Versuch II: Linearisierung, Steuerbarkeit und Beobachtbarkeit

Andreas Jentsch, Ali Kerem Sacakli

Praktikumsbericht – Praktikum Matlab/Simulink II

TECHNISCHE UNIVERSITÄT DARMSTADT

REGELUNGSTECHNIK UND MECHATRONIK rtm

2.1 Lienarisierung

Im folgenden Abschnitt wird die Funktion zur Linearisierung des Doppelpendel-Systems um einen Arbeitspunkt, sowie ihre Rückgabewerte an bestimmten Arbeitspunkten dokumentiert. Die Implementierung der Funktion ist in Listing 2.1 aufgeführt.

Bevor das System linearisiert wird sind zwei Fragen zu klären:

1. Welche Arbeitspunkte sind sinnvoll?
2. Was bedeutet es physikalisch, wenn M_{AP} ungleich null ist?

Die Antworten lauten wie folgt:

1. Es ist nur sinnvoll das System in Arbeitspunkten zu linearisieren, in denen es sowohl vollständig beobachtbar, als auch steuerbar ist.
2. Bei der Größe M_{AP} handelt es sich um den statischen Wert der Stellgröße M im Arbeitspunkt. Ist diese ungleich null muss der Motor das Moment M

Listing 2.1: Code der Linearisierungsfunktion

```
function [ A, B, C, D] = linearisierung( f, h, AP )

syms phi1 phi2 dphi1 dphi2 ddphi1 ddphi2 M;

x = [phi1;dphi1;phi2;dphi2];
u = M;

f_M_AP = subs(f(2),x,AP);

M_AP = solve(f_M_AP == 0 , M);

A = jacobian(f,x);
B = jacobian(f,u);
C = jacobian(h,x);
D = jacobian(h,u);

A = subs(A,[x,u],[AP,M_AP]);
B = subs(B,[x,u],[AP,M_AP]);
C = subs(C,[x,u],[AP,M_AP]);
D = subs(D,[x,u],[AP,M_AP]);


A = double(A);
```

```
B = double(B);
C = double(C);
D = double(D);

end
```

Die Linearisierung um die Arbeitspunkte

$$\mathbf{x}_{\mathrm{AP}_1} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$$
$$\mathbf{x}_{\mathrm{AP}_2} = \begin{bmatrix} \pi & 0 & \pi & 0 \end{bmatrix}$$
$$\mathbf{x}_{\mathrm{AP}_3} = \begin{bmatrix} \pi/2 & 0 & \pi & 0 \end{bmatrix}$$

ergibt für die allgmeine Zustandsraumdarstellung:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$
$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$$

die folgenden Systemmatrizen:

$$\mathbf{A}_{\mathrm{AP1}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -126{,}1286 & 0 & 63{,}0643 & 0 \\ 0 & 0 & 0 & 1 \\ 189{,}1929 & 0 & 168{,}1714 & 0 \end{bmatrix} \qquad \mathbf{B}_{\mathrm{AP1}} = \begin{bmatrix} 0 \\ 142{,}8571 \\ 0 \\ -214{,}2857 \end{bmatrix}$$
$$\mathbf{C}_{\mathrm{AP1}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{D}_{\mathrm{AP1}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathbf{A}_{\mathrm{AP2}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 126{,}1286 & 0 & -63{,}0643 & 0 \\ 0 & 0 & 0 & 1 \\ -189{,}1929 & 0 & 168{,}1714 & 0 \end{bmatrix} \qquad \mathbf{B}_{\mathrm{AP2}} = \begin{bmatrix} 0 \\ 142{,}8571 \\ 0 \\ -214{,}2857 \end{bmatrix}$$
$$\mathbf{C}_{\mathrm{AP2}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{D}_{\mathrm{AP2}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathbf{A}_{\mathrm{AP3}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 73{,}5750 & 0 \end{bmatrix} \qquad \mathbf{B}_{\mathrm{AP3}} = \begin{bmatrix} 0 \\ 62{,}5 \\ 0 \\ 0 \end{bmatrix}$$
$$\mathbf{C}_{\mathrm{AP3}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{D}_{\mathrm{AP3}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

2.2 Vergleich der Linearisierten Modelle

Die Eigenwerte der Zustandsraummodelle um die drei Arbeitspunkte lauten wie folgt:

$$\lambda_1 = \begin{bmatrix} 16{,}0744i \\ -16{,}0744i \\ 5{,}9929i \\ -5{,}9929i \end{bmatrix}$$

$$\lambda_2 = \begin{bmatrix} -16{,}0744 \\ 16{,}0744 \\ -5{,}9929 \\ 5{,}9929 \end{bmatrix}$$

$$\lambda_3 = \begin{bmatrix} 8{,}5776 \\ -8{,}5776 \\ 0 \\ 0 \end{bmatrix}$$

System 1 ist ungedämpft schwingungsfähig und grenzstabil. System 2 ist instabil. System 3 ist instabil.

