Universität Bremen

Fachbereich Produktionstechnik
Bremer Institut für Messtechnik,
Automatisierung und Qualitätswissenschaft
Prof. Dr.-Ing. habil. A. Fischer

BIMAQ

Automatisierungstechnik

Universität Bremen
Fachbereich Produktionstechnik
Bremer Institut für Messtechnik,
Automatisierung und Qualitätswissenschaft
Prof. Dr.-Ing. habil. A. Fischer
BIMAQ

Automatisierungstechnik - Übung 1

Aufgabe 1: Matlab – Wertzuweisung, Script

Lösen Sie bitte folgende Aufgaben mit Hilfe der Hilfe von Matlab (Befehle *doc* und *help*).

Hilfreiche Stichworte sind:

- *general* General purpose commands
- *ops* Operators and special characters
- *elfun* Elementary math functions
- *elmat* Elementary matrices and matrix manipulation
- *lang* Programming language constructs

a) Berechnen Sie im Command Window Fläche und Umfang eines Kreises mit dem Radius r=3. Verwenden Sie dafür die Konstate *pi* und die Funktion *sqrt* (siehe auch *help elfun*).

b) Löschen Sie die auf dem Workspace angelegten Variablen mit `clear`.

c) Schreiben Sie ein einfaches Skript *kegel.m* mit der Funktion für das Volumen.

$$V = \frac{1}{3} r^2 h$$

Geben Sie r=3 und h=2 vor. Was passiert mit dem Workspace, wenn Sie *kegel* ausführen?

d) Definieren Sie im Skript vor der Berechnung r=5 und rufen Sie das Skript erneut auf. Was passiert, wenn sich die auf dem Command Window definierte Variable und die im Skript unterscheiden?

Aufgabe 2: Matlab – Vektoren und Matrizen

Hilfe zu den elementaren Matrizen-Funktionen finden sie unter dem Stichwort *elmat*.

a) Erstellen Sie sich einen Vektor mit den Zahlen 1 bis 5 (*help colon*).

b) Erstellen Sie sich zwei 3x3 Matrizen und wenden Sie nacheinander die Operatoren „+“ ,„*“ , „ .*“, „.^“, dot(a,b) an.

c) Vergleichen Sie die beiden Ergebnisse für „*“ und dot(a,b) . Wie ist das Ergebnis zu erklären?

d) Teilen Sie eine der Matrizen aus b durch 0? Welchen Unterschied zu Programmiersprachen wie Java, C/C++, Python (ohne Paket numpy) gibt es?

e) Erstellen Sie eine Diagonalmatrix mit dem Vektor aus a)

f) Erstellen sie folgende Matrix ohne sie direkt, elementweise zu definierten!

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

g) Geben Sie alle Werte der ersten Spalte einer Matrix aus b) aus.
(Hilfestellung: „:“ als Operator)

Aufgabe 3: Matlab – Schleifen

Berechnet werden soll die Position einer Masse m=1200 kg, die mit einer Kraft F=800 N für T=5 s lang von $v_0 = 0$ beschleunigt wird. Die Position soll alle h=0.5 Sekunden angegeben werden.

Das Weg-Zeit-Gesetzt lautet: $s = \frac{1}{2} \cdot a \cdot t^2 + v_0$ mit $F = m \cdot a$

a) Schreiben Sie ein Skript zur Berechnung des Positionsvektor s mit Hilfe einer For-Schleife
(konventionelle Programmierung).
b) Legen Sie im Skript aus a) einen neuen Abschnitt (section) an und berechnen Sie den Positionsvektor s indem Sie für die Zeit einen Vektor verwenden.
c) Welche Vorteile könnte die zweite Schreibweise für die Bearbeitung auf Multiprozessor-Systemen haben und wieso wird sie durch diese Formulierung möglich?

Aufgabe 4: Matlab – Grafische Ausgabe

Geben Sie das Ergebnis aus Aufgabe 3: grafisch aus und erweitern Sie dafür das dort erstellte Skript.

a) Verwenden Sie den Befehl `plot` um den Positionsvektor auszugeben.
b) Ergänzen Sie die Beschriftung für die beiden Achsen und geben Sie der Grafik eine Überschrift. Markieren Sie weiterhin jeden berechneten Punkt auf der Kurve.
c) Geben Sie die Position über der Zeit aufgetragen für die vektorielle Berechnungsmethode an.
d) Markieren Sie das Ergebnis für t = 2.5 s und lesen Sie den berechneten Wert für Y aus der Grafik ab.

