Time Domain Simulation and Optimization for Design

Homework Set 2

S.D. Sudhoff

Spring 2019

Problem 5: Overview

- Code a trapezoidal predictor corrector routine and use it to determine whether the earth survives a near-earth asteroid event.
- Scenario: You will be simulating the interactions between the sun, the earth, and asteroid of one ten-millionth the earth's mass (all other objects are neglected). Your simulation will start on January 3, 2018 with the earth at perihelion. Initial conditions are given in an attached data file, but the asteroid is in the plane of earth's orbit facilitating a 2-D simulation. **The question: will the earth survive?**

Problem 5: Specifications

- Given
  - earth_asteroid_model.m (dynamic model)
  - earth_asteroid_simulation.m (simulation script)
- Write a trapezoidal predictor-corrector routine
  - odetpc.m (I/O for this routine must be as follows)

Problem 5: 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
%
```

Problem 5: odetpc.m continued

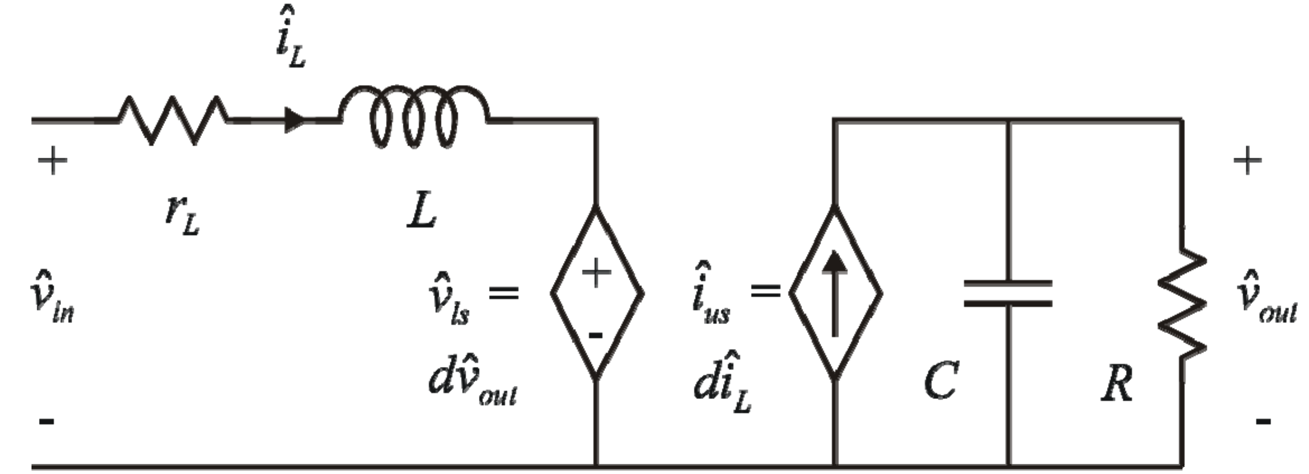
```
% Outputs:
%
% t        = a vector of times at which the state vector has been found
% y        = a matrix wherein each row contains the state vector at a
%            given time.  Each column is the time history of a particular
%            state
```

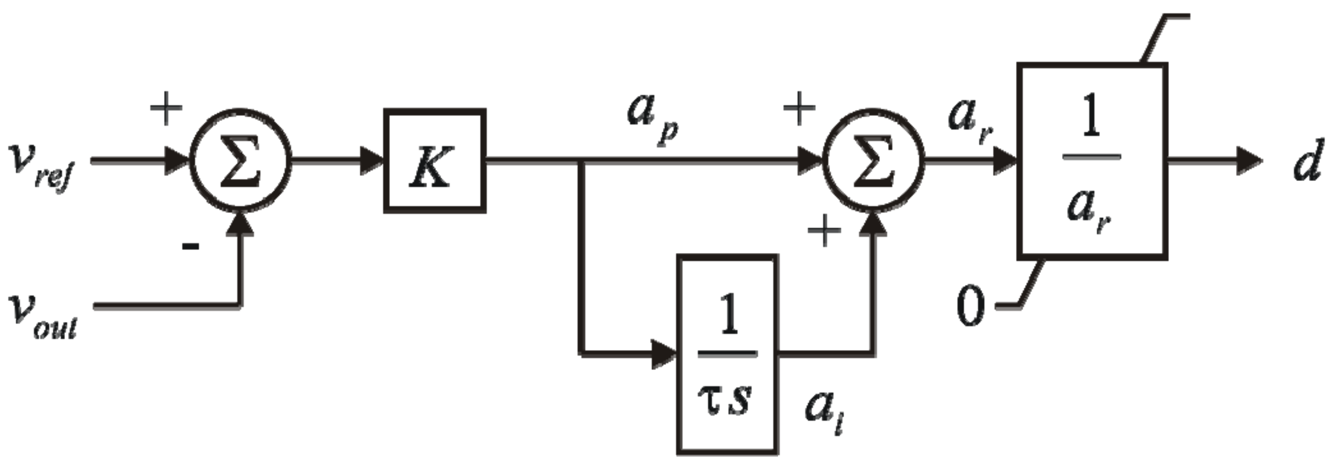
Problem 5: Deliverables