2.3 Normalformen des Zustandsraummodelles

Listing 2.2: Code der Diagonalisierungsfunktion

```
function [ AD, BD, CD, DD ] = diagonalForm( A, B, C, D )

[V, Deig] = eig(A);

if rank(V)==length(V)
    V_inv = inv(V);

    AD = Deig;
    BD = V_inv*B;
    CD = C*V;
    DD = D;
else
    disp('Matrix A ist nicht diagonalähnlich!')
end

end
```

Die Transformation des um AP2 linearisierten Systems in die Diagonalform mit der Funktion in Listing 2.2 ergibt folgende Systemmatritzen:

$$\mathbf{A}_\mathrm{D} = \begin{bmatrix} -16{,}0744 & 0 & 0 & 0 \\ 0 & 16{,}0744 & 0 & 0 \\ 0 & 0 & -5{,}9929 & 0 \\ 0 & 0 & & 5{,}9929 \end{bmatrix} \qquad \mathbf{B}_\mathrm{D} = \begin{bmatrix} 138{,}1293 \\ 138{,}1293 \\ -21{,}3960 \\ 21{,}3960 \end{bmatrix}$$

$$\mathbf{C}_\mathrm{D} = \begin{bmatrix} -0{,}0267 & 0{,}0267 & 0{,}0943 & 0{,}0943 \\ 0{,}0560 & -0{,}0560 & 0{,}1349 & 0{,}1349 \end{bmatrix} \qquad \mathbf{D}_\mathrm{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Die Transormation des selben Systems mit der MATLAB Funktion canon ergibt folgende Systemmatritzen:

$$\mathbf{A}_\mathrm{D} = \begin{bmatrix} -16{,}0744 & 0 & 0 & 0 \\ 0 & -5{,}9929 & 0 & 0 \\ 0 & 0 & 16{,}0744 & 0 \\ 0 & 0 & & 5{,}9929 \end{bmatrix} \qquad \mathbf{B}_\mathrm{D} = \begin{bmatrix} 2{,}5354 \\ -0{,}7024 \\ 2{,}5354 \\ 0{,}7024 \end{bmatrix}$$

$$\mathbf{C}_\mathrm{D} = \begin{bmatrix} -1{,}4559 & 2{,}8726 & 1{,}4559 & 2{,}8726 \\ 3{,}0534 & 4{,}1093 & -3{,}0534 & 4{,}1093 \end{bmatrix} \qquad \mathbf{D}_\mathrm{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Die beiden Transformierten Systeme unterscheiden sich in der Sortierung der Eigenwerte in der **A** matrix, sowie in den Werten der **B** und **C** Matrizen. Dies ist durch eine unterschiedliche Sortierung der

Die Transformation des um AP1 linearisierten Systems mit der MATLAB Funktion canon ergibt folgende Systemmmatritzen:

$$\mathbf{A}_\mathrm{D} = \begin{bmatrix} 0 & 16{,}0744 & 0 & 0 \\ -16{,}0744 & 0 & 0 & 0 \\ 0 & 0 & & 5{,}9929 \\ 0 & 0 & -5{,}9929 & 0 \end{bmatrix} \qquad \mathbf{B}_\mathrm{D} = \begin{bmatrix} 5{,}3736 \\ 0 \\ 0 \\ -2{,}1428 \end{bmatrix}$$

$$\mathbf{C}_\mathrm{D} = \begin{bmatrix} 0 & -1{,}3739 & -1{,}8832 & 0 \\ 0 & 2{,}8813 & -2{,}6939 & 0 \end{bmatrix} \qquad \mathbf{D}_\mathrm{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

