Fachgebiet
Simulation und Optimale Prozesse

TECHNISCHE UNIVERSITÄT
ILMENAU

Fakultät für Informatik und
Automatisierung
Institut für Automatisierungs-
und Systemtechnik

Praktikum

Versuch AS – G34

„Zustandsregelung – MATLAB®[1]“

Verantwortlicher Hochschullehrer: Prof. Dr.–Ing. habil. P. Li
Versuchsverantwortlicher: Dr.–Ing. S. Hopfgarten

Name,Vorname	**Matrikel-Nr.**
Mitarbeiter	
Note, Unterschrift	

[1]MATLAB® ist eingetragenes Warenzeichen der The MathWorks, Inc.

1 Ziel

In der regelungstechnischen Ausbildung wird neben der ein- und mehrschleifigen Ausgangsregelung die Zustandsregelung vorgestellt. Ausgehend von der Zustandsbeschreibung dynamischer Systeme werden Methoden zum Steuerungsentwurf angegeben, z.B. Polzuweisung (Polplatzierung, Polvorgabe, Eigenwertvorgabe), optimale Regelung (Riccati-Regler). Mit diesem Praktikum soll die Verhaltensweise von Zustandsregelkreisen untersucht werden. Zustandsregler sind für unterschiedliches Verhalten der Ausgangs- und Zustandsgrößen zu entwerfen sowie stationäres und dynamisches Verhalten zu beurteilen. Dies soll anhand von mathematischen Systemmodellen für praktisch relevante Aufgabenstellungen simulativ durchgeführt werden.

2 Realisierung des Praktikums

Das Praktikum wird unter Nutzung des Programmpacketes MATLAB® durchgeführt. Dazu sind Zustandsbeschreibungen für die entsprechenden Systeme zu erstellen, die Daten zur Systembeschreibung in MATLAB® zu hinterlegen, die entsprechenden Routinen zur Berechnung der Rückführmatrix zu benutzen, die Simulation auszuführen und die Ergebnisse grafisch darzustellen und zu diskutieren. Die relevanten Befehle sind im Anhang aufgelistet.

3 Vorbereitungsaufgaben

3.1 Aufstellen der Zustandsbeschreibung

Stellen Sie eine vollständige Zustandsbeschreibung für das abgebildete System zweier Rührkesselreaktoren auf! Als Systemzustände $x_1(t)$ und $x_2(t)$ werden die Konzentrationen $c_1(t)$ und $c_2(t)$,

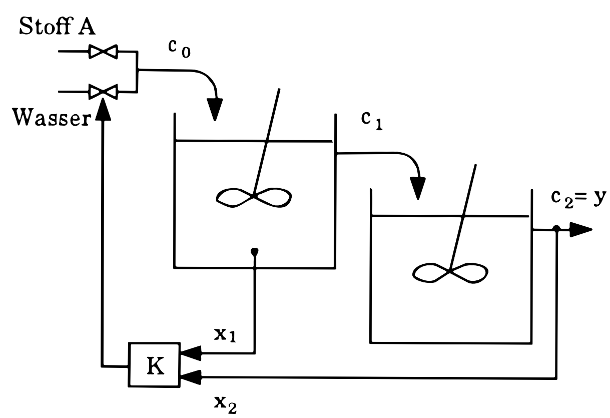

Abbildung 1: Geregelte Rührkesselreaktoren

als Steuergröße die Konzentration $c_0(t)$ gewählt. Als Ausgangsgröße $y(t)$ soll $c_2(t)$ betrachtet werden. Der Durchfluss $Q(t) = Q$ sei konstant. Nutzen Sie zur Herleitung der Zustandsbeschreibung die Massenbilanzgleichung. Folgende Werte sind gegeben: $Q = 2\,\frac{m^3}{min}$, $V_1 = 6\,m^3$, $V_2 = 2\,m^3$.

3.2 Voraussetzungen

Welche Voraussetzungen müssen erfüllt sein, damit das Zustandsregelungskonzept anwendbar ist?

