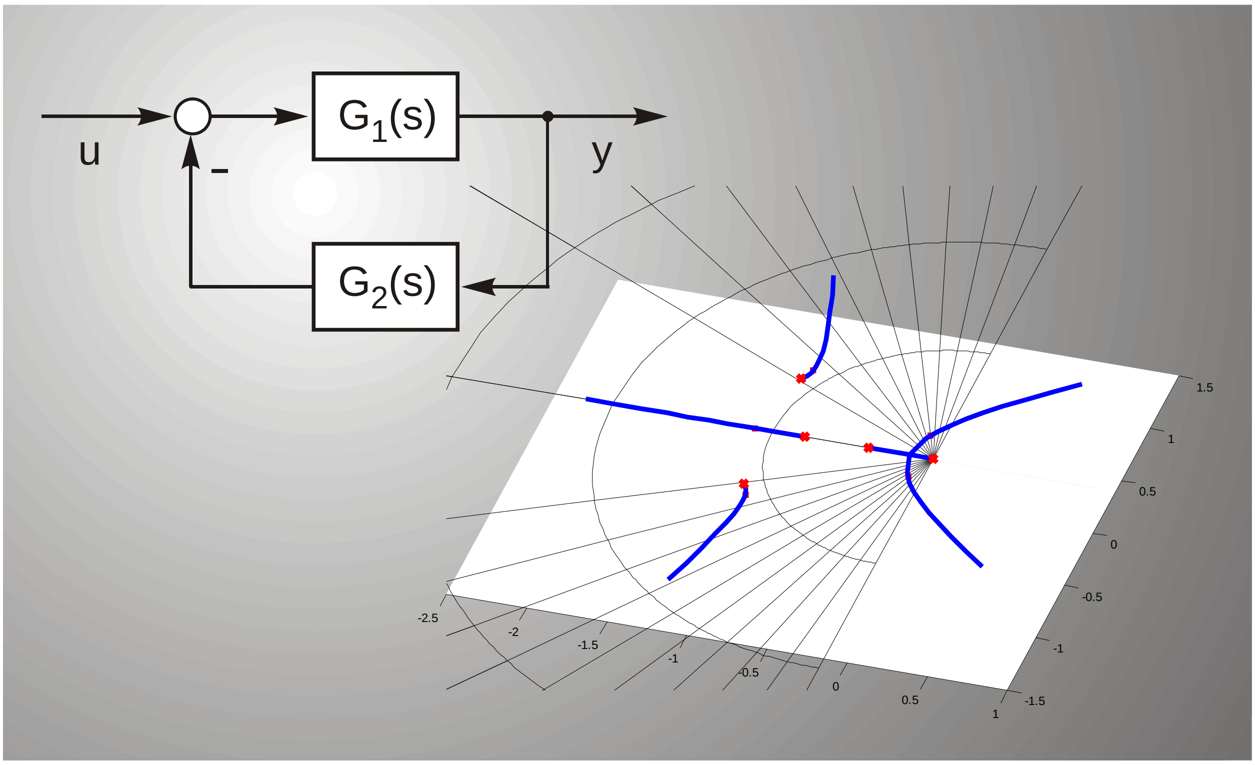
Aufgaben zur Übung 2

Systemdynamik und Regelungstechnik

M. Sc. Patrick Sauter



Sommersemester 2014

http://www.irs.kit.edu/1528.php

Karlsruher Institut für Technologie

Institut für Regelungs- und Steuerungssysteme
Institutsleiter: Prof. Dr.-Ing. Sören Hohmann

Erläuterungen zu den Übungsblättern:

Die Übungsblätter enthalten verschiedene Typen von Aufgaben:

- Anwendungsaufgaben:
  - *Übungsaufgaben* sind mit UE gekennzeichnet und werden in der Übung vorgerechnet. Die Lösungswege sind auf den in der Übung gezeigten Folien enthalten, welche nach der Übung im Internet herausgegeben werden.
  - *Tutoriumsaufgaben* sind mit TU gekennzeichnet und sind ausschließlich zum selbstständigen Bearbeiten vor bzw. im Tutorium gedacht. Diese sind regelmäßig auch in MATLAB/SIMULINK zu bearbeiten. Lösungen zu Tutoriumsaufgaben werden im jeweiligen Tutorium besprochen und ggf. nach der Tutorienwoche bereitgestellt.

