

# Matlab - CheatSheet

©Jan Brupbacher

12. April 2020

## Inhaltsverzeichnis

1	Basics. . . . .	2
2	Einige kleine Kniffs . . . . .	2
3	Least Square implementation . . . . .	2
4	Transfer function . . . . .	2
5	Gleichung (System) symbolisch lösen . . . . .	2
6	Bode and Nyquist plot . . . . .	3
7	State-Space to Transfer function. . . . .	3

# 1 Basics

```

1 v = [1 2 3]           % Zeilenvektor
2 w = [1; 2; 3]         % Spaltenvektor
3 A = [1 2; 3 4]        % 2x2 Matrix
4 B = eye(n)            % generiert eine n*n Einheitsmatrix

```

## 2 Einige kleine Kniffs

```

1 A = B'                % ' bedeutet transponiert
2 v = u.*u              % . bedeutet Elementweise Operation -> v ist vektor
3 w = u'*u              % w ist skalar (Skalarprodukt von u und u)

```

## 3 Least Square implementation

```

1 Phi = [1 2 3]'        % transponiert damit Spaltenvektor!
2 y = [0.5 1.9 3.2]'
3 theta = inv(Phi'*Phi)*Phi'*y % 'Naiv'
4 theta = pinv(Phi)*y    % Weniger Operationen
5 theta = Phi\y          % Ohne Inversion (beste Variante und
6                        % numerisch robust)

```

## 4 Transfer function

```

1 sys1 = tf(2*[1 1], poly([0 -2])) % sys1 = 2*(s+1)/((s+1)*s)
2 sys2 = tf(poly([-1 -2]), poly([0 -1])) % poly macht aus Nullstellen ein
3                                     % Polynom
4 sys = minreal(feedback(sys1, sys2, -1)) % Feedbackschaltung mit feedback
5 stability = isstable(sys)             % minreal => loescht kritische
6                                     % Polstellen ???
7 step(sys1)                           % Sprungantwort des Systems
8 step(sys1, 2)                        % Sprungantwort der ersten 2 sek
9 zpke(sys1)                           % Faktorisieren der Polynome
10                                     % (umkehrung von tf())

```

## 5 Gleichung (System) symbolisch lösen

```

1 syms s                % s ist eine symbolische Variable
2
3 b = 1/(s+1);
4 c = 1/s;
5 d = 1/(s+1);
6 e = 2;
7 f = 2;
8 g = -26;
9 h = -12;
10 i = 1/(s^2+3*s+2);
11
12 sys = (b+c+d+e)/(1-(b*f + b*c*g + b*c*d*h + c*d*e*i));
13 simplify(sys)         % simplify => Gleichung vereinfachen
14 pretty(simplify(sys)) % pretty => schoene Darstellung von Bruechen und
15                       % Exponenten

```

## 6 Bode and Nyquist plot

```
1 G = tf(2, [1 0]) * tf((1, [0.5 1])^2
2 nyquist(G)
3 bode(G)
4 grid
5 margin(G) % Grafische Anzeigen von Gm und Pm
6 [Gm, Pm, Wcg, Wcp] = margin(G) % direktes abspeichern der Werte
7 % Gm: Gain margin ; Pm: Phase margin
8 % Wcg: omega of Gm ; Wcp: omega of Pm
```

## 7 State-Space to Transfer function

```
1 [b, a] = ss2tf(A, B, C, D) % [b, a] ist input von tf()
2 % sprich b und a sind Vektoren
3 % A: n*n Matrix
4 % B: n*m Matrix
5 % C: k*n Matrix
6 % D: k*m Matrix
7 [A, B, C, D] = tf2ss(b, a) % ACHTUNG: kann in diese Richtung mehrere
8 % Ergebnisse haben
```