Modellierung dynamischer Systeme in Matlab

Lösen von Differentialgleichungen in Matlab I

Lösen von nicht-linearen Differentialgleichung ist ebenfalls in Matlab (ohne Simulink) möglich

Vorteile:

- *Modelling with Code* (Versionierung, Parametrierung)
- Flexible Anbindung an weitere IT-Systeme (Datenbanken, UI, etc.)
- Export als binary file für 3rd-Party Simulationsumgebungen (z.B. als DLL via Code Generierung in C/C++)

Formulierung des mathematischen Modells als

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = f(t, \mathbf{x}(t)) \qquad \text{mit Anfangswert} \qquad \mathbf{x}(t_0) = \mathbf{x}_0$$

Hinweis: Die Funktion $f(t, \mathbf{x}(t))$ muss nicht zwangsweise eine lineare Abbildung der Zustandsgrößen $\mathbf{x}(t)$ sein.

Function Handles in Matlab I

Funktionen sind in Matlab KEINE *first-class citizens*:

> *Funktionale Programmierung im klassischen Sinn nicht möglich*

Trotzdem lassen sich sogenannte *function handles* definieren

```
% my_function.m
function y = my_funktion(x)
   y = x.^2;
end
```

```
>> renamed_function = @my_function;
renamed_function =
  function_handle with value:
    @my_function

>> renamed_function(3)
ans =
     9
```

Function Handles in Matlab II

Funktion lassen sich in Matlab auch *anonym* (d.h. als sogenannte Lambda-Funktionen) definieren

```
>> square = @(x) x.^2

square =

  function_handle with value:

    @(x)x.^2
```

Im Gegensatz zu anderen Programmiersprachen ist es nicht möglich beliebige Ausdrücke zu realisieren wie z.B.

- if-else Verzweigungen
- komplexe Logik

Lösen von Differentialgleichungen in Matlab II

Definition einer Funktion zur Abbildung der Zustandsgrößen

```
function dxdt = xpunkt(t, x)
   dxdt = [
      ...
   ]
```

Lösen der Differentialgleichung mittels des Runge-Kutta Verfahrens `ode45()`

```
>> [t, x] = ode45(@xpunkt, tlim, x0)
```

