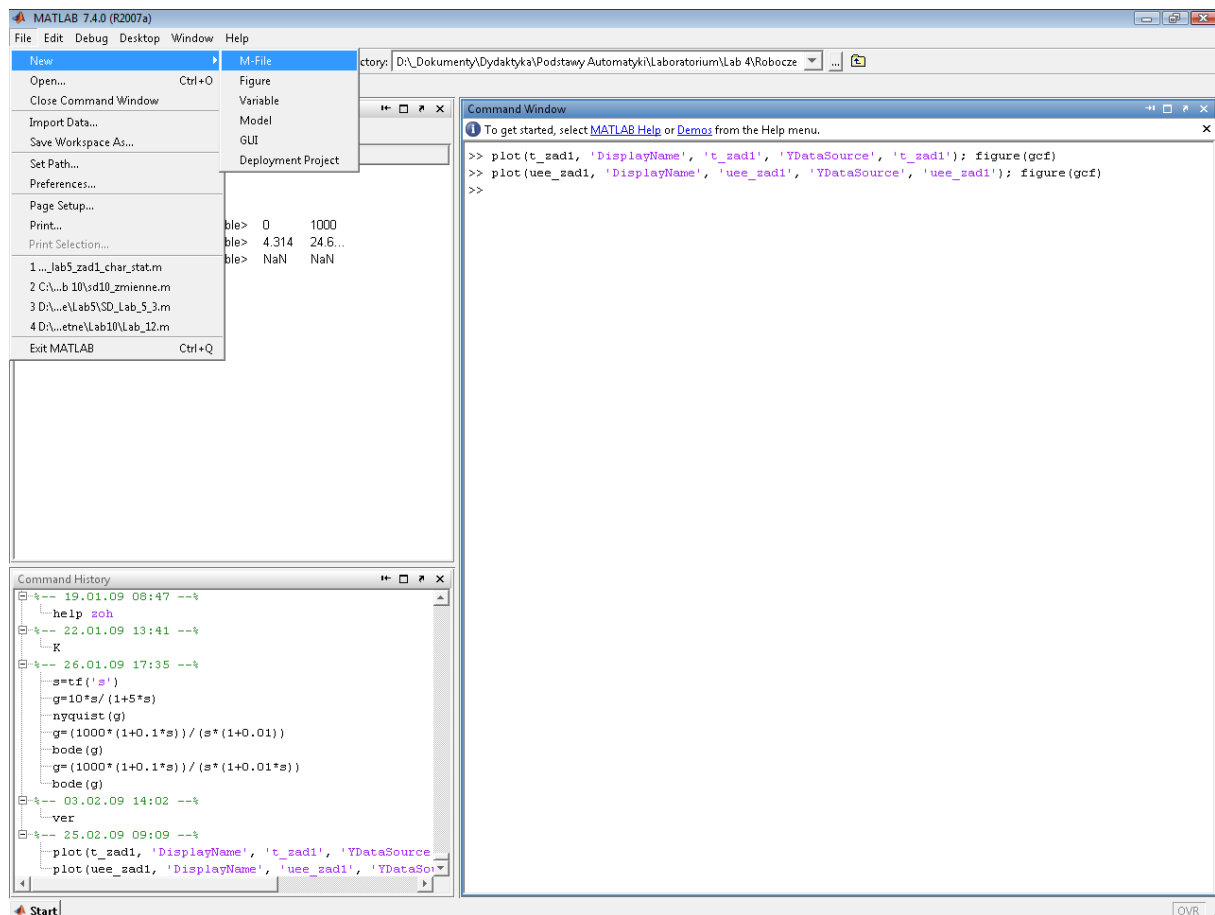
W MATLAB'ie użytkownik ma do dyspozycji dwa rodzaje funkcji zewnętrznych: skrypty i funkcje. Każda z tych funkcji ma postać pliku tekstowego ASCII i jest tworzona za pomocą dowolnego edytora tekstowego. Dzięki możliwości tworzenia własnych funkcji, MATLAB uzyskał właściwości języka proceduralnego.

Funkcje zewnętrzne – skrypty (m-pliki)

MATLAB jest zwykle wykorzystywany w trybie interaktywnym; kiedy zostanie wprowadzone polecenie, MATLAB przetwarza je natychmiast i wyświetla odpowiedź. MATLAB jest jednak w stanie wykonywać sekwencję poleceń zapisanych w pliku. Plik dyskowy zawierający polecenia MATLAB'a jest nazywany M-plikiem, ponieważ rozszerzeniem nazwy takiego pliku jest **.m** (Rys. 1). Na przykład, plik o nazwie `bessel.m` zawiera polecenia MATLAB'a, które służą obliczeniu funkcji Bessel'a.

