A Mathematik

A.1 Matrizen

A.1.1 Inverse Matrix

Für eine Matrix $A \in \mathbb{R}^{n \times n}$ mit $\det(A) \neq 0$ ist inverse Matrix A^{-1} so definiert, dass

$$A \cdot A^{-1} = \mathbb{I}. \tag{1}$$

Im Allgemeinen lässt sich die Inverse mit der Formel

$$A^{-1} = \frac{1}{\det(A)} \cdot \mathrm{adj}(A), \qquad \{\mathrm{adj}(A)\}_{ij} = (-1)^{i+j} \cdot \det(A_{ij}) \tag{2}$$

berechnen. Spezialfälle:

- $n = 2$:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \Rightarrow \quad A^{-1} = \frac{1}{a \cdot d - b \cdot c} \cdot \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \tag{3}$$

- $n = 3$:

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad \Rightarrow \quad A^{-1} = \frac{1}{\det(A)} \begin{bmatrix} e \cdot i - f \cdot h & c \cdot h - b \cdot i & b \cdot f - c \cdot e \\ f \cdot g - d \cdot i & a \cdot i - c \cdot g & c \cdot d - a \cdot f \\ d \cdot h - e \cdot g & b \cdot g - a \cdot h & a \cdot e - b \cdot d \end{bmatrix} \tag{4}$$

A.1.2 Eigenwertproblem

Die Eigenwerte einer Matrix $A \in \mathbb{R}^{n \times n}$ sind die Lösung der Gleichung

$$\det(\lambda \cdot \mathbb{I} - A) = 0. \tag{5}$$

Der Eigenraum $\vec{v}_\lambda$ eines Eigenwertes λ ist dann gegeben durch

$$\mathrm{kern}(\lambda \cdot \mathbb{I} - A) \quad \Leftrightarrow \quad (\lambda \cdot \mathbb{I} - A) \cdot \vec{v} = 0. \tag{6}$$

A.2 Komplexe Analysis

Für eine komplexe Zahl

$$z = a + j \cdot b = r \cdot (\cos(\varphi) + j \cdot \sin(\varphi)) = r \cdot e^{j \cdot \varphi} \tag{7}$$

gilt

$$|z| = r = \sqrt{a^2 + b^2}, \qquad \angle(z) = \varphi = \begin{cases} \arctan\left(\frac{b}{a}\right) & \text{für } a > 0, \\ \arctan\left(\frac{b}{a}\right) + \pi & \text{für } a < 0. \end{cases} \tag{8}$$

A.2.1 Rechenregel

Für drei komplexe Zahlen z_1, z_2, z_3 sei

$$z = \frac{z_1 \cdot z_2}{z_3}. \tag{9}$$

Dann gilt:

$$|z| = \frac{|z_1| \cdot |z_2|}{|z_3|}, \qquad \angle(z) = \angle(z_1) + \angle(z_2) - \angle(z_3). \tag{10}$$

B Mechanik

B.1 Dynamik und Kräfte

Der Impulssatz ("Linear Momentum Principle") besagt:

$$m \cdot \ddot{\vec{\mathrm{OS}}} = \sum \vec{F}. \tag{11}$$

Der Spin-/Drallsatz ("Angular Momentum Principle") besagt:

$$I_{\mathrm{S}} \cdot \ddot{\theta} + m \cdot r_{\vec{\mathrm{OS}}} \times \ddot{\vec{\mathrm{OS}}} = \sum r_{\vec{\mathrm{OP}}} \times \vec{F_{\mathrm{P}}} + \sum \vec{M}, \tag{12}$$

$$I_{\mathrm{S}} \cdot \ddot{\theta} = \sum r_{\vec{\mathrm{SP}}} \times \vec{F_{\mathrm{P}}} + \sum \vec{M}. \tag{13}$$

Die Federkraft einer Feder mit Federkonstante k ist gegeben durch

$$F = k \cdot x. \tag{14}$$

Die Dämpferkraft eines Dämpfers mit Dämpferkonstante d ist gegeben durch

$$F = d \cdot \dot{x}. \tag{15}$$

B.2 Arbeit und Leistung

Die Leistung einer Kraft ist:

$$P = F \cdot v. \tag{16}$$

Die Leistung einer Drehmoment ist:

$$P = M \cdot \omega. \tag{17}$$

Bemerkung. Hier wird es angenommen, dass die Kraft und Geschwindigkeit bzw. Moment und Winkelgeschwindigkeit die gleiche Richtung besitzen.