Weitere Lösungsalgorithmen in Matlab

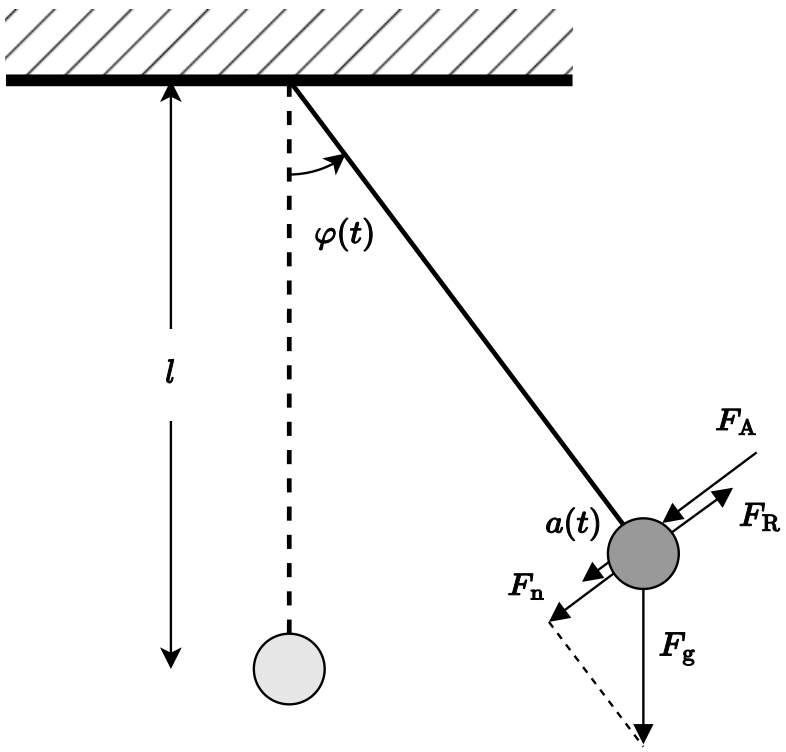
- `ode23`
- `ode113`
- `ode15s`
- `...`

Beispiel: Simulation eines Pendels mit Antrieb

$$F_\mathrm{n}(t) = m \cdot g \cdot \sin(\varphi(t))$$

$$F_\mathrm{R}(t) = -\frac{2}{\pi} \cdot F_{\mathrm{R},0} \cdot \arctan(10 \cdot \dot{\varphi}(t))$$

$$a(t) = -l \cdot \ddot{\varphi}(t)$$



Externer Antrieb

$$F_\mathrm{A}(t) = 20\,\mathrm{N} \cdot \sin(2\pi \cdot 0{,}2\,\mathrm{Hz} \cdot t)$$

Kräftegleichgewicht

$$m \cdot a(t) = F_\mathrm{n}(t) - F_\mathrm{R}(t) + F_\mathrm{A}(t)$$

Aufstellen der Differentialgleichung

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \varphi(t) \\ \dot{\varphi}(t) \end{bmatrix}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} \dot{\varphi}(t) \\ \ddot{\varphi}(t) \end{bmatrix} = \begin{bmatrix} \dot{\varphi}(t) \\ -\frac{F_\mathrm{n}(t) - F_\mathrm{R}(t) + F_\mathrm{A}(t)}{l \cdot m} \end{bmatrix}$$