- *Trainingsaufgaben* (TR):

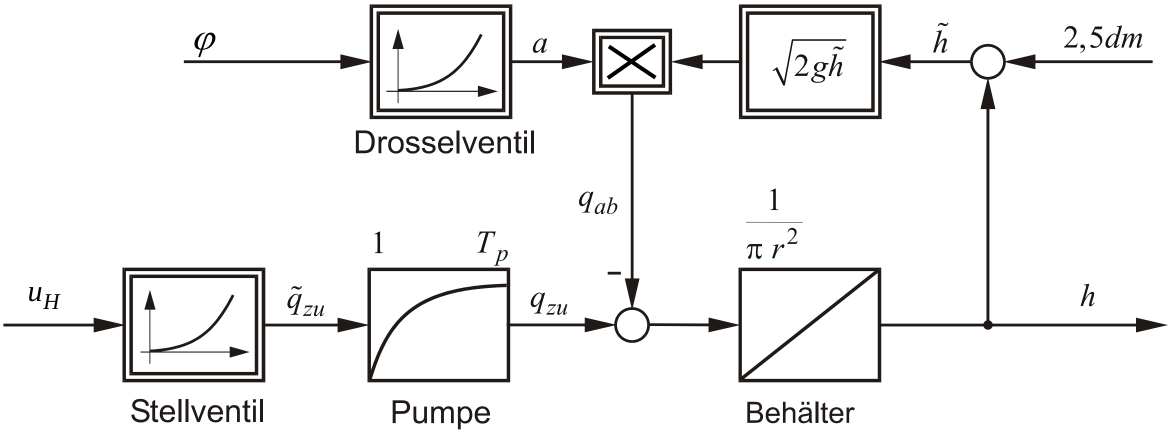
  Diese enthalten mehrere ähnliche Teilaufgaben zum Erlernen und Trainieren (auch im Hinblick auf die Klausur) von elementaren, klar abgegrenzten Methoden. Der Schwierigkeitsgrad innerhalb einer Trainingsaufgabe steigt zunehmend an, beginnend bei der ersten Teilaufgabe.
  In der Übung werden von Trainingsaufgaben lediglich die ersten Teilaufgaben im Rahmen der Wiederholung des Stoffs behandelt. Ggf. werden die anderen Teilaufgaben teilweise oder auf Nachfrage im Tutorium besprochen.
  Es liegt in der Hand jedes einzelnen Studierenden, ob und wie viel er sich mit den restlichen Teilaufgaben beschäftigt. Es wird jedoch dringend empfohlen, auch und gerade diese Methoden selbstständig zu üben, da sie das Handwerkszeug für die SRT darstellen und oft erst in der Anwendung hinreichend verstanden werden.
  Eine Kurzlösung zu den Trainingsaufgaben wird im Anschluss an die Tutorienwoche im Internet herausgegeben.

Der Schwierigkeitsgrad von Aufgaben wird durch Pluszeichen symbolisiert.

+	bedeutet	*leichte Aufgabe*
+++(+)	bedeutet	*Klausurniveau (oder höher)*

Aufgabe 7 (TU) ++

Betrachten Sie erneut den Füllstandsversuch aus Aufgabe 6. Die folgende Abbildung zeigt das bereits ermittelte nichtlineare Signalflussbild.



a) Implementieren Sie das nichtlineare System innerhalb eines Subsystems in SIMULINK. Verwenden Sie für die Ventilkennlinie des Stellventils eine *user defined function* und für diejenige des Drosselventils eine *Lookup-Table*.
Übergeben Sie die Signale u und h zwischen dem Subsystem und dem übergeordneten SIMULINK-Plan mit Hilfe der *Input*- und *Output-Ports*. Benutzen Sie als Signalquelle für das Eingangssignal u zunächst eine Konstante und stellen Sie den Verlauf der Füllstandshöhe in einem Scope dar.
Machen Sie sich mit dem System vertraut, indem Sie die Ventilstellung von Hand variieren und die Auswirkungen auf die Füllstandshöhe betrachten. Welche Form hat das Ausgangssignal, wenn Sie den Ventilhub mit einem sinusförmigen Eingangssignal um einen Betriebspunkt beaufschlagen?

Der in Aufgabe 6 bereits ermittelte Arbeitspunkt lautet

$$\varphi_R = 0{,}25\ rad\ ,\ u_{H,R} = 1{,}5\ \text{cm sowie}\ h_R = 2{,}5\ \text{dm}.$$

b) Implementieren Sie auch das linearisierte System in einem weiteren Subsystem in SIMULINK. Übergeben Sie diesem als Eingangssignal die Abweichung vom Arbeitspunkt $\Delta u_H = u_H - u_{H,R}$ und stellen Sie die Füllstandshöhe $h = \Delta h + h_R$ des linearisierten Systems im selben Scope dar, wie die des nichtlinearen Systems.

c) Vergleichen Sie das Systemverhalten der beiden Systeme bei kleinen Auslenkungen aus dem Arbeitspunkt. Wie verhält sich das linearisierte System gegenüber dem nichtlinearen System dagegen bei großen Auslenkungen bzw. an einem deutlich abweichenden Arbeitspunkt?