C MATLAB

C.1 Allgemein

Befehl	Beschreibung
`A(i,j)`	Eintrag von `A` in Position `i` (Zeile) und `j` (Spalte)
`abs(X)`	Betrag von allen Einträgen von `X`
`angle(X)`	Phase von allen Einträgen von `X` (in Bogenmass)
`X'`	Komplex konjugiert und transponiert von `X`
`X.'`	Nicht Komplex konjugiert und transponiert von `X`
`conj(X)`	Komplex konjugiert von allen Einträge von `X`
`real(X)`	Realteil von allen Einträge von `X`
`imag(X)`	Imaginärteil von allen Einträge von `X`
`eig(A)`	Eigenwerte von `A`
`[V,D]=eig(A)`	Eigenwerte `D` (Diagonaleinträge), Eigenvektoren `V` (Spaltenvektoren)
`s=svd(A)`	Singularwerte der Matrix `A`
`[U,Sigma,V]=svd(A)`	Singular Values Decomposition der Matrix `A`
`rank(A)`	Rang der Matrix `A`
`det(A)`	Determinante der Matrix `A`
`inv(A)`	Inverse der Matrix `A`
`diag([a1,...,an])`	Diagonalmatrix mit `a1,...,an` als Diagonaleinträge
`zeros(x,y)`	Nullmatrix der Dimension `x`×`y`
`zeros(x)`	Nullmatrix der Dimension `x`×`x`
`eye(x,y)`	Identitätsmatrix der Dimension `x`×`y`
`eye(x)`	Identitätsmatrix der Dimension `x`×`x`
`ones(x,y)`	One-Matrix (alle Einträge = 1) der Dimension `x`×`y`
`ones(x)`	One-Matrix (alle Einträge = 1) der Dimension `x`×`x`
`max(A)`	Grösstes Element im Vektor `A` (`A` Matrix: Max in Spaltenvektoren)
`min(A)`	Kleinstes Element im Vektor `A` (`A` Matrix: Max in Spaltenvektoren)
`sum(A)`	Summe der Elemente von `A` (`A` Matrix: Summe Zeile pro Zeile)
`dim=size(A)`	Dimension der Matrix `A` (`size=[#Zeilen #Spalten]`)
`dim=size(A,a)`	`a=1`: `dim=#Zeilen`, `a=2`: `dim=#Spalten`, sonst `dim=1`
`t=a:i:b`	`t=[a,a+i,a+2i,...,b-i,b]` (Zeilenvektor)
`y=linspace(a,b)`	Zeilenvektor mit 100 "linear-spaced" Punkte im Intervall `[a,b]`
`y=linspace(a,b,n)`	Zeilenvektor mit `n` "linear-spaced" Punkte im Intervall `[a,b]`
`y=logspace(a,b)`	Zeilenvektor mit 50 "logarithmically-spaced" Punkte im Intervall `[10^a,10^b]`
`y=logspace(a,b,n)`	Zeilenvektor mit `n` "logarithmically-spaced" Punkte im Intervall `[10^a,10^b]`
`I=find(A)`	`I`: Indizen von den nichtnull Elemente von `A`
`disp(A)`	Print on screen von `A` (String: `'name'`)

