

## Übungsblatt 5: Verzögerungsglieder und LTI-Systeme in Python

Prof. Dr. Moritz Diehl, Leonard Fichtner

1. Untersuchen Sie die Systeme mit folgenden Differentialgleichungen:

I.  $\ddot{y} + 2\dot{y} + y = u$

II.  $\ddot{y} + y = u - \dot{y}$

III.  $2\ddot{y} + 8\dot{y} = 2u - 2y$

IV.  $4\ddot{y} + 2\dot{y} = u - y$

- (a) Zu welcher Gruppe von regelungstechnischen Übertragungsgliedern gehören die Systeme? (0,5 P.)  
(b) Berechnen Sie jeweils die Zeitkonstante  $T$  und die Dämpfung  $d$ . (1 P.)  
(c) Bringen Sie alle Systeme in Zustandsraumdarstellung (zB. Regelungsnormalform). (1 P.)  
(d) Wie wird die Ausgangstrajektorie eines Systems für (0,5 P.)

$$u(t) = \begin{cases} 0 & \text{für } t < 0 \\ 1 & \text{für } t \geq 0 \end{cases} \quad \text{und} \quad y(0) = \dot{y}(0) = \ddot{y}(0) = \dots = 0 \quad \text{genannt?}$$

- (e) (Python) Simulieren Sie die Sprungantwort der Systeme I bis IV für 20 Sekunden und plotten Sie den Verlauf von  $y(t)$  aller Systeme in dasselbe Fenster. (1 P.)  
(f) Welche Auswirkung hat die Dämpfung auf das Systemverhalten? (1 P.)  
(g) \*(Python) Plotten Sie die Ausgangstrajektorie von System II mit einer halb so schnellen Zeitskala, also in 'Zeitlupe', zusammen mit der unveränderten Ausgangstrajektorie von System IV im selben Fenster. (\*0,5 P.)  
(h) \*Welche Auswirkung hat die Zeitkonstante auf das Systemverhalten? (\*0,5 P.)

2. (Python) Mit dem `control` Paket lassen sich LTI-Systeme sehr bequem analysieren. In dieser Übung wollen wir dieses Paket verwenden, um das folgende System zu analysieren

$$\dot{x}(t) = \begin{bmatrix} -4 & 3 & -3 \\ -1 & -1 & -2 \\ -2 & 4 & -4 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} u(t), \quad y(t) = \begin{bmatrix} 3 & 1 & 0 \end{bmatrix} x(t) + u(t).$$

- (a) Geben Sie das System in Python ein. (0,5 P.)  
TIPP: Ein Zustandsraummodell kann mit Hilfe des Befehls `ss(...)` erzeugt werden.
- (b) Plotten Sie die Sprungantwort des Systems unter Verwendung des Befehls `step_response(...)`.  
Wie lange benötigt das System zum Einschwingen (2% settling time)? (1 P.)
- (c) Simulieren Sie die Systemantwort auf den Anfangswert  $x_0 = [1, -1, 1]$  mit Hilfe des `initial_response(...)` Befehls. (0,5 P.)
- (d) Plotten Sie die Systemantworten auf Rechtecksignale mit den Frequenzen  $f \in \{0.1 \text{ Hz}, 1 \text{ Hz}, 10 \text{ Hz}\}$ . Das Rechtecksignal soll zwischen 0.5 und  $-0.5$  wechseln.  
TIPP: Simulieren Sie das System mit `forced_response(...)`.  
Dies ist eine Simulationsroutine des `python-control` Pakets für lineare Systeme und sollte nicht mit `nlsim(...)` verwechselt werden, welches in den vorherigen Übungsblättern zur Simulation nichtlinearer Systeme verwendet wurde.  
TIPP: Ein Rechtecksignal mit vorgebarerer Frequenz kann in Python erzeugt werden, indem Sie zunächst mit `np.linspace(...)` einen Zeitvektor für zwei Perioden des Signals mit 100 Zeitpunkten erstellen. Durch Verwendung von `scipy.signal.square(...)` lässt sich aus diesem Vektor das Rechtecksignal erzeugen. Damit `square(...)` benutzt werden kann muss das Paket `scipy` installiert werden. Alternativ dazu kann auch ein Sinus-Signal erzeugt werden und mit `np.sign(...)` das Vorzeichen extrahiert werden sodass ein Rechtecksignal entsteht. (1 P.)