

Matlab Hinweise  
**Digitale Signalverarbeitung I/II**  
– SS 2010 –

**Henning Paul**

NW1, Raum N2360, Tel.: 0421/218-62399

eMail: [paul@ant.uni-bremen.de](mailto:paul@ant.uni-bremen.de)



Universität Bremen, FB1  
Institut für Telekommunikation und Hochfrequenztechnik  
Arbeitsbereich Nachrichtentechnik  
Prof. Dr.-Ing. A. Dekorsy  
Postfach 33 04 40  
D-28334 Bremen

WWW-Server: <http://www.ant.uni-bremen.de>

Version vom 28. Juli 2010

## A Folien zur Übung

### Zuordnung von Zahlen, Vektoren und Matrizen

`k = 1;` Variable  $k$  wird der Wert 1 zugewiesen, Semikolon unterdrückt Bildschirmausgabe

`k = 1:20;` Array (Zeilenvektor)  $k$  mit den Elementen 1 bis 20

`k = 2:5:25;` Array (Zeilenvektor)  $k$  mit Werten von 2 bis 25 mit der Schrittweite 5  
 $k = [2, 7, 12, 17, 22]$

`k = [1,2,3];` Array (Zeilenvektor)  $k = (1, 2, 3)$

`k = [1;2;3];` Array (Spaltenvektor)  $k = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$

`k = [1,2,3;7,1,4;3,6,9]` Matrix  $k = \begin{pmatrix} 1 & 2 & 3 \\ 7 & 1 & 4 \\ 3 & 6 & 9 \end{pmatrix}$

`k = k(:);`  $k$  in Spaltenvektor wandeln (Spaltenweise von links nach rechts)

$$k = \begin{pmatrix} 1 \\ 7 \\ 3 \\ 2 \\ 1 \\ \vdots \end{pmatrix}$$

`x = k(1);`  $x$  erstes Element aus  $k$  zuweisen (Index beginnt mit 1 nicht mit 0 wie bei C)

`x = k(1,2);`  $x$  wird aus  $k$  1. Zeile das 2. Element zugewiesen (also hier 2)

`k = [a;b;c];` Matrix  $k = \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 7 & 1 & 4 \\ 3 & 6 & 9 \end{pmatrix}$  mit den Zeilenvektoren  
 $a = [1, 2, 3]$ ,  $b = [7, 1, 4]$ ,  $c = [3, 6, 9]$

`k = [A,B,C];` Matrix  $k = \begin{pmatrix} A & B & C \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 7 & 1 & 4 \\ 3 & 6 & 9 \end{pmatrix}$  mit den Spaltenvektoren  
 $A = \begin{pmatrix} 1 \\ 7 \\ 3 \end{pmatrix}$ ,  $B = \begin{pmatrix} 2 \\ 1 \\ 6 \end{pmatrix}$ ,  $C = \begin{pmatrix} 3 \\ 4 \\ 9 \end{pmatrix}$

## Zuordnung von Zahlen, Vektoren und Matrizen

`x = k(1:3,1:2);`       $x$  wird aus  $k$  1. bis 3.Zeile jeweils 1. und 2.Element gebildet

$$\begin{pmatrix} 1 & 2 \\ 7 & 1 \\ 3 & 6 \end{pmatrix}$$

`k = zeros(y,x);`      Erzeugt Matrix mit  $y$  Zeilen und  $x$  Spalten initialisiert mit 0.

`k = zeros(z);`      Erzeugt  $Z \times Z$  Matrix initialisiert mit 0.

`k = ones(y,x);`      Erzeugt Matrix analog zu `zeros()` initialisiert mit 1.

## Operatoren

`+ -`      Addition, Subtraktion

`* / ^`      (Matrix-)Multiplikation/ Division / Potenz

`.* ./ .^`      Elementweise Multiplikation/ Division / Potenz

## Grafik

`stem(x,y)`



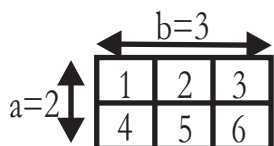
$y$ -Werte über  $x$ -Achse aufgetragen. `stem(y)` erzeugt eigene  $X$ -Achse mit der Länge von  $y$

`plot(x,y,option)`



Analog zu `stem()`, mit zusätzlichen Optionen (Farbe, Linien-Form..)

`subplot(a,b,c)`



unterteilt das Bild und legt aktuelles Teilbild fest

---

## Grafik

---

<code>figure</code>	Öffnet ein neues Figure.
<code>figure(x)</code>	Figure mit der Nummer x wird als aktuelles Figure gesetzt.
<code>hold on</code>	Aktuelles Bild wird nicht mehr überschrieben. Neue Plots werden eingefügt.
<code>hold off</code>	Aktuelles Bild wird freigegeben. Nur hold springt zw. frei-gegebenen und festen Zustand hin und her.
<code>grid</code>	Gitter an aus, analog zu hold
<code>axis([x_min x_max y_min y_max])</code>	Achsenausschnitt fest
<code>xlabel('Text')</code>	X-Achsenbeschriftung, (ylabel(), xlabel(), title() analog)