- Deliverables
  - odetpc.m (code listing; well-documented) submitted with filename ‘odetpcGeorgeAiry.m’
  - Code listing of odetpc.m with filename ‘odetpcGeorgeAiry.pdf’
  - Code will be graded based on
    - Numerical accuracy
    - Computational efficiency
    - Documentation

Problem 6

- Given the files for the Simulink model add a controller to our dc/dc converter





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

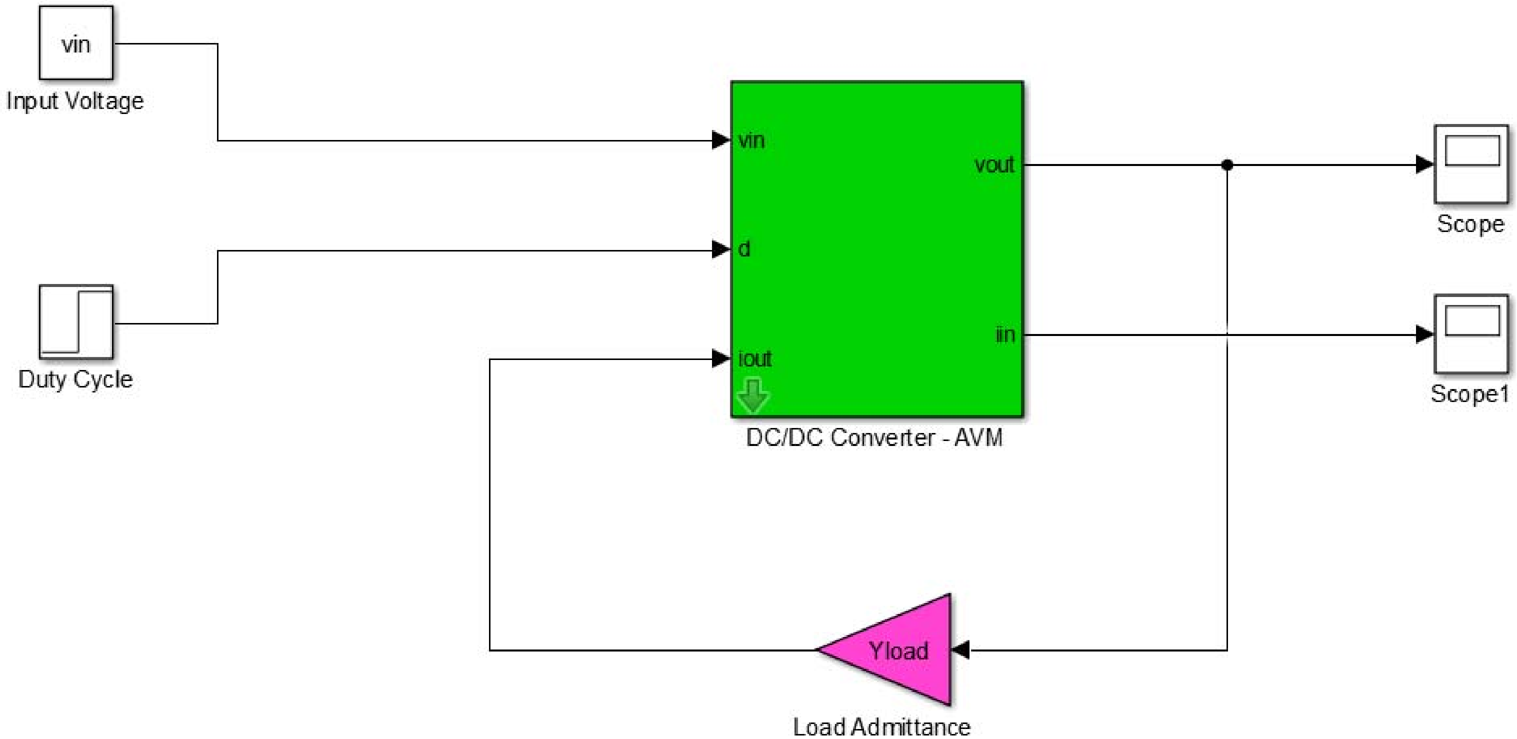
$r_L = 500\,\mathrm{m\Omega}$

$C = 1000\,\mu\mathrm{F}$

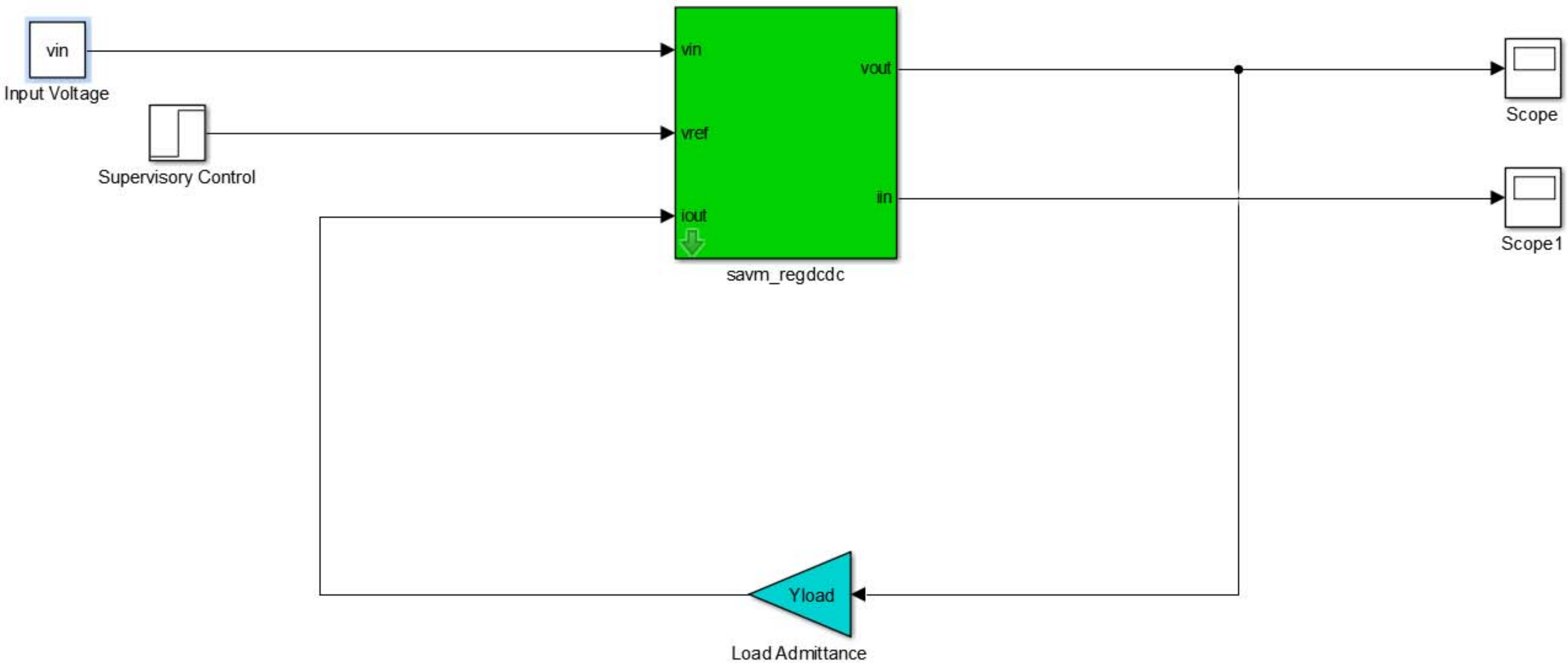
$K = 10^{-3}\,\mathrm{V}^{-1}$

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

