Modelowanie systemów dynamicznych

Krzysztof Wesołowski,
grupa, 11:00, s. 317c
Prowadzący: dr inż. Adam Piłat
Data odbycia zajęć: 5 X 2009

Sprawozdanie numer 1

Część I – podsumowanie zajęć.

Zajęcia służyły głównie zapoznaniu się z Matlabem i jego linią komend. Po odświeżeniu posiadanej wiedzy wykonano kilka skryptów które tutaj zamieszczam (nie ma potrzeby przytaczać pierwszych ćwiczeń z obsługi/składni gdyż uwzględnione są w części II).

Listing 1 – tworzenie skryptu, aproksymacja, rysowanie wykresu

```
x=0:0.1:6*pi; % zbiór argumentów od 0 do 6pi, rozdzielczo?? 0.1
y=x.^2.*cos(x); % tworzenie wektora wartosci funkcji x^2*cos(x)
                % warto zwrócic uwage na kropke po x oznaczajaca
                % potegowanie "element wise", oraz po 2, aby wynik tego
                % potegowania przemnozyc w ten sam sposób przez wektor
                % cosinusów argumentów (funkcje dzialajace na pojedynczej
                % liczbie automatycznie po podaniu im jako argumentu
                % wektora dzialaja jako element-wise

p=polyfit(x,y,3);    % aproksymacja wielomianowa zadanych punktów (podanych
                     % w formie wektora argumento i wektora wartosci -
                     % trzeci argument oznacza stopie? wielomianu.
                     % Wynikiem dzialania jest wektor wspó?czynników
                     % wielomainu aproksymujacego

y1=polyval(p,x);     % Polynomial value - zwraca wartosc wielomianu o
                     % wspolczynnikach p dla punktu x. Równiez
automatycznie
                     % sobie radzi gdy x jest wektorem

plot(x,y,'k',x,y1,'r') % rysowanie obu wykresów, 'k' czarny, 'r' czerwony

% Warto zwrocic uwage, iz wywolanie skryptu jest praktycznie rownoznaczne
z
% uruchomieniem kolejnych polecen - co wplywa na cala zawartosc aktualnego
% workspace (zmienne), nie zmieniajac tylko historii polecen.
```

Listing 2 – tworzenie funkcji, obliczanie bledu aproksymacji

```
function e = wesoly_2(stopien)   % funkcja zwraca 1 element - e
                                 % i przyjmuje 1 argument - stopien
    x=0:0.1:6*pi;
    y=x.^2.*cos(x);
    p=polyfit(x,y,stopien);
    y1=polyval(p,x);
    blad=y-y1;                   % roznica wektorow
    plot(x,y,'b',x,y1,'g');      % rysowanie obu, nie jest konieczne
    e=blad;                      % zwraca wektor - funkcje bledu

% Tak przygotowana funkcja tylko zwraca wartosc - zmieniajac odpowiednia
% zmienna lub zmienna automatyczna ans. Wszelkie zmienne w jej ciele sa
% tylko lokalne.
```

Listing 3 – skrypt wywołujący funkcje, wielokrotne wykresy

```
for i= 1:8           % petla od 1 do 8 wlacznie
    figure(i);       % wybranie itego obrazu - domyslny to pierwszy, wiec
                     % tutaj stworzymy 8 odrebnych rysunkow
    wesoly_2(i);     % wywolanie funkcji ktora nie tlyko oblicza, ale
                     % i rysuje wykres na aktualnie aktywnym obszarze
end;
```

Listing 4 – skrypt wywołujący funkcje, okno podzielone na wykresy

```
for i= 1:8
    subplot(2,4,i); % wybranie aktywnej czesci okna figure1 (domyslnego)
                    % pierwsze dwa argumenty to ilosc wierszy i kolumn,
                    % ostatni to aktualnie wybrana czesc okna.
    wesoly_2(i);    % rysowanie na aktywnym obszarze - wybranej 1/8 okna
end;
```

W sprawozdaniu pomijam listing z transformatą Fouriera – można go łatwo wyświetlić za pomocą polecenie type: type fftdemo

Część II – zadania

Zadanie A

```
>> A=[2 1 3;-2 0 -1] ;
>> B=[1 -1 2;2 3 -1;-3 2 1];
>> C=A*B
C =
    -5     7     6
     1     0    -5
```