Aufgabe 8 (TR)

Im Folgenden sind einige zeitkontinuierliche bzw. zeitdiskrete Signale

$$Y(s)\ /\ y(t)\ /\ Y_z(z)\ /\ \left(y_k\right)$$

oder Systeme

$$G(s)\ /\ y^{(n)} = f(y, y', ..., u, u', ...)\ /\ \mathrm{G}_z(z)\ /\ \mathrm{y}_k = f(y_{k-1}, y_{k-2}, ..., u_k, u_{k-1}, ...)$$

gegeben. Berechnen Sie die jeweilige Frequenzbereichsdarstellung, sofern die Zeitbereichsdarstellung gegeben ist und umgekehrt.

a)

$$\ddot{y} = u - \sqrt{2}\dot{u} + \tfrac{3}{2}\dot{y}$$

b)

$$y_k = \tfrac{1}{2}y_{k-1} + 3u_k - 6u_{k-1}$$

c)

$$Y_z(z) = \frac{8z}{(z-1)^2}$$

d)

$$\left(y_k\right) = 2k^2 + e^{-2k\sqrt{2}}$$

e)

$$y(t) = 2\sin(3t) + 1$$

f)

$$Y(s) = \frac{1}{s^2}\frac{2(s-1)}{s^2+4}$$

g)

$$G_z(z) = \frac{3z}{z-1} - \frac{z}{z-e^{-T}}$$

h)

$$G(s) = \frac{s+4}{s^2+4s+3}e^{-3s}$$

Aufgabe 9 (TR)

Ordnen Sie den folgenden kontinuierlichen Systemen anhand ihrer Pole und Nullstellen sowie der Sprungantworten die Eigenschaften *stabil/instabil*, *Allpass(AP)/ Minimalphasensystem(MS)/Nichtminimalphasensystem (NMS)*, *schwingfähig, stationär genau* zu. Diskretisieren Sie diese Systeme dann in MATLAB mit dem Befehl *c2d* und den jeweils angegebenen Abtastzeiten. Überprüfen Sie anhand der Sprungantworten inwiefern sich den zeitdiskreten Systemen dieselben bzw. abweichende Eigenschaften zuordnen lassen als die ursprünglichen kontinuierlichen Pendants aufweisen! Verwenden Sie dazu geeignete MATLAB-Befehle.

a)

$$G_1(s) = \frac{2s+6}{s+1}, T = 0.1$$

b)

$$G_2(s) = \frac{3}{8} + \frac{1}{2s}, T = 1$$

c)

$$G_3(s) = \frac{s-2}{s^2+2s+5}, T_1 = 0.5, T_2 = 2$$

d)

$$G_4(s) = \frac{s^2-5s+6}{s^2+5s+6}, T_1 = 1, T_2 = 0.1$$

Aufgabe 10 (TR)

Bestimmen Sie das stationäre Verhalten der im Folgenden beschriebenen zeitkontinuierlichen Systeme mithilfe des Endwertsatzes der Laplace-Transformation. (Berechnen Sie ggf. zuvor die Übertragungsfunktion.) Bestimmen Sie auch den Wert der Ausgangsgröße für t=+0 bei einem Eingangssprung als Führungsgröße. Überprüfen Sie Ihre Ergebnisse mit MATLAB, indem Sie die Systeme definieren, die Sprungantworten plotten und ggf. den stationären Endwert ablesen.

a)
$$G_1(s) = \frac{1}{2s+2}$$

b)
$$G_2(s) = \frac{s+3}{s-3}$$

c)
$$G_3(s) = \frac{(s-5)}{s+2} e^{-\frac{3}{2}s}$$

d)
$$\ddot{y} = u - 3y$$

Aufgabe 11 (UE)

Gegeben ist die Übertragungsfunktion eines offenen Kreises mit

$$F_o(s) = \frac{-2 \cdot k \cdot T_1 \cdot s^2}{(1+T_1 \cdot s)\cdot(1+T_2 \cdot s)}, \quad T_1 > 0,\ T_2 > 0,\ k > 0.$$

a) Wo liegen die Pole und Nullstellen des Systems qualitativ? Was sagt dies über Stabilität, Minimalphasigkeit und stationäre Genauigkeit aus? Zeichnen Sie die Lage der Pole für $k=1, T_1=2$ und $T_2=1$ mit MATLAB!

