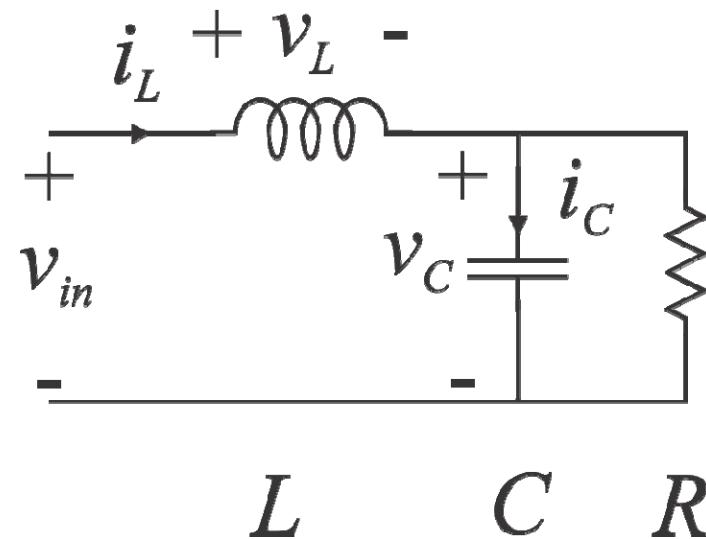

Time Domain Simulation and Optimization for Design

Homework Set 1

S.D. Sudhoff
Spring 2019

Problem 1

- Write a Matlab function to simulate a RLC circuit using the resistor companion technique



Function Input/Output

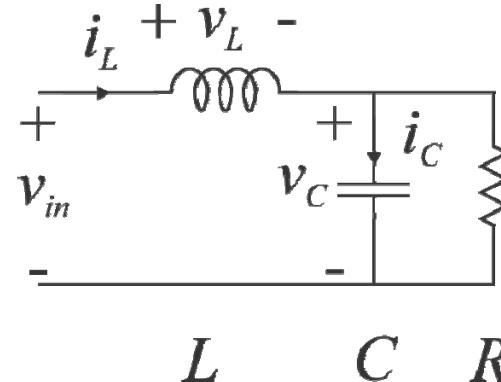
```
function [vC,iL] = RLCsimulation(t,P)
% RLCSimulation Simulates a simple RLC circuit using resistor campanion
% technique
%
% [vC,iL] = RLCsimulation(t,P)
%
% Inputs:
% t      = Vector of time values
% P      = Structure with circuit parameters
% P.vin = input voltage (V)
% P.vC0 = initial capacitor voltage (V)
% P.iL0 = initial inductor current (A)
% P.R   = Resistor resistance (Ohms)
% P.C   = Capacitor capacitance (F)
% P.L   = Inductor inductance (H)
%
% Outputs:
% vC     = Vector of capacitor voltages corresponding to t (s)
% iL     = Vector of inductor currents corresponding to t (s)
%
% Internal:
% h      = Time step (s)
% n      = Index variable
%
% Written by:
% S.D. Sudhoff
% School of Electrical and Computer Engineering
% 1285 Electrical Engineering Building
% West Lafayette, IN 47907-1285
% Phone: 765-494-3246
% Fax: 765-494-0676
```

Problem 1 Deliverables

- Deliver the function `RLCsimulation(t, P)`
- Use filename `RLCyourname.m`
 - Examples
 - `RLCJuliaAdams.m`
 - `RLCEdWood.m`
- Code graded on
 - Documentation
 - Execution speed
 - Accuracy

Problem 2

- Consider the circuit



- The capacitor and inductor energy are defined as

$$E_C = \frac{1}{2} Cv_C^2 \quad E_L = \frac{1}{2} Li_L^2$$

- Derive $f(t, \mathbf{x})$ for $\mathbf{x} = [E_C \ E_L]^T$
- Don't worry about negative voltage or current
- Deliver Problem2RogerCorman.pdf

Problem 3

- Consider our satellite model

$$[px] = \text{satellite_model}(t, x, P)$$

- Using the data in `satellite_call.m` and position and velocity specified therein as the initial condition, and the using the routine

$$[t, y] = \text{odefsrk}(\text{fhandle}, \text{tspan}, \text{yic}, \text{par}, \text{maxt})$$

determine the satellite's orbit over 50 hours.

Problem 3

- Deliverable 1: A mat file with deliverable of the form
‘Problem3aEvelynAnkers.mat’ containing the data structure
‘orbit’ with fields:
 - orbit.t – time vector (s)
 - orbit.px – x-axis position (m) corresponding to orbit.t
 - orbit.py – y-axis position (m) corresponding to orbit.t
 - orbit.vx – x-axis velocity (m/s) corresponding to orbit.t
 - orbit.vy – y-axis velocity (m/s) corresponding to orbit.t

Problem 3

- Deliverable 2
 - Define the position and velocity errors as

$$p_e = \frac{\max_t \sqrt{(p_x^*(t) - p_x(t))^2 + (p_y^*(t) - p_y(t))^2}}{\min_t \sqrt{(p_x^*(t))^2 + (p_y^*(t))^2}}$$

$$v_e = \frac{\max_t \sqrt{(v_x^*(t) - v_x(t))^2 + (v_y^*(t) - v_y(t))^2}}{\min_t \sqrt{(v_x^*(t))^2 + (v_y^*(t))^2}}$$

- Provide a code listing and a plot of p_e and v_e vs. h in the file ‘Problem3bWilliamCastle.pdf’

Problem 4

- Derive an update rule for the trapezoidal method for linear systems analogous to the update rule we derived for the backwards Euler's method for linear systems
- Provide a derivation in the file
‘Problem4JohnAjar.pdf’