Zadanie B

```
>> C=[C; 16,10,5]
>> D=inv(C);
>> D*C
ans =
    1.0000         0   -0.0000
         0    1.0000    0.0000
         0         0    1.0000
```

Zadanie C

```
>> D=[D,[12;6;5]]
>> D=[D;15,2,7,3]
```

Zadanie D

```
>> [WektW,WartW]=eig(D)
WektW =
    0.5453    0.6085   -0.4359   -0.1664
    0.2801    0.2943    0.2146   -0.7927
    0.2243    0.2573    0.8740    0.5864
    0.7575   -0.6906   -0.0031   -0.0082

WartW =
   16.6092         0         0         0
         0  -13.6782         0         0
         0         0   -0.0530         0
         0         0         0    0.2213
```

Zadanie E

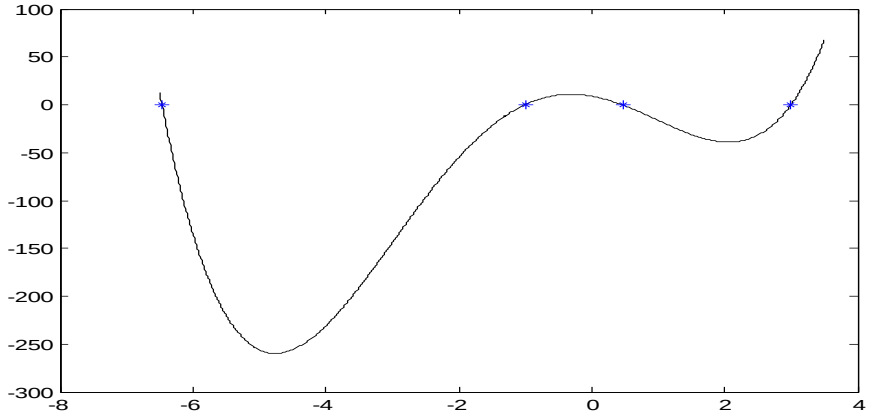
Listing 5 – zadanie_5.m

```
wiel=[1 4 -18 -12 9];
pierwiastki=roots(wiel);
x=-6.5:0.01:3.5;
y=polyval(wiel,x);
plot(x,y,'k',pierwiastki,zeros(1,length(pierwiastki)),'*');
pierwiastki
```

```
>> zadanie_5

pierwiastki =

   -6.4641
    3.0000
   -1.0000
```



Zadanie F

Listing 6 – func.m

```
function val = func(arg)
    val= (sin(arg).^2+ 2*sin(arg)).*cos(arg).^2;
```

Gdy mamy taka funkcje wywołania sprowadzają się do:

```
x=0:GESTOSC:4*pi; plot(x,func(x));
```

Ewentualnie dla lepszego porównania warto nałożyć wykresy:

```
hold on;
```

GESTOSC	Obserwacje
1	Wykres odtwarza tylko zarys, Matlab domyślnie używa spline'ów 1 stopnia do interpolacji funkcji na wykresie.
0.1	Wykres czytelny przy małym oknie, przy maksymalizacji można zauważyć interpolacje prostymi gołym okiem
0.01	Praktycznie płynny wykres, ewentualne bledy można zauważyć tylko przy dużej pochodnej (bo mamy stały krok niezależnie od jej wartości)
0.001	Wykres o jakości powyżej możliwość monitora (chyba że użyjemy dodatkowego zoom)
0.000001	Obciążenie komputera staje się widoczne, trzeba poczekać na wyrenderowanie wykresu.

Podsumowując – należy dopasowywać rozdzielczość do danych wejściowych (nie ma sensu używać wyższej), zaś gdy operujemy na funkcjach analitycznych dostosować ją do potrzeb, w szczególności gdy celem jest tylko wyświetlanie.

Nie przytaczam tutaj wykresów, gdyż zwłaszcza w formacie PDF i ogólnie w dokumencie różnice te są widoczne tylko dla pierwszych 2 gęstości, poza tym wykres taki można łatwo samemu wygenerować korzystając z dołączonego kodu.

System – inaczej układ, o określonych wejściach i wyjściach, jest ogólnym określeniem obiektu sterowania lub całego układu regulacji. Samo pojęcie systemu oznacza zespół elementów, wraz z opisem ich działania jak i relacji między nimi.