

Simulation eines Aufwärtswandlers

```
# -----  
# Simulation eines Aufwärtswandlers, Tilman Küpper, 2019-04-04.  
# Für weitere Details zur Schaltungssimulation mit Python siehe:  
# https://kuepper.userweb.mwn.de/elektronik/formelsammlung-elektronik.pdf  
# -----  
from scipy.integrate import odeint  
from numpy import pi, arange, fmod  
import matplotlib.pyplot as plt  
  
freq = 200000.0      # Frequenz (in Hz)  
tast = 0.333333     # Tastgrad (0...1)  
Ra = 25.0           # Lastwiderstand (in Ohm)  
Ue = 5.0           # Eingangsspannung (in Volt)  
C = 0.00001        # Kapazität (in F)  
L = 0.0001         # Induktivität (in H)  
Rl = 0.2           # Widerstand der Spule (in Ohm)  
Rc = 0.1           # Widerstand des Kondensators (in Ohm)  
Us = 0.4           # Schwellenspannung der Diode (in V)  
rf = 0.5           # diff. Widerstand der Diode (in Ohm)  
Rdson = 0.1        # Widerstand des eingeschalteten MOSFETs (in Ohm)  
step = 0.01 / freq # Simulations-Zeitschritt (mind. 100 pro Periode)  
stop = 0.01        # Simulationsdauer (in Sekunden)  
y0 = [0.45, 7.5]   # Startwerte für il und uc  
  
# -----  
# In welchem Zustand befindet sich die Schaltung zum Zeitpunkt t?  
# (Zustand 0 --> MOSFET leitet, Zustand 1 --> MOSFET sperrt)  
# -----  
def zustand(t):  
    if fmod(t + pi, 1 / freq) < (tast / freq):  
        return 0  
    else:  
        return 1  
  
# -----  
# Ausgangsspannung aus Spulenstrom und Kondensatorspannung berechnen  
# -----  
def ua(il, uc, t):  
    if zustand(t) == 0:  
        return Ra / (Ra + Rc) * uc  
    else:  
        return Ra / (Ra + Rc) * (Rc * il + uc)  
  
# -----  
# Definition des Differentialgleichungssystems  
# -----  
def stepup_dgl(y, t):  
    il = y[0]  
    uc = y[1]  
  
    if zustand(t) == 0:  
        dil_dt = (Ue - il * (Rl + Rdson)) / L  
        duc_dt = -uc / (C * (Ra + Rc))  
    else:  
        dil_dt = (Ue - Us - ua(il, uc, t) - il * (Rl + rf)) / L  
        duc_dt = (ua(il, uc, t) - uc) / (C * Rc)  
  
    dy_dt = [dil_dt, duc_dt]  
    return dy_dt
```

```

# -----
# Differentialgleichungssystem numerisch lösen,
# zeitliche Verläufe von Spannungen und Strömen plotten.
# -----
t_arr = arange(0, stop, step)
y_arr = odeint(stepup_dgl, y0, t_arr, hmax=step)
il_arr = y_arr[:, 0]
uc_arr = y_arr[:, 1]

# Aus den bereits bekannten Verläufen von Spulenstrom und Kondensatorspannung
# werden nun noch die Verläufe der Ausgangsspannung und des Kondensatorstroms
# berechnet, damit diese ebenfalls gezeichnet werden können.
ua_arr = []
ic_arr = []

for t, il, uc in zip(t_arr, il_arr, uc_arr):
    UA = ua(il, uc, t)
    ua_arr.append(UA)
    ic_arr.append(zustand(t) * il - UA / Ra)

# Nur die letzten 500 Simulationsschritte werden gezeichnet;
# der Einschwingvorgang am Anfang interessiert uns nicht.
d = 500
plt.subplot(2, 1, 1)
plt.plot(t_arr[-d:] * 1000, ua_arr[-d:], 'b-', linewidth=2)
plt.grid(True, color='gray', linestyle='dashed')
plt.ylabel('U / V')

plt.subplot(2, 1, 2)
plt.plot(t_arr[-d:] * 1000, il_arr[-d:], 'r-', linewidth=2)
plt.plot(t_arr[-d:] * 1000, ic_arr[-d:], 'r-', linewidth=2)
plt.grid(True, color='gray', linestyle='dashed')
plt.ylabel('I / A')
plt.xlabel('t / ms')

```

