

Matlab Kurzreferenz

Sommer 12

Vektoren und Matrizen (help matfun, help ops, help arith)

a=[1 2 3], b=[4, 5, 6]	Definition eines Zeilenvektors
c=[7; 8; 9]	Definition eines Spaltenvektors
d=1:3	erzeugt den Vektor [1 2 3]
d=1:0.2:3	erzeugt den Vektor [1 1.2 1.4 ... 3]
zeros(1,3)	erzeugt den Vektor [0 0 0]
ones(1,5)	erzeugt den Vektor [1 1 1 1 1]
d(12)	12. Element des Vektors d
length(d)	Länge des Vektors
[X,Y]=size(d)	X: Anzahl der Zeilen, Y: Anzahl der Spalten
a'	a^T transponieren (adjungieren bei komplexen Vektoren/Matrizen)
a*b	nicht definiert $(1 \times n)(1 \times n)$
a*b'	Skalarprodukt $(1 \times n)(n \times 1)$
a'*b	dyadisches Produkt $(n \times 1)(1 \times n)$
a.*b	elementweise Multiplikation
A=[1 2 3;4 5 6]	Definition einer Matrix. Zeilen durch Semicolon od. Zeilenumbruch trennen
A=[a; b; c']	Definition einer Matrix mittel Vektoren (zeilenweise)
A(1,:)	erste Zeile
A(:,3)	dritte Spalte
A(2,2:3)	2. und 3. Element der 2. Spalte
2*A	Multiplikation mit Skalar
A'*A	Matrizenmultiplikation $A^T A$
A.*A	Elementweise Multiplikation
rank(A)	Rang
det(A)	Determinante
adj(A)	Adjunkte
A^-1	inverse Matrix
inv(A)	inverse Matrix
expm(A)	Matrixexponentialfunktion e^A
exp(A)	e^x wird elementweise ausgeführt
poly(A)	charakteristisches Polynom (liefert die Koeffizienten als Vektor)
eig(A)	Eigenwerte
[V,D]=eig(A)	V: Rechtseigenvektoren, D: Eigenwerte in Diagonalmatrix

Polynome (help polyfun)

In Matlab werde Polynome als Vektoren ihrer Koeffizienten dargestellt.

Beispiel: $p(x) = x^4 + 2x^2 - 4x + 5$ wird durch den Vektor $p=[1 \ 0 \ 2 \ -4 \ 5]$ repräsentiert.

<code>roots([1 0 2 -4 5])</code>	Wurzeln (Nullstellen) eines Polynoms
<code>p=poly([1 0 2])</code>	bestimmt Koeffizienten des Polynoms mit den angegebenen Wurzeln
<code>polyval(p,x)</code>	Auswertung des Polynoms p an der Stelle x
<code>polyadd(p1,p2)</code>	Addition der Polynome $p1$ und $p2$
<code>conv(p1,p2)</code>	Multiplikation der Polynome $p1$ und $p2$
<code>deconv(p1,p2)</code>	Division der Polynome $p1$ und $p2$
<code>residue(Z,N)</code>	Partialbruchzerlegung

Graphikausgabe (help graph2d)

<code>handle=figure</code>	erzeugt ein neues Plotfenster und liefert dessen <code>handle</code> (optional)
<code>plot(x,y)</code>	zeichnet Vektor y über Vektor x ; beide Vektoren müssen gleiche Länge haben
<code>semilogx(x,y)</code>	zeichnet Vektor y über Vektor x bei logarithmischer x -Achse
<code>semilogy(x,y)</code>	zeichnet Vektor y über Vektor x bei logarithmischer y -Achse
<code>hold on</code>	behält alle Plots im Graphikfenster Achtung: sonst werden alte Plots überschrieben!
<code>grid on</code>	fügt Gitterlinien in das Plotfenster ein
<code>xlabel('x-Achse')</code>	fügt der x -Achse eine Achsenbezeichnung hinzu
<code>ylabel('y-Achse')</code>	fügt der y -Achse eine Achsenbezeichnung hinzu
<code>title('Titel')</code>	fügt dem Diagramm einen Titel hinzu
<code>axis</code>	Einstellungen zur Achsenskalierung (alle)
<code>xlim</code>	x -Achsenskalierung
<code>ylim</code>	y -Achsenskalierung
<code>zoom(Faktor)</code>	zoomt den aktuellen Graphen um einen Faktor auf
<code>zoom on</code>	aktiviert zoom-Modus in aktuellem Graphen
<code>zoom xon</code>	aktiviert zoom-Modus in x -Richtung
<code>zoom yon</code>	aktiviert zoom-Modus in y -Richtung
<code>print -dpng Bildname</code>	speichert das aktuelle Plotfenster im png-Format als Bild ab
<code>print -depsc Bildname</code>	speichert das aktuelle Plotfenster im eps-Format - bunt

Control System Toolbox (help control)

<code>G=tf(p1,p2)</code>	Übertragungsfunktion mit Zählerpolynom $p1$ und Nennerpolynom $p2$
<code>G=minreal(G)</code>	Kürzen stabiler Null- und Polstellen aus einer Übertragungsfunktion
<code>pole(G)</code>	Polstellen einer Übertragungsfunktion
<code>zero(G)</code>	Nullstellen einer Übertragungsfunktion
<code>damp(G)</code>	Dämpfung und Frequenz von Nullstellen einer Übertragungsfunktion
<code>dcgain(G)</code>	stationäre Verstärkung einer Übertragungsfunktion
<code>pzmap(G)</code>	Plot des Pol- Nullstellendiagramms einer Übertragungsfunktion
<code>sgrid, zgrid</code>	Gitterlinien für s -Ebene oder z -Ebene
<code>bode(G)</code>	Plot des Bodediagramms einer Übertragungsfunktion
<code>step(G)</code>	Plot der Sprungantwort einer Übertragungsfunktion
<code>nyquist(G)</code>	Plot des Nyquistdiagramms einer Übertragungsfunktion
<code>rlocus(G)</code>	Plot der Wurzelortskurve einer Übertragungsfunktion
<code>G.inputdelay=T</code>	Totzeit einer Übertragungsfunktion auf den Wert T [s] setzen
<code>sys=ss(A,B,C,D)</code>	Erstellen eines zeitkontinuierlichen Zustandsraummodell mit den Systemmatrizen (A, B, C, D)
<code>sysD=c2d(sys, Ta)</code>	Umwandlung eines zeitkontinuierlichen in ein zeitdiskretes Zustandsraummodell mit Abtastzeit Ta [s]
<code>[A,B,C,D]=ssdata(sys)</code>	liefert die Systemmatrizen (A, B, C, D) eines Zustandsraummodells
<code>lsim(sys,u,t)</code>	Plot einer Systemantwort auf den Eingang $u(t)$

Definition einer Funktion (help function)

```
% function [c]=pythagoras(a,b) % pythagoras.m
%
% berechnet c aus der Formel von Pythagoras
%
% a : Matrix von Werte der Kathete a
% b : Matrix von Werte der Kathete b

function [c]=pythagoras(a,b)

c = sqrt(a.^2+b.^2)
```