b) Zeichnen Sie mit MATLAB für diese Werte die Nyquist-Ortskurve sowie das Bode-Diagramm des offenen Kreises!

c) Analysieren Sie den geschlossenen Regelkreis auf Stabilität! Berechnen Sie dazu die Lage der Pole des geschlossenen Kreises und lassen Sie sich die Sprungantwort des Systems anzeigen, um ihr Ergebnis zu überprüfen!

d) Für welche Werte von k ist der geschlossene Kreis stabil?

Im Folgenden soll das entsprechende zeitdiskrete System untersucht werden. Die Abtastzeit betrage dabei T=0.5s.

e) Wie lautet die z-Übertragungsfunktion des diskretisierten offenen Kreises?

f) Plotten Sie auch für das diskretisierte System die Lage der Pole und Nullstellen des offenen Kreises, sowie Nyquist-Ortskurve und Bode-Diagramm und vergleichen Sie mit den jeweiligen kontinuierlichen Darstellungen! Betrachten sie ebenfalls die Sprungantwort des geschlossenen zeitdiskreten Kreises!

Aufgabe 12 (TU) ++

a) Implementieren Sie folgendes PT_2-Glied als Transfer-Function in SIMULINK:

$$\frac{10}{s^2+0.5s+10}$$

Beaufschlagen Sie das Übertragungsglied mit einem Einheitssprung (Konstante) und betrachten Sie die simulierte Sprungantwort in einem Scope. Geben Sie außerdem mit einem Output-Port die Sprungantwort an den MATLAB-Workspace aus und speichern Sie Zeit- und Amplitudenvektor unter neuem Variablennamen ab.

Simulieren Sie nun das gleiche System mit einem Eulerverfahren mit fester Schrittweite. Dies lässt sich über die *Configuration Parameters* im Menü *Simulation* einstellen (fixed-step → ode1). Variieren Sie dabei die Schrittweite zwischen 0,005s und 0,5s und speichern Sie jeweils die Ausgangssignale unter neuem Namen im Workspace ab.

Plotten Sie schließlich die verschiedenen Resultate jeweils gemeinsam mit der ersten Lösung und betrachten Sie die Auswirkungen der Schrittweite auf die Simulationsgenauigkeit und die numerische Stabilität.

Weiterhin soll das System $G(s)=\frac{10}{2s+5}$ in MATLAB simuliert werden. Dazu ist das System im Zeitbereich zu implementieren und das Eulerverfahren von Hand zu programmieren.

b) Berechnen Sie die kontinuierliche Differentialgleichung des Systems und implementieren Sie dieses als MATLAB-function!

Der Funktion sollen als Argumente die aktuellen Werte der Ausgangsgröße sowie der Eingangsgröße übergeben werden. Als Ergebnis soll die Funktion die Ableitung der Ausgangsgröße zurückgeben.

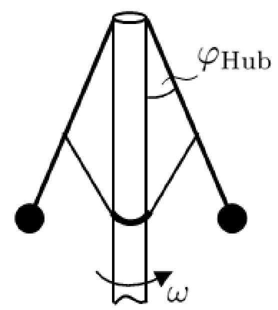
Programmieren Sie nun in einem MATLAB-Skript die Simulation der Differentialgleichung mittels Eulerverfahren! Definieren Sie dazu zunächst einen Startwert der Ausgangsgröße sowie einen von Ihnen gewählten Eingangsgrößenvektor! Rufen Sie anschließend in einer Schleife zyklisch die Funktion der Differentialgleichung sowie die Integration der Ableitung gemäß Eulerverfahren auf! Achten Sie auch auf die Wahl der Simulationsschrittweite!

Anmerkung: Da für solche Simulationsverfahren nur erste Ableitungen nach der Zeit auftreten dürfen, empfiehlt es sich bei komplizierteren Systemen, Differentialgleichungen höherer Ordnung durch das Einführen von Hilfsgrößen in Differentialgleichungssysteme erster Ordnung zu überführen.

