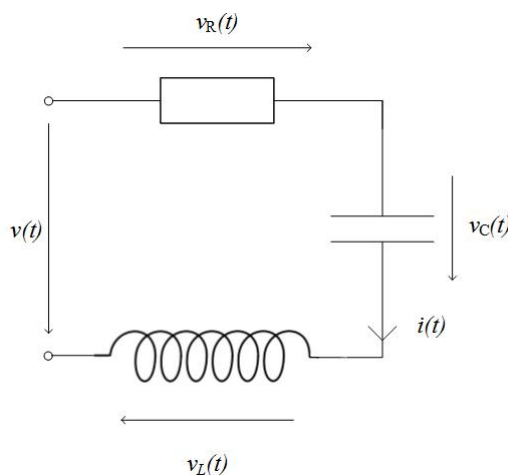


## Python Übung Blatt I - Simulation

Prof. Dr. Moritz Diehl, Leonard Fichtner

---

Auf diesem Blatt soll der elektrische RLC-Kreis aus dem Skript (S. 34f) auf zwei Arten simuliert werden: Zunächst mit der bereits bekannten `nlsim()`-Funktion, dann mit dem `python-control`-Paket. Das Modell ist wie folgt gegeben:



$$x(t) = \begin{bmatrix} v_C(t) \\ i(t) \end{bmatrix}, \quad u(t) = v(t), \quad y(t) = v_C(t), \quad \dot{x} = f(x, u)$$
$$f(x, u) = \begin{bmatrix} i(t)/C \\ \frac{1}{L}(v(t) - Ri(t) - v_C(t)) \end{bmatrix} \quad \text{mit } x(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Dabei ist  $R = 25 \Omega$ ,  $C = 50 \mu\text{F}$  und  $L = 200 \text{ mH}$ .

### 1. Simulieren mit `nlsim()`

- (a) Simulieren Sie das System über einen Zeitraum von 500 ms. Dabei ist  $u(t) = 1 \text{ V}$ . Nutzen Sie hierfür die Funktion `nlsim()`, die bereits von Übungsblatt 3 bekannt ist. `nlsim()` ist nicht von Python gegeben und kann auf der Kursseite heruntergeladen werden (Modul `toolbox_sr1.py` bei Übungsblatt 3). `nlsim()` benötigt außerdem die Funktion `rk()`, die am selben Ort zu finden ist. Dort ist auch ein Beispiel zur Benutzung gegeben. Verwenden Sie Zeitschritte von  $\Delta t = 0,5 \text{ ms}$ . Erstellen Sie einen Plot der Trajektorien von  $u(t)$  und  $y(t)$ .

*Hinweis: Erstellen Sie zunächst die beiden Funktionen  $f(x, u)$  und  $y(x, u)$ . Erstellen Sie dann ein Skript, in dem Sie  $x_0$ ,  $\Delta t$  und  $u$  definieren und anschließend `nlsim()` aufrufen.*

- (b) Wiederholen Sie die Simulation mit einem Rechtecksignal als Input. Dieses soll zwischen 0 und 1 V wechseln mit der Kreisfrequenz  $\omega = 10\pi \text{ s}^{-1}$ .

*Hinweis: Generieren Sie Rechtecksignal mithilfe eines Sinussignals und dem Befehl `np.sign()`.*

### 2. Simulieren mit `python-control`

Das `python-control`-Paket stellt Funktionen zum Entwurf sowie zur Analyse linearer Steuerungs- und Regelungssysteme zur Verfügung. Wir wollen uns hier ein paar dieser Funktionen ansehen.

- (a) Erstellen Sie unser Modell als *LTI model*. Überlegen Sie sich dafür zunächst, wie die Systemmatrizen  $A, B, C$  und  $D$  aussehen.

*Hinweis: `ss()`*

- (b) Simulieren Sie das System mit dem konstanten Kontrollinput aus Aufgabe 1(a).

*Hinweis: `forced_response()`*

- (c) Die soeben durchgeführte Simulation war eine Analyse der *step response*. Hierfür stellt `python-control` auch den Befehl `step_response()` zur Verfügung. Wiederholen Sie die Simulation unter Verwendung von diesem.

- (d) Führen Sie eine Simulation durch, bei der das Rechtecksignal aus Aufgabe 1(b) als Inputsignal verwendet wird.