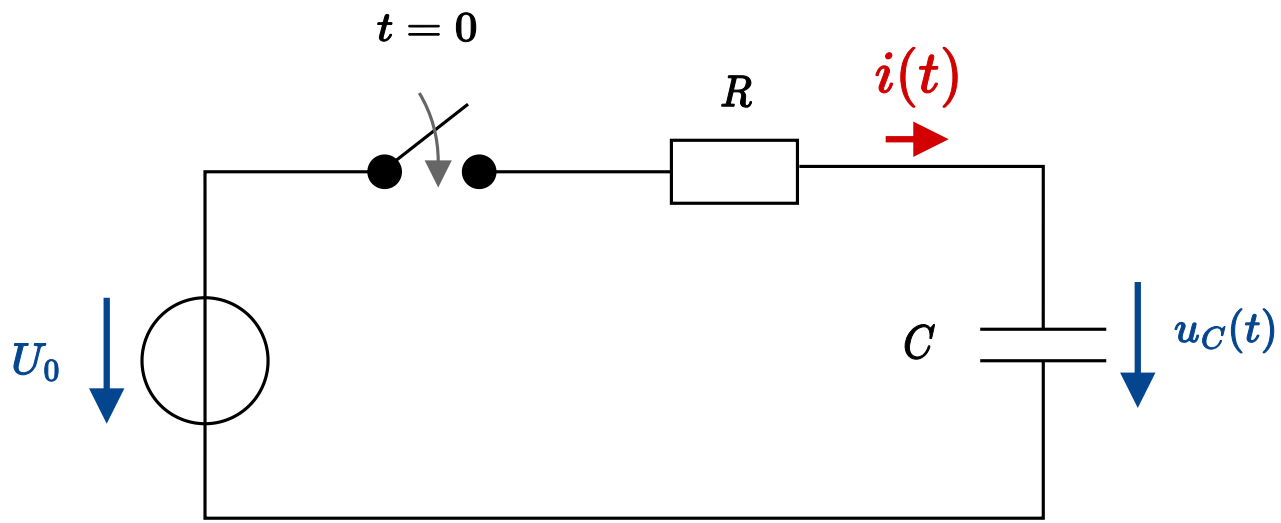
Aufgabe XVII

1. Erstellen Sie ein Simulationsmodell für den Einschaltvorgang eines ungeladenen Kondensators mit .

$$U_0 = 2\,\mathrm{V} \qquad R = 1\,\Omega \qquad C = 1\,\mathrm{mF}$$

2. Vergleichen Sie das Ergebnis mit der analytischen Lösung der Differentialgleichung

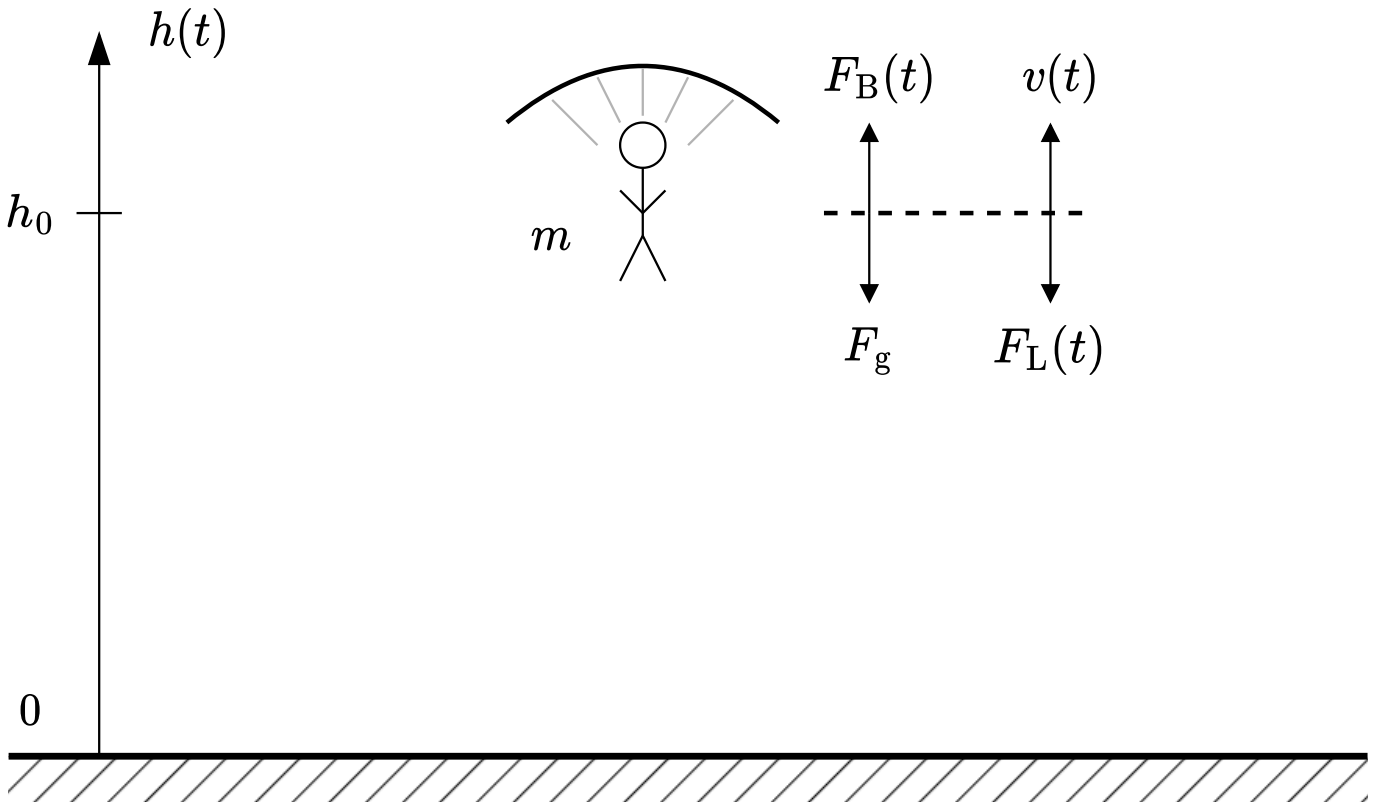
$$u_C(t) = U_0 \cdot \left(1 - \mathrm{e}^{-\frac{t}{RC}}\right)$$