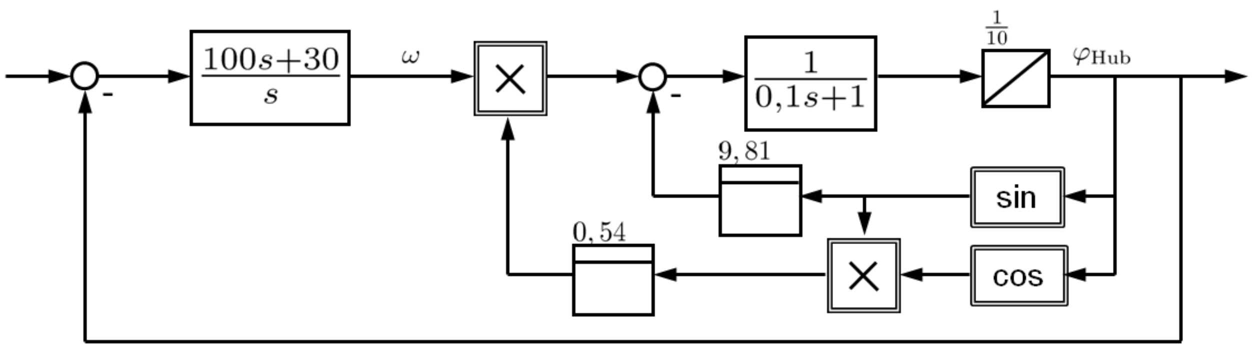
Bsp: $\ddot{y} = y + u$ ➔ mit $y = h_1,\ \dot{y} = h_2$ folgt: $\begin{aligned}\dot{h}_1 &= h_2\\ \dot{h}_2 &= h_1 + u\end{aligned}$

Aufgabe 13: MATLAB/SIMULINK (TR)

Diese Aufgabe soll dazu dienen, den Umgang mit Matlab und Simulink zu trainieren (auch mit Hilfe des Kompendiums). Bitte versuchen Sie, diese Aufgabe eigenständig zu lösen, um für sich selbst festzustellen, ob Sie noch grundsätzliche Schwierigkeiten mit der Software haben.



a) Bauen Sie in Simulink das im Folgenden gegebene vereinfachte Signalflussbild eines geregelten Fliehkraftpendels nach. Stellen Sie dabei die Drehgeschwindigkeit ω und den Hubwinkel φ_{Hub} in einem Scope dar.



Hinweis: Durch das Produkt der Stellgröße ω mit $\sin(\varphi_{Hub})$ kann das System nur anlaufen, wenn der Anfangswert des Hubwinkels ungleich 0 ist. Dies kann erreicht werden, indem die Eigenschaft *Initial Condition* des Integrators auf einen entsprechenden Wert (z.B. 0.01) gesetzt wird.

Geben Sie konstante Sollwerte zwischen 0 und $90° = \pi/2 \approx 1{,}57$ vor und simulieren Sie den aufgebauten Regelkreis. Verwenden Sie auch andere Signalquellen aus der Bibliothek *Simulink/Sources* (z.B. *Sine Wave*) zur Vorgabe von Sollwertverläufen.
Variieren Sie auch die Simulationsverfahren und die Schrittweiten um den Effekt numerischer Instabilität sichtbar zu machen.

b) Definieren Sie in Matlab die folgenden Vektoren und Matrizen!

$$a = \begin{bmatrix} 1 & 3 & 1 & 3 & 2 & 1 & 1 & 3 & 3 \end{bmatrix}$$
$$b = \begin{bmatrix} 1 & 1 & 3 & 3 & 5 & 3 & 1 & 3 & 1 \end{bmatrix}$$
$$D = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 3 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

$t = [0...20]$ in 200 Schritten (Befehl *linspace*), $u = \sin(t)$

Plotten Sie die Vektoren *b* über *a* sowie *u* über *t* mit dem Befehl *plot*!
Berechen Sie in Matlab das Produkt $a \cdot b^T$ sowie das Quadrat und die Inverse von *D*!

c) Erzeugen Sie in MATLAB einen dynamischen *plot* eines von einem komplexen Drehzeiger beschriebenen Kreises! Initialisieren Sie dazu ein MATLAB-*figure*, und plotten Sie in einer Schleife die Zeigerstellung zum jeweiligen Zeitpunkt sowie den bisher beschriebenen Kreisbogen! Machen Sie sich ggf. den *hold*-Befehl zunutze.

d) Plotten Sie in eine neues Matlab-*figure* die Funktionen $f_1(t) = \sqrt{t}$ und $f_2(t) = e^{-3t} \cdot \sin(5t)$ und experimentieren Sie mit den Befehlen *grid*, *xlim*, *ylim*, *xlabel*, *ylabel*, *title* sowie *legend* um den plot zu modifizieren und zu beschriften!