3.3 Überprüfung der Voraussetzungen

Prüfen Sie anhand des gegebenen Beispieles, ob die Voraussetzungen erfüllt sind?

3.4 Entwurf einer Zustandsregelung

Welche Schritte sind beim Zustandsreglerentwurf erforderlich?
Entwerfen Sie eine Zustandsregelung, deren Ausgangsverhalten (zweiter Zustand) dem aperiodischen Grenzfall entspricht und im Falle der Führungsübergangsfunktion auch stationäre Genauigkeit aufweist! Wählen Sie ein Polpaar $s_{1/2} = -2$ (s = Laplacevariable).

4 Versuchsdurchführung

4.1 Geregelte Rührkesselreaktoren

4.1.1 Systembeschreibung

Benutzen Sie die in der Vorbereitungsaufgabe 3.1 aufgestellte Systembeschreibung und hinterlegen Sie sie in geeigneter Form in MATLAB®!

4.1.2 Voraussetzung

Prüfen Sie die Voraussetzung zur Anwendung der Zustandsregelung!

4.1.3 Verhalten der Regelstrecke

Illustrieren Sie sich das Verhalten der Regelstrecke!

4.1.4 Zustandsregler

a) Realisieren Sie den in der Vorbereitungsaufgabe 3.4 entworfenen Zustandsregler und untersuchen Sie das Verhalten des geschlossenen Kreises unter Nutzung der entsprechenden MATLAB®-Funktionen (siehe Anhang)! Entwerfen Sie weitere Einstellungen nach dem Verfahren der Polzuweisung und/oder unter Anwendung des Riccati-Reglers!

b) Entwerfen Sie einen Zustandsregler mit stationärer Genauigkeit beim Führungsverhalten und folgenden Pollagen: b1) $s_{1/2} = -0.5$, b2) $s_{1/2} = -5$, b3) $s_1 = -1,\ s_2 = -2$ und b4) $s_{1/2} = -2 \pm 3j$.

c) Diskutieren und bewerten Sie die Zustandsverläufe und die Steuergröße im Fall unter a) und in den 4 Fällen unter b) ! Vergleichen Sie das Verhalten mit dem der Regelstrecke!

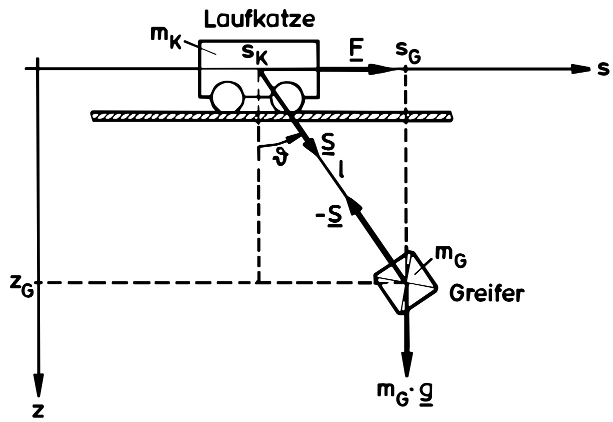

Abbildung 2: Verladebrücke

4.2 Verladebrücke

4.2.1 Systembeschreibung

Die in Abb. 2 schematisch dargestellte Verladebrücke (Laufkatze, Brückenkran) wird unter der Annahme vernachlässigbarer Reibung und ungedämpfter Schwingungen durch folgende Diffenzialgleichungen beschrieben:

$$\begin{aligned}(m_K + m_G)\,\ddot{s}_K + m_G l\ddot{\vartheta}\,cos\,\vartheta - m_G l\dot{\vartheta}^2\,sin\,\vartheta &= F(t)\\ \ddot{s}_K + l\ddot{\vartheta} + g\,sin\,\vartheta &= 0\end{aligned}$$