Aufgabe 5: Matlab – Function

Erstellen Sie mit Matlab eine neue Funktion („New → Function“) und implementieren Sie die Berechnung aus Aufgabe 3:, so dass Zeit und Position inkl. Startwert jeweils eine Spalte des Ausgabevektors sind.

a) Weisen Sie das Ergebnis der Funktion einer einzelnen Variablen zu und geben Sie die Position wie in Aufgabe 4: b) aus.
b) Weisen Sie den Rückgabewert einem Vektor aus Position und Zeit zu und geben Sie das Ergebnis in einem Plot aus (Position über Zeit).
c) Geben Sie für a) nur das 2-Tupel für den Endwert aus.

Kurzlösungen

Lösung 1

a) Matlab-Code

```
r = 3
A = pi * r^2
A =
   28.2743
```

b) Matlab-Code

```
clear r A
```

c) Datei kegel.m mit einer Zeile

```
V = 1/3 * r^2 * h;
```

Matlab-Code

```
r = 3;
h = 2;
kegel
```

Auf dem Workspace liegt dann die Variable V = 6.

d) Datei kegel.m mit einer Zeile

```
V = 1/3 * r^2 * h;
```

Matlab-Code

```
r = 3;
h = 2;
kegel
```

Auf dem Workspace liegt dann die Variable V = 16.667

Lösung 2

a) `v = 1:5`

b) zum Beispiel:

```
a = [1,2,3; 4,5,6; 7,8,9]
b = [9,8,7; 6,5,4; 3,2,1]
```

a+b =

```
  10   10   10
  10   10   10
```

```
   10   10   10

a*b =
   30   24   18
   84   69   54
  138  114   90
= dot(a,b,3)

a.*b =
    9   16   21
   24   25   24
   21   16    9

a.^b =
        1      256     2187
     4096     3125     1296
      343       64        9
```

c) dot(a,b) behandelt a und b jeweils als Sammlung von Vektoren. Um das gleiche Verhalten zu erreichen, muss dot(a,b,3) verwendet werden.

d) a/0 =

```
  Inf   Inf   Inf
  Inf   Inf   Inf
  Inf   Inf   Inf
```

Matlab „beherrscht“ die Division durch 0. In anderen Sprachen führt dies in der Regel zu einem Fehler, der gesondert behandelt werden muss, damit es nicht zu einem Programmabbruch kommt.

e) `diag(1:5)`

f) `eye(5) + fliplr(eye(5))`

g) `a(:,1)`

Lösung 3

a) und

b) Matlab-Skript

```
%% Konstanten
m = 1200;
F = 800;
a = F/m;
v0 = 0;
h = 0.5;
T = 5;
```

```
%% Berechnung konventionell
k = T/h;
for i=0:k
    t1 = i * h;
    s1(i+1) = 0.5 * a * t1^2 + v0;
end
%% Berechnung vektoriell
t2 = 0:h:T;
s2 = 0.5 * a * t2.*t2 + v0;
```

c) Im zweiten Fall sind weniger Anweisung erforderlich und der Interpreter/Compiler kann die Berechnung für die parallele Berechnung auf mehrere Prozessoren optimieren. Damit ergibt sich ein Geschwindigkeitsvorteil bei komplexen Berechnungen.

Lösung 4

Erweitern des Skripts aus der vorherigen Aufgabe.

a) `plot(s1)`

b)
```
plot(s1,'-+')
xlabel('Schrittnummer')
ylabel('Position in m')
title('Bewegungsprofil')
```

c) `plot(t2,s2,'-*')`

d) Data Cursor liefert X:2.5, Y:2.083

Lösung 5

Matlab-Funktion:

```
function [ s, t ] = Beschleunigung( m, F, T, h, v0 )
    a = F/m;
    t = 0:h:T;
    s = 0.5 * a * t.^2 + v0;
end
```

a) erg = Beschleunigung(1200, 800, 5, 0.5, 0)
Für Plot kann das abgeänderte Skript aus Aufgabe 4 a) verwendet werden.

b) [Position, Zeit] = Beschleunigung(1200, 800, 5, 0.5, 0)
Für Plot kann das abgeänderte Skript aus Aufgabe 4 b) verwendet werden.