M-pliki nazywane są też funkcjami zewnętrznymi MATLAB'a.

Dowolny M-plik składa się z poleceń MATLAB'a, wśród których mogą znajdować się odwołania do innych M-plików. Dowolny M-plik może odwoływać się do samego siebie rekursywnie. Przykładowy m-plik pokazany jest na Rys. 2.



Rys. 1. Otwarcie nowego m-pliku

```

1 % wysokość maksymalna zbiorników
2 hmax = 2;
3
4 % w ilu punktach policzyć
5 ile = 10;
6
7
8 % jak to się przeloży na przyrost h
9 dh = hmax/ile;
10
11 % dane zbiorników
12 A = 1.45; % zbiornik o stałym przekroju (cylindryczny lub szescian)
13 % A - średnica podstawy cylindrycznej lub szerokość kwadratu w
14 % podstawie
15
16 alfa = 60; % kat w stopniach dla zbiornika - odwrócony stożek
17
18 R = hmax/2; % promień zbiornika kulistego
19
20 V1 = zeros(ile,1);
21 V2 = zeros(ile,1);
22 V3 = zeros(ile,1);
23
24 i = 1;
25
26 for h=0:dh:(dh*ile)
27
28     V1(i,1) = A * A * h; % szescian
29     V2(i,1) = (pi/(tand(alfa)*tand(alfa)))*h^2;
30     V3(i,1) = pi*(2*R*(h^2/2)-(h^3/3));
31
32     i = i + 1;
33 end
34
35 figure;
36 plot(0:dh:(dh*ile),V1,0:dh:(dh*ile),V2,0:dh:(dh*ile),V3);
37 xlabel('wysokość zbiorników, h');
38 ylabel('V(h)');
39 legend(['V1 - szescian'],['V2 - odwrócony stożek'],['V3 - kula'],'Location','NorthWest');
40 title('Charakterystyki statyczne zbiorników V(h)')
41

```

Rys. 2. Przykładowy m-plik

Skrypty są zestawem większych lub częściej używanych poleceń (zbiorem tekstowym zawierającym polecenia), które mają być wykonane przez interpreter MATLAB'a. Skrypt nie musi spełniać żadnych dodatkowych wymogów formalnych poza poprawnością znajdujących się w nim poleceń.

Wywołanie skryptu dokonuje się przez podanie jego nazwy w wierszu poleceń lub w skrypcie. W jego efekcie zostaną wykonane wszystkie instrukcje znajdujące się w skrypcie.

Komentarze w skrypcie poprzedza się znakiem %. Komentarze nie są analizowane przez interpreter, a pełnią tylko rolę dokumentacyjną.

Przykład 1

```

% Przykładowy skrypt MATLAB'a kreślący wykres funkcji
x=-5:0.1:5;
y=x.^2;
plot(x,y);
grid;
title('Przykładowy wykres funkcji');
xlabel('x');
ylabel('y');

```

Funkcje zewnętrzne – funkcje

Definicję funkcji umieszcza się w skrypcie o nazwie identycznej z nazwą definiowanej funkcji i rozszerzeniem `.m`. Funkcje przyjmują argumenty wejściowe oraz zwracają argumenty wyjściowe.

Definicja funkcji w MATLAB'ie ma postać:

```
function [argumenty wyjściowe] = nazwa_funkcji(argumenty wejściowe)
    ciąg-instrukcji
```

W przypadku, gdy funkcja nie zwraca argumentów wyjściowych może zostać zapisana w następującej postaci:

```
function nazwa_funkcji (argumenty wejściowe)
```

Analogicznie jak w skrypcie, komentarze w funkcji poprzedza się znakiem `%`.

W zasadniczej części funkcji mogą znajdować się wywołania do innych funkcji, konstrukcje programowe realizujące operacje wejścia/wyjścia, obliczenia, komentarze, linie puste itp.

W funkcji wyróżniamy dwa rodzaje zmiennych: lokalne i globalne.

Zmienne utworzone w trakcie wykonywania funkcji to zmienne lokalne. Zmienne te nie są dostępne w przestrzeni roboczej MATLAB'a.

Zmienne globalne są dostępne w przestrzeni roboczej MATLAB,a i poprzedza je słowo **global**.