mit s_K - Katzposition, $\ddot{s}_K$ - Katzbeschleunigung, ϑ - Greiferwinkel, $\ddot{\vartheta}$ - Greiferwinkelbeschleunigung, m_K - Katzmasse, m_G - Greifermasse (einschließlich Last), l - Länge des Greiferarms, g - Erdbeschleunigung, F - Antriebskraft.
Da nur kleine Ausschläge des Greiferarms mit Last angenommen werden (ϑ sehr klein), gelten folgende Vereinfachungen: $sin\,\vartheta \approx \vartheta$, $cos\,\vartheta \approx 1$ und $\dot{\vartheta}^2\,sin\,\vartheta \approx 0$.
Legen Sie naheliegende Zustände fest, stellen Sie die vereinfachte Systembeschreibung auf und überführen Sie sie in die Zustandsbeschreibung in Matrizenschreibweise!
Hinterlegen Sie die Systembeschreibung in MATLAB® für folgende Systemparameter (alle Angaben in SI-Basiseinheiten): $m_K = 1000$, $m_G = 4000$, $l = 10$ und $g = 9.81$.

4.2.2 Voraussetzung

Prüfen Sie die Voraussetzung zur Anwendung der Zustandsregelung!

4.2.3 Verhalten der Regelstrecke

Illustrieren Sie sich das Verhalten der Regelstrecke!

4.2.4 Zustandsregler

Entwerfen und realisieren Sie Zustandsregler nach dem Verfahren der Polzuweisung und/oder unter Anwendung des Riccati-Reglers! Diskutieren Sie die Ergebnisse!

