ECE61016 Power Electronics Converters and Systems

Time-Domain Simulation

Homework 1

S.D. Sudhoff

Fall 2016

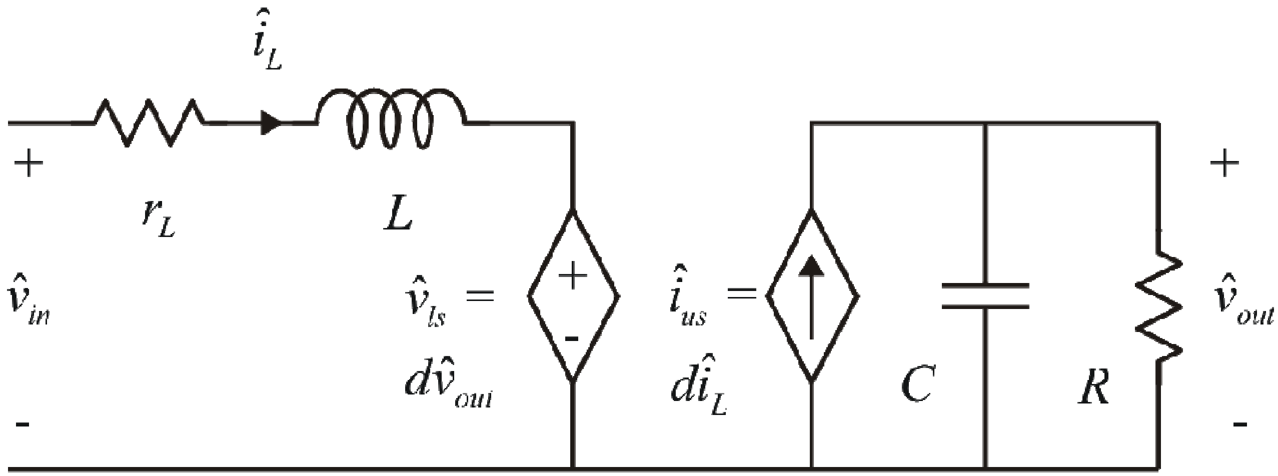
Aspects of Assignment

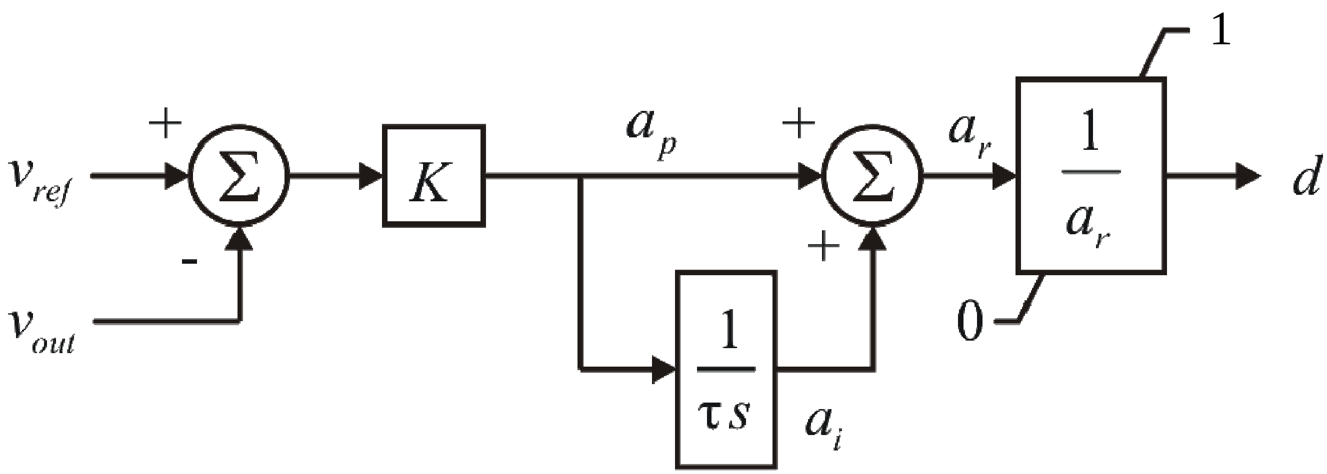
- Problem 1: Add a controller to our dc/dc converter
- Problem 2: Code a trapezoidal predictor corrector routine and use in our system.
- Problem 2+: If you have already had ECE633, also work the following. Consider the scalar system

$$px = ax + bu$$

Assuming a<0, and u is constant, find a bound on h when using the 4th order Runga-Kutta algorithm. It should be in terms of a.

DCDC Converter and Control





$L = 5\,\text{mH}$

$r_L = 100\,\text{m}\Omega$

$C = 100\,\mu\text{F}$

$K = 10^{-3}\,\text{V}^{-1}$

$\tau = 10^{-3}\,\text{s}$

Problem 1

- Given
  - odefsrk.m (integration routine)
  - regdcdc_rksim.m (processing script)
- Write regulated dc/dc converter model
  - regdcdc.m
- Deliverables
  - regdcdc.m (in printout)
  - Plots of run including inductor current, output voltage, duty cycle (in printout).
  - Do not change scales for plots. If you need to do this, your answer is incorrect!