2.4 Untersuchung von Steuerbarkeit und Beobachtbarkeit

Listing 2.3: Code der Steuerbarkeitsfunktion nach Kalman

```
function [ S_s_Kalman] = checkCtrbKalman( A, B )

n = length(A);
```

```
if (rank(A)==n)

else
    disp('Matrix␣A␣hat␣nicht␣vollen␣Rang')
end
S_s_Kalman = B;
for count = 1:n-1
    S_s_Kalman = [S_s_Kalman, A^count *B ];
end
end
```

Listing 2.4: Code der Beobachtbarbarkeitsfunktion nach Kalman

```
function [ S_b_Kalman ] = checkObsvKalman( A, C )

n = length(A);

if rank(A) == n
else
    disp('Matrix␣A␣hat␣nicht␣vollen␣Rang');
end
S_b_Kalman = C;
for count = 1:n-1
    S_b_Kalman = [S_b_Kalman ; C * A^count];
end
end
```

Die MATLAB Funktionen ctrb und obsv ergeben die selben Steuerbar- beziehungsweise Beobachtbarkeitsmatritzen

$$\mathbf{S}_{SAP1} = 10000 \cdot \begin{bmatrix} 0 & 0{,}0143 & 0 & -3{,}1532 \\ 0{,}0143 & 0 & -3{,}1532 & 0 \\ 0 & -0{,}0214 & 0 & 6{,}3064 \\ -0{,}0214 & 0 & 6{,}3064 & 0 \end{bmatrix}$$

$$\mathbf{S}_{BAP1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -126{,}1286 & 0 & 63{,}0643 & 0 \\ 189{,}1929 & 0 & -168{,}1714 & 0 \\ 0 & -126{,}1286 & 0 & 63{,}0643 \\ 0 & 189{,}1929 & 0 & -168{,}1714 \end{bmatrix}$$

wie die selbst geschriebenen Funktionen `checkCtrbKalman` in Listing 2.3 und `checkObsvKalman` in Listing 2.4. Dies gilt für alle Arbeitspunkte

Listing 2.5: Code der Steuerbarkeitsfunktion nach Gilbert

```
function [ isCtrb_Gilbert ] = checkCtrbGilbert( A, B )

[T, lam] = eig(A);

if length(unique(diag(lam))) ~= length(A)
    disp('A hat mehrfache Eigenwerte');
end

Bd = T/B;
isCtrb_Gilbert = all(any(Bd,2));

if isCtrb_Gilbert
    disp('System ist steuerbar nach Gilbert');
else
    disp('System ist NICHT steuerbar nach Gilbert');
end
end
```

Listing 2.6: Code der Beobachtbarbarkeitsfunktion nach Gilbert

```
function [ isObsv_Gilbert ] = checkObsvGilbert( A, C )
```

```

[T, lam] = eig(A);

if length(unique(diag(lam))) ~= length(A)
    disp('A␣hat␣mehrfache␣Eigenwerte');
end

Cd = C*T;
isObsv_Gilbert = all(any(Cd,1));

if isObsv_Gilbert
    disp('System␣ist␣beobachtbar␣nach␣Gilbert');
else
    disp('System␣ist␣NICHT␣beobachtbar␣nach␣Gilbert');
end
end
```

Listing 2.7: Code der Steuerbarkeitsfunktion nach Hautus

```
function [ isCtrb_Hautus ] = checkCtrbHautus( A,B )

lam = eig(A);
isCtrb_Hautus = 0;
for count = 1:length(lam)
    ranks(count) = rank(horzcat(eye(length(A))*lam(count) - A, B));
end

if all(ranks == length(A))
    disp('System␣ist␣steuerbar␣nach␣Hautus');
    isCtrb_Hautus = 1;
else
    disp('System␣ist␣NICHT␣steuerbar␣nach␣Hautus');
end

end
```

Listing 2.8: Code der Beobachtbarbarkeitsfunktion nach Hautus

```
function [ isObsv_Hautus ] = checkObsvHautus( A,C )

lam = eig(A);
```

```
isObsv_Hautus = 0;
for count = 1:length(lam)
    ranks(count) = rank(vertcat(eye(length(A))*lam(count) - A, C));
end

if all(ranks == length(A))
    disp('System␣ist␣beobachtbar␣nach␣Hautus');
    isObsv_Hautus = 1;
else
    disp('System␣ist␣NICHT␣beobachtbar␣nach␣Hautus');
end

end
```