C.2 RT-Befehle

Befehl	Beschreibung
`sys=ss(A,B,C,D)`	State-Space M. mit `A,B,C,D` im Zeitbereich
`sys=ss(A,B,C,D,Ts)`	State-Space M. mit `A,B,C,D` und Sampling Zeit `Ts` (zeitdiskret)
`sys=zpk(Z,P,K)`	State-Space M. mit Nullstellen `Z`, Pole `P` und Gain `K`
`sys=zpk(Z,P,K,Ts)`	State-Space M. mit Nullstellen `Z`, Pole `P`, Gain `K` und Sampling Zeit `Ts`
`sys=tf([bm ...b0],[an ...a0])`	Übertragungsfunktion mit `bn` in Zähler und `an` in Nenner
`P=tf(sys)`	Übertragungsfunktion von `sys`
`P.iodelay=...`	Fügt der Funktion `P` eine Totzeit hinzu.
`pole(sys)`	Pole eines Systems
`zero(sys)`	NST eines Systems
`[z,p,k]=zpkdata(sys)`	`z`: Nullstellen, `p`: Pole, `k`: statische Verstärkung
`ctrb(sys)` oder `ctrb(A,b)`	Steuerbarkeitsmatrix
`obsv(sys)` oder `obsv(A,c)`	Beobachtbarkeitsmatrix
`series(sys1,sys2)`	Serieschaltung von `sys1` und `sys2`
`feedback(sys1,sys2)`	`sys1` mit `sys2` als (negative) Feedback
`[Gm,Pm,Wgm,Wpm]=margin(sys)`	`Gm`: Verstärkungsreserve, `Pm`: Phasenreserve, `Wpm`: Durchtrittsfrequenz
`[y,t]=step(sys,Tend)`	`y`: Sprungantwort von `sys` bis `T`, `t`: Zeit
`[y,t]=impulse(sys,Tend)`	`y`: Impulsantwort von `sys` bis `Tend`, `t`: Zeit
`y=lsim(sys,u,t)`	Simulation von `sys` mit dem Input `u` für die Zeit `t`
`sim('Simulink model',Tend)`	Simultion von `Simulink Model'` bis `Tend`
`p0=dcgain(sys)`	Statische Verstärkung ($P(0)$)
`K=lqr(A,B,Q,R)`	Verstärkungsmatrix `K` (Lösung des LQR-Problems)
`[X,L,K]=care(A,B,Q)`	`X`: Lösung der Riccati Gleichung, `G`: Verstärkungsmatrix
`Paug=augw(G,W1,W3,W2)`	Space State M. für $\mathcal{H}_\infty$
`[K,Cl,gamma]=hinfsyn(Paug)`	$\mathcal{H}_\infty$: `K`: Regler
`fr=evalfr(sys,f)`	`sys` in `f` evaluiert ($s = f$)
`sysd=c2d(sys,Ts,method)`	Disktretisierung von `sys` nach `method` mit Sampling Zeit `Ts`

C.3 Plot und Diagramme

Befehl	Beschreibung
`nyquist(sys)`	Nyquist-Diagramm des Systems `sys`
`nyquist(sys,{a,b})`	Nyquist-Diagramm im Intervall `[a,b]` des Systems `sys`
`bode(sys)`	Bode-Diagramm des Systems `sys`
`bode(sys,{a,b})`	Bode-Diagramm im Intervall `[a,b]` des Systems `sys`
`bodemag(sys)`	Bode-Diagramm (nur Betrag) des Systems `sys`
`bodemag(sys,{a,b})`	Bode-Diagramm (nur Betrag) im Intervall `[a,b]` des Systems `sys`
`rlocus(sys)`	Wurzelortskurven-Diagramm
`impulse(sys)`	Impulsantwort des Systems `sys`
`step(sys)`	Sprungantwort des Systems `sys`
`pzmap(sys)`	Pole-Nullstelle Map des Systems `sys`
`svd(sys)`	Singularwertverlauf des Systems `sys`
`plot(X,Y)`	Plot von `Y` als Funktion von `X`
`plot(X,Y,...,Xn,Yn)`	Plot von `Yn` als Funktion von `Xn` (für alle `n`)
`stem(X,Y)`	Diskreter Plot von `Y` als Funktion von `X`
`stem(X,Y,...,Xn,Yn)`	Diskreter Plot von `Yn` als Funktion von `Xn` (für alle `n`)
`xlabel('name')`	Name der `x`-Achse
`ylabel('name')`	Name der `y`-Achse
`title('name')`	Titel des Plots
`xlim([a b])`	Schranke für die `x`-Achse (Plot zwischen `a` und `b`)
`ylim([a b])`	Schranke für die `y`-Achse (Plot zwischen `a` und `b`)
`grid on`	Grid
`title('name')`	Titel des Plots
`legend('name1',...,'namen')`	Legende
`subplot(m,n,p)`	Grid `m`×`n`, Plot in Position `p`
`semilogx(X,Y)`	Logarithmitic Plot mit `y`-Achse linear