---

## Systemtheorie

---

<code>[h,w]= freqz(b,a,1)</code>	Berechnung des Frequenzgangs h und der entsprechenden Kreisfrequenzen w für ein digitales Filter, dargestellt durch die Vektoren b und a.
<code>zplane(z,p)</code>	Darstellung der Pol- und Nullstellen eines zeitdiskreten Systems. z: Spaltenvektor der Nullstellen, p: Spaltenvektor der Polstellen.
<code>zplane(b,a)</code>	Darstellung der Pol- und Nullstellen eines zeitdiskreten Systems. b: Zeilenvektor der Zählerkoeffizienten, a: Zeilenvektor der Nennerkoeffizienten.
<code>[h,t] = impz(b,a)</code>	Berechnung der Impulsantwort eines Filters, dargestellt durch die Vektoren a und b.
<code>w = conv(u,v)</code>	Faltung des Vektors u mit dem Vektor v.
<code>y = filter(b,a,X)</code>	Filterung des Datenvektors X mit einem Filter, dargestellt durch die Vektoren a und b.

---

## wichtige Funktionen

---

<code>i, j, pi</code>	vorgegebene Konstanten (können überladen werden)
<code>real(x)</code>	Real-Teil von $x$
<code>imag(x)</code>	Imaginär-Teil von $x$
<code>help fkt</code>	gibt Hilfstext zur Funktion <code>fkt</code> , z.B: <code>help imag</code>
<code>cos(x)</code>	Kosinus ( $x$ in rad) analog für <code>sin(x)</code> , <code>tan(x)</code> , <code>acos(x)</code> ...
<code>fft(x,N)</code>	$N$ -Punkte FFT. Falls $N$ größer als Länge von $X$ auffüllen mit Nullen. Analog <code>ifft</code> , $N$ optional.
<code>[a,b]=size(x)</code>	gibt die Größe von $x$ aus ( $a$ = Anzahl der Zeilen, $b$ = Anzahl der Spalten)
<code>length(x)</code>	Länge des Vektors $x$ , entspricht <code>max(size(x))</code>
<code>sum(x)</code>	Summe des Vektors $x$ bzw. Summe aller Spalten der Matrix $x$
<code>max(x)</code>	Bildet das Maximum, analog zu <code>sum(x)</code> bzw. <code>min(x)</code>

---

## Steuerung des Programmablaufs

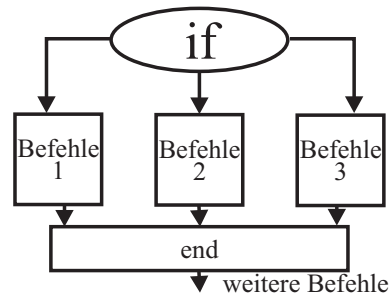
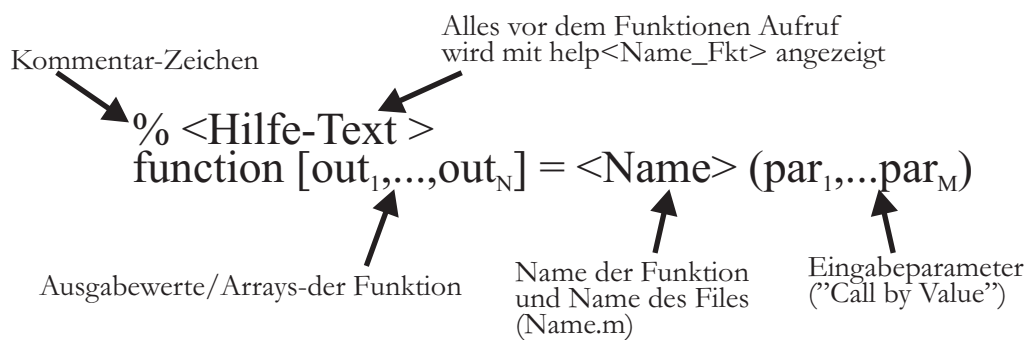
---

<code>for i = a:s:e</code> other statements <code>end;</code>	<code>for</code> - Schleife mit Anfangswert $a$ und Endwert $e$ sowie der optionalen Stufigkeit $s$ . Anstelle dessen sind auch Konstrukte der Form $i = [x_1 x_2 .. x_n]$ zulässig. Schleife wird $n$ -mal durchlaufen, mit den Indizes $x_1 x_2 .. x_n$
<code>if A1;</code> statements1 <code>elseif A2</code> statements2 <code>else</code> statements3 <code>end;</code>	<code>if</code> Konstrukt, verzweigt den Programmablauf. Ausdrücke (Bedingungen) $A1$ und $A2$ haben oftmals die Form: $a > b$ , $a == b$ , $a <= b$ .... Die <code>else</code> und <code>elseif</code> - Anweisungen sind optional. Siehe hierzu Bild 1

---

## Funktionen-Header

---

Abbildung 1: *if*-Verzweigungen

Im Programm erfolgt dann der Aufruf:

```
[out1,...,outN] = <Name> (par1,...,parM)
```