Weźmy więc analizowany wcześniej przykład m-pliku i zamieńmy go na funkcję:

Przykład 2

```
function wykres (a,b)

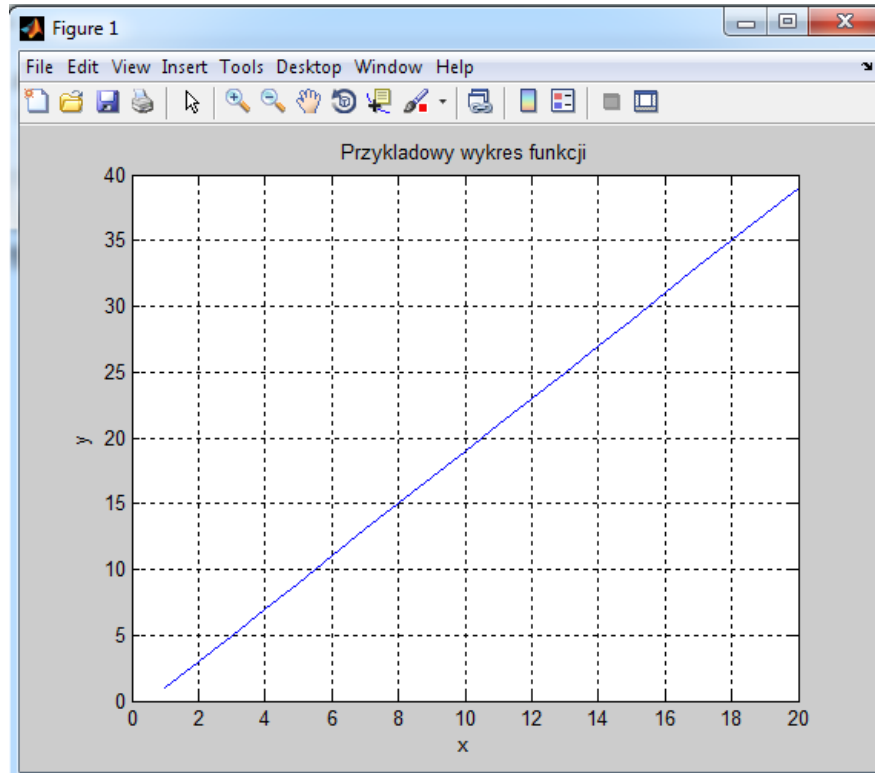
plot(a,b);
grid;
title('Przykładowy wykres funkcji');
xlabel('x');
ylabel('y');
```

Zapisując podany wyżej kod jako `wykres.m` (**Ważne! m-plik z funkcją musi się nazywać identycznie jak sama funkcja**) będziemy mogli w głównym oknie środowiska Matlab wywołać funkcję `wykres` równie wygodnie jak np. funkcję `sinus`:

```

>> a=[1:1:20]
a =
     1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19    20
>> b=[1:2:40]
b =
     1     3     5     7     9    11    13    15    17    19    21    23    25    27    29    31    33    35    37    39
>> wykres(a,b)
f: >>

```



Inny przykład funkcji, która posiada argument wyjściowy:

Przykład 3

```

function [Z] = odcinanie(Z,m)

x=size(Z,1);
z=size(Z,2);

for a=1:x
    for b=1:z

        if Z(a,b)>m
            Z(a,b)=randi([0,m]);
        end
    end
end;

disp(Z);

```

Kod ten służy do limitowania wartości macierzy, dzięki czemu macierz b:

```
b=[1 12 3 5 6]
```

po wywołaniu:

```
b=odcinanie(b,12)
```

stanie się macierzą:

```
b=[1 3 3 5 6]
```

lub podobną.

Kolejny przykład funkcji, która posiada kilka argumentów wejściowych i wyjściowych (funkcja wyliczająca sumę oraz iloczyn dwóch liczb):

Przykład 4

```
function [iloczyn,suma]=moja_funkcja(x,y) %Defin. nazwy oraz argumentów
iloczyn = x*y; %Przypis. wart. wyjściu „iloczyn”
suma = x+y; %Przypis. wartości wyjściu „suma”
```

Użycie własnej funkcji:

```
>> moja_funkcja(1,2) %Komenda zwróci tylko pierwszą wartość
ans =
     2

>> [iloczyn,suma] = moja_funkcja(1,2) %Komenda zwróci obydwie wartości

iloczyn =
     2

suma =
     3
```

Bibliografia

Brzózka J., Dorobczyński L. *Matlab – środowisko obliczeń naukowo – technicznych*. Wydawnictwo MIKOM, 2005.
Mrozek B., Mrozek Z. *Matlab i Simulink. Poradnik użytkownika. Wydanie II*. Wydawnictwo HELION, 2004.
Zalewski A., Cegiela R. *Matlab – obliczenia numeryczne i ich zastosowania*. Wydawnictwo NAKOM, 1996.