Aufgabe XVIII

Erstellen Sie ein Simulationsmodell für den Fallschirmspringerflug

Die genaue Beschreibung der Systemmodellierung und Zahlenwerte finden Sie hier



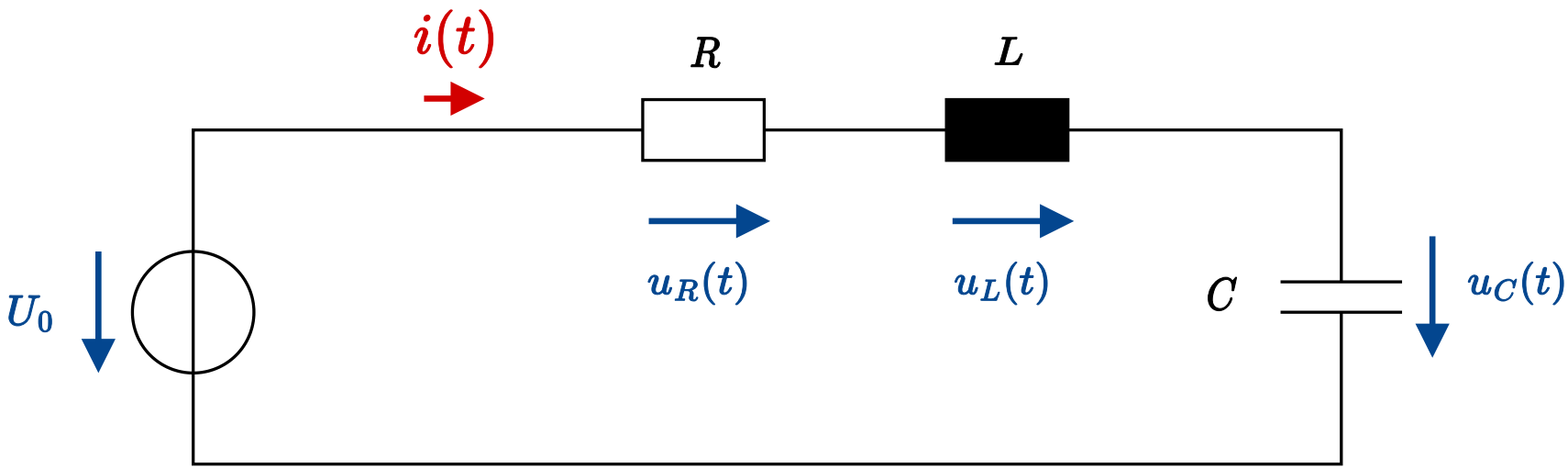
Aufgabe XIX

Erstellen Sie ein Simulationsmodell für den Einschaltvorgang eines RLC-Schwingkreises mit

$$U_0 = 1\,\mathrm{V} \qquad R = 10\,\Omega \qquad L = 10\,\mathrm{mH} \qquad C = 100\,\mathrm{nF}$$

Plotten Sie den zeitlichen Verlauf von

- Kondensatorspannung $u_C(t)$
- Induzierter Spannung der Induktivität $u_L(t)$



Aufgabe XX

Erstellen Sie ein Simulationsmodell in Matlab zur Simulation eines epidemischen Verlaufes

$$\dot{S}(t) = -\alpha \cdot S(t) \cdot I(t)$$

$$\dot{I}(t) = \alpha \cdot S(t) \cdot I(t) - \beta \cdot I(t)$$

$$\dot{R}(t) = \beta \cdot I(t)$$

Dabei entsprechen die betrachteten Größen

- $S(t)$: Zahl der anfälligen Personen (*susceptible*)
- $I(t)$: Zahl der infizierten Personen (*infectious*)
- $R(t)$: Zahl der genesenen Personen (*recovered*)

YouTube: Mehr "Corona-Mathematik": Wie Epidemien modelliert werden

Aufgabe XXI

1. Erstellen Sie ein Simulationsmodell in Matlab für den DC-Motor
2. Erstellen Sie ein Simulationsmodell in Matlab für das HGÜ-Modell

Referenzen

[1] A. Angermann, M. Beuschel, M. Rau, U. Wohlfarth, *Matlab - Simulink - Stateflow,* Oldenburg Verlag.