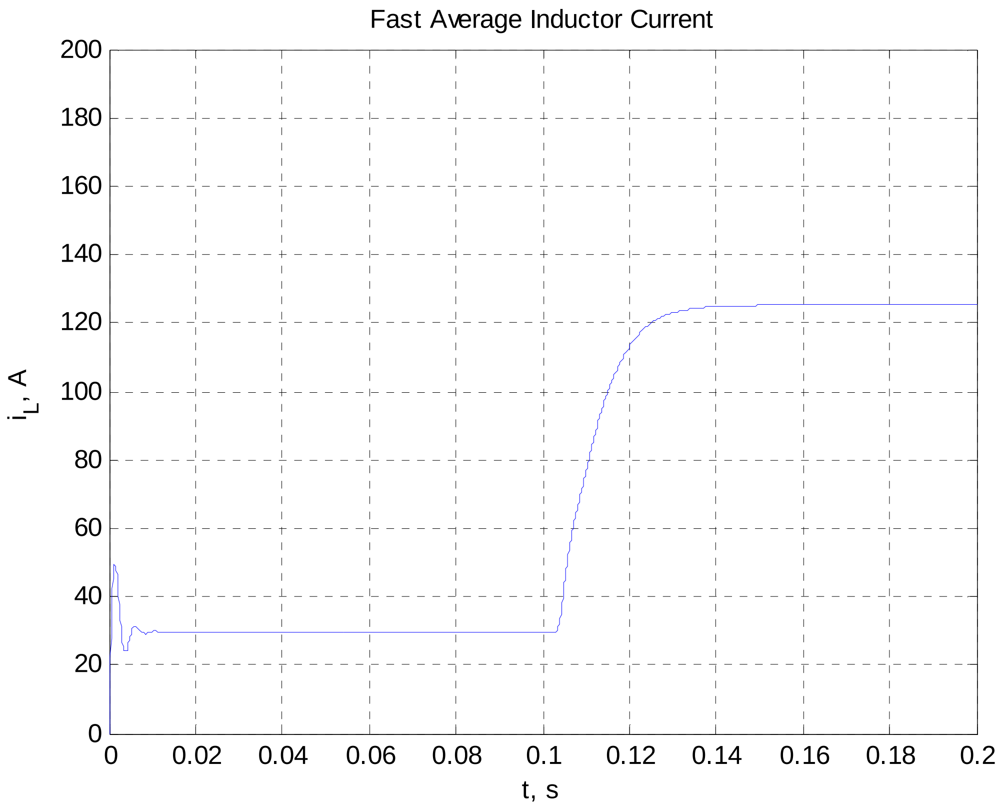
regdcdc.m

```
function [varargout]=regdcdc(t,x,P)
% This routine contains the dynamics of an simple regulated dc/dc converter
% model
%
% [px] = regdcdc(t,x,P);
% [il,vout,vls,ius,iout,d] = regdcdc(t,x,P);
%
% Inputs:
%
% t        = time (s)
% x        = state vector
%  x(1)    = fast average of inductor current (A)
%  x(2)    = fast average of output voltage (V)
%  x(3)    = fast average of integrator output ()
% P        = structure with parameters
%  P.L     = inductor inductance (H)
%  P.rL    = inductor series resistance (Ohms)
%  P.C     = capacitor capacitance (F)
%  P.R     = load resistance (Ohms)
```

regdcdc.m (continued)

```
%  P.vin   = input voltage (V)
%  P.vref0 = initial reference voltage (V)
%  P.vref1 = final reference voltage (V)
%  P.tref  = time for switch in reference voltage (s)
%  P.K     = controller gain
%  P.tau   = controller time constant (s)
%
% Outputs:
%
% px       = time derivative of state vector
%  px(1)   = time derivative of fast average of inductor current (A)
%  px(2)   = time derivative of fast average of output voltage (V)
%  il      = fast average inductor current (A)
%  vout    = fast average output voltage (V)
%  vls     = fast average lower switch voltage (V)
%  ius     = fast average current out of upper switch (A)
%  d       = duty cycle
```

Problem 1 – Sample Run



Problem 2

- Given
  - regdcdc_tpcsim.m (simulation script)
  - regdcdc.m (model dynamics from problem 1)
- Write
  - odetpc.m (trapezoidal predictor-corrector algorithm)
- Deliverables
  - printout of odetpc.m
  - printout of study (inductor current, output voltage, duty cycle)

odetpc.m

```
function [t,y]=odetpc(fhandle,tspan,yic,par,SP)
% This routine solves a ordinary differential equation
% using a trapezoidal predictor corrector method.
%
% [t,y] = odetpc(fhandle,tspan,yic,par,maxt);
%
% Inputs:
%
% fhandle  = a handle to the function whose output is the time derivative
%            of the system model.  The inputs to this function are time,
%            state, and parameter vales.
% tspan    = a vector whose elements describe at which point in time the
%            solution is sought
% yic      = a vector which describes the initial condition of the system
%            being simulated
% par      = a structure which cointains data or parameters needed to
%            evaluate the time derivative of the state variables
% SP       = structure of simulation algorithm parameters
%  SP.maxt = the maximum allowed time step (s)
%  SP.maxit= maximum allowed iterations
%  SP.maxre= maximum relative error in state variable
%  SP.maxae= maximum absolute error in state variable
%
```

odetpc.m continued

```
% Outputs:
%
% t        = a vector of times at which the state vector has been found
% y        = a matrix wherein each row contains the state vector at a
%            given time.  Each column is the time history of a particular
%            state
```