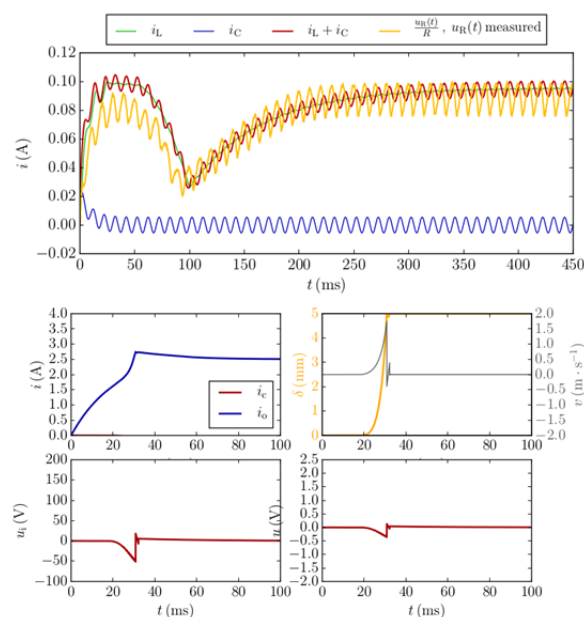


Software

Knihovna pro simulaci elektromagnetických ventilů



```
Otevřít Václav Uložit
class Valve:
    def __init__(self):
        self.Rc = 10.0; self.Ro = 10.0
        self.Lc = 0.04/2.0; self.Lo = 1.044/2.0
        self.Nc = 1000; self.No = 2300
        rho = 7500
        V = np.pi*(0.01**2*0.01**2)*0.001
        self.m = V*rho + 0.1
        self.d = 0.01
        self.load()
        self.fg = 0.0*np.pi*0.01**2
        self.fg = -self.m*9.81
        self._dt = 1e-4
        self._dic = 0.0; self._dlo = 0.0;
        self._dd = 0.0
    def load(self):
        ffile = open('fild.txt','r')
        data = json.load(ffile)
        ffile.close()
        Ic = data['Ic']; Io = data['Io']; d = data['d']
        fm = []; Pc = []; Po = []
        for t in range(len(Ic)):
            fto = []; Pcto = []; Pcto = []
            for i in range(len(Ic)):
                fto.append(data['fmg'][i*(len(Ic)+1)])
                Pcto.append(data['Pc'][i*(len(Ic)+1)])
                Pcto.append(data['Po'][i*(len(Ic)+1)])
            fm.append(fto); Pc.append(Pcto); Po.append(Pcto)
        self.savemat('data.mat', mdict={'fm': data['fm'], 'Ic': data['Ic'], 'delta': data['d']})
        self.fmg = interpolate.RegularGridInterpolator((Ic, Io, d), fm)
        self.fpc = interpolate.RegularGridInterpolator((Ic, Io, d), Pc)
        self.fpo = interpolate.RegularGridInterpolator((Ic, Io, d), Po)
    def force(self, Ic, Io, d):
        return fmg(Ic, Io, d) + self.fg + self.fg
    def inductance(self, Ic, Io, d):
        dic = self._dic; dlo = self._dlo
        Ic = self.Lc; Io = self.Lo
        if dic != 0.0:
            ic = dic
            if ic>100.0/self.Rc:
                ic = 100.0/self.Rc + dic
            if ic<100.0/self.Rc - dic:
```

- V souladu s platnou metodikou Úřadu vlády ČR je uplatňován software.
- Software vznikl v přímé souvislosti s řešením projektu *Elektromagnetické ventily s vysokou mírou vestavěné inteligence* (TH04010270).
- Knihovna pro simulaci elektromagnetických ventilů implementuje matematický model elektromagnetického ventilu a umožňuje tak simulovat chování ventilu zařazeného do pneumatického nebo hydraulického systému.
- Model umožňuje simulovat dynamické děje v elektromagnetickém ventilu (přechodové jevy v napájecích obvodech ventilu a dynamika kotvy ventilu) s uvažováním transformační i pohybové složky zpětně indukovaného napětí a tlakových změn v kanálu ventilu. Obecně nelineární matematický model magnetického pole v magnetickém obvodu je v modelu reprezentován pomocí mapy magnetického indukčního toku nebo je možné využít mapy parametrů řešené rovnice (síla a indukčnost).
- Pro numerické řešení matematického modelu je použita metoda Runge-Kutta 4. řádu. Vnitřní interpolace mapy indukčního toku je řešena pomocí vícerozměrné interpolace „nearest-neighbour“. Pro ukládání všech dat je využit formát JSON (JavaScript Object Notation).
- Knihovna je implementována v podobě samostatného modulu pro jazyk Python 3 a využívá objektorvě orientovaného návrhu. Je multiplatformní a pro svůj běh vyžaduje knihovnu SciPy.

EVIDENČNÍ ČÍSLO:

22190-SW006-2019

KONTAKTNÍ OSOBA:

Ing. František Mach, Ph.D.

tel.: +42037763 4663

fmach@rice.zcu.cz

ŘEŠITELSKÉ

PRACOVNÍŠTĚ:

Západočeská univerzita v Plzni

Fakulta elektrotechnická

Regionální inovační centrum

elektrotechniky

Univerzitní 8, 306 14 Plzeň