5 Anhang – Benötigte MATLAB®-Funktionen und Syntax

Zu allen Funktionen kann man sich eine ausführliche Hilfe verschaffen mit **doc** *funktionsname bzw.* **help** *funktionsname.*

```
Example of assignment of data to matrix/vector:
    A=[0 1;3 4]; (First semicolon separates lines; second semicolon
    suppresses output in command window.)

    A' is transpose of A

    A*B is matrix multiplication

ss  Create state-space model or convert LTI model to state space.

    sys = ss(A,B,C,D) creates a continuous-time state-space (SS) model
    sys with matrices A,B,C,D.

ctrb  Compute the controllability matrix.

    CO = ctrb(A,B) returns the controllability matrix [B AB A^2B ...].

    CO = ctrb(SYS) returns the controllability matrix of the
    state-space model SYS with realization (A,B,C,D).  This is
    equivalent to ctrb(sys.a,sys.b).

lsim  Simulate time response of LTI models to arbitrary inputs.

    LSIM(SYS,U,T) plots the time response of the LTI model SYS to the
    input signal described by U and T.  The time vector T consists of
    regularly spaced time samples and U is a matrix with as many columns
    as inputs and whose i-th row specifies the input value at time T(i).
    For example,
            t = 0:0.01:5;   u = sin(t);   lsim(sys,u,t)
    simulates the response of a single-input model SYS to the input
    u(t)=sin(t) during 5 seconds.

acker  Pole placement gain selection using Ackermann's formula.

    K = acker(A,B,P)  calculates the feedback gain matrix K such that
    the single input system
            .
            x = Ax + Bu

    with a feedback law of  u = -Kx  has closed loop poles at the
    values specified in vector P, i.e.,  P = eig(A-B*K).

place  Pole placement technique
```

```
    K = place(A,B,P)  computes a state-feedback matrix K such that
    the eigenvalues of  A-B*K  are those specified in vector P.
    No eigenvalue should have a multiplicity greater than the
    number of inputs.

LQR  Linear-quadratic regulator design for continuous-time systems.

    [K,S,E] = LQR(A,B,Q,R,N)  calculates the optimal gain matrix K
    such that the state-feedback law  u = -Kx  minimizes the cost
    function

        J = Integral {x'Qx + u'Ru + 2*x'Nu} dt
                                          .
    subject to the state dynamics  x = Ax + Bu.

    The matrix N is set to zero when omitted.  Also returned are the
    Riccati equation solution S and the closed-loop eigenvalues E:
                        -1
      SA + A'S - (SB+N)R  (B'S+N') + Q = 0 ,    E = EIG(A-B*K) .

eig    Eigenvalues and eigenvectors.
    E = eig(X) is a vector containing the eigenvalues of a square
    matrix X.

inv    Matrix inverse.
    inv(X) is the inverse of the square matrix X.

rank   Matrix rank.
    rank(A) provides an estimate of the number of linearly
    independent rows or columns of a matrix A.

step  Step response of LTI models.

    step(SYS) plots the step response of the LTI model SYS (created
    with either tf, zpk, or ss).

    [Y,T,X] = step(SYS) also returns, for a state-space model SYS, the
    state trajectory X, a LT-by-NX-by-NU array if SYS has NX states.

plot   Linear plot.
    plot(X,Y) plots vector Y versus vector X.

    Various line types, plot symbols and colors may be obtained with
    plot(X,Y,S) where S is a character string made from one element
    from any or all the following 3 columns:

           b     blue          .     point              -     solid
           g     green         o     circle             :     dotted
```

```
        r     red           x     x-mark            -.    dashdot
        c     cyan          +     plus              --    dashed
        m     magenta       *     star
        y     yellow        s     square
        k     black

augstate  Appends states to the outputs of a state-space model.

    ASYS = AUGSTATE(SYS)  appends the states to the outputs of
    the state-space model SYS.  The resulting model is:
       .                        .
       x  = A x + B u   (or  E x = A x + B u for descriptor SS)

      |y| = [C] x + [D] u
      |x|   [I]     [0]

    This command is useful to close the loop on a full-state
    feedback gain  u = Kx.  After preparing the plant with
    AUGSTATE,  you can use the FEEDBACK command to derive the
    closed-loop model.

feedback  Feedback connection of two LTI models.

    SYS = FEEDBACK(SYS1,SYS2) computes an LTI model SYS for
    the closed-loop feedback system

           u --->O---->[ SYS1 ]----+---> y
                 |                 |           y = SYS * u
                 +-----[ SYS2 ]<---+

    Negative feedback is assumed and the resulting system SYS
    maps u to y.  To apply positive feedback, use the syntax
    SYS = FEEDBACK(SYS1,SYS2,+1).

    SYS = FEEDBACK(SYS1,SYS2,FEEDIN,FEEDOUT,SIGN) builds the more
    general feedback interconnection:
                       +--------+
           v --------->|        |--------> z
                       |  SYS1  |
           u --->O---->|        |----+---> y
                 |     +--------+    |
                 |                   |
                 +-----[  SYS2  ]<---+

    The vector FEEDIN contains indices into the input vector of SYS1
    and specifies which inputs u are involved in the feedback loop.
    Similarly, FEEDOUT specifies which outputs y of SYS1 are used for
    feedback.  If SIGN=1 then positive feedback is used.  If SIGN=-1
```

```
or SIGN is omitted, then negative feedback is used.  In all cases,
the resulting LTI model SYS has the same inputs and outputs as SYS1
(with their order preserved).
```

Literatur

[1] {P. Li, J. Reger, C. Ament}.Vorlesung Regelungs- und Systemtechnik 2 + 3. TU Ilmenau

[2] Taschenbuch Elektrotechnik. *1. Auflage, Berlin 1977, Bd. 2; 3. Auflage, Berlin 1987 Bd. 1.*

[3] O. Föllinger. Regelungstechnik, Hüthig, 1992

[4] J. Lunze. Regelungstechnik 1. 2. Auflage, Springer, 1999

[5] J. Lunze. Regelungstechnik 2. 1. Auflage, Springer, 1997

[6] MATLAB® – The Language of Technical Computing, The MathWorks, Inc., Natick, Massachusetts, 2000

[7] MATLAB® – Control System Toolbox: For Use with MATLAB®, The MathWorks, Inc., Natick, Massachusetts, 2000