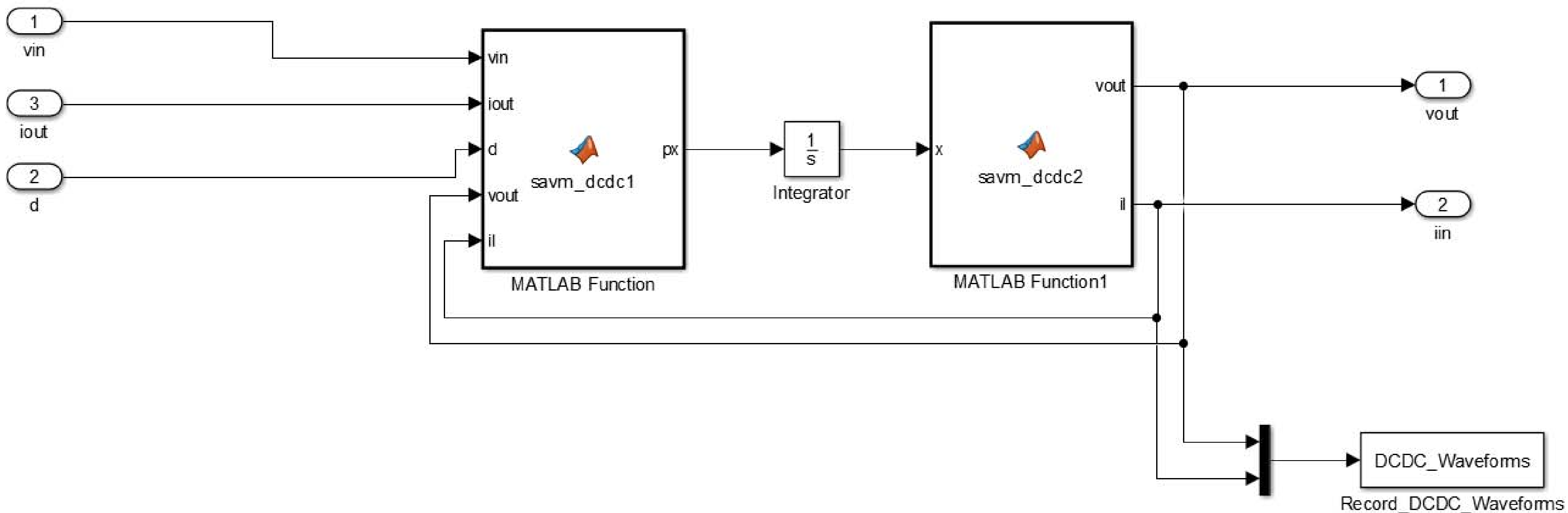
Problem 6: Starting Point



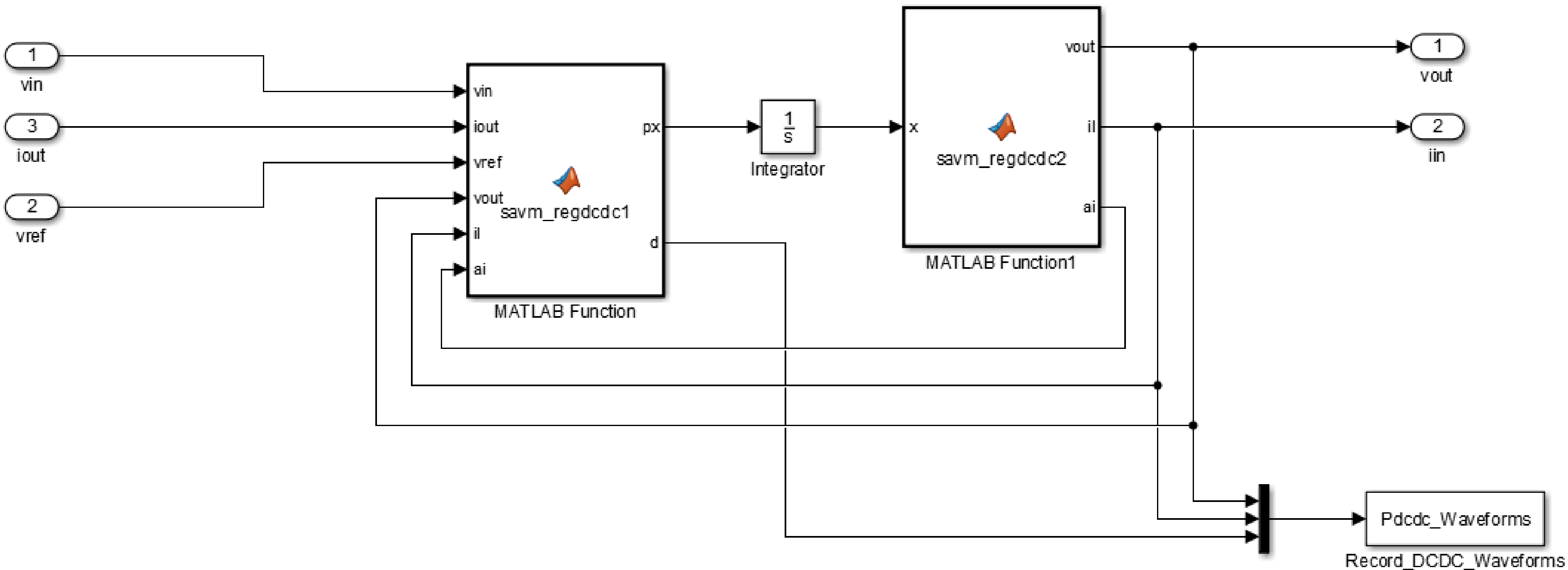
Problem 6: Ending Point



Problem 6: Starting Point



Problem 6: Ending Point



Problem 6

- Print relevant files/scripts/Simulink diagrams.
- Plot waveforms on the following scale
  - Output voltage versus time, v_{out}, 0 to 1000 V
  - Duty cycle versus time, d, 0 to 1
  - Inductor current versus time, i_L, 0 to 200 A

  - Time, t, should go from 0 to 0.5 s on all plots.

- Deliver one file Probem6CarolineHershel.pdf

Problem 7

- Consider Problem 6. Suppose the inductance and capacitance are all subject to +/- 20% random and uncorrelated variations. Does the control still work? Use brute force with at least 50 simulation runs to form a conclusion.

- Print all scripts and diagrams that are different from those of Problem 6. Provide relevant plots and conclusion.

- Deliver one file Probem7WilliamHershel.pdf

Problem 7: Code Snip

```
for i=1:50

   % run the study
   sim('savm_regdcdc');

   Pdcdc.L=Lmin+(Lmax-Lmin)*rand;                % inductor inductance (H)
   Pdcdc.C=Cmin+(Cmax-Cmin)*rand;                % capacitor capacitance (F)

   % plot the results
   savm_regdcdc_plot('DCDC Converter',Pdcdc_Waveforms,10);
   figure(10)
   ylim([0 1000]);
   hold on
   figure(11)
   ylim([0 200]);
   hold on
   figure(12)
   ylim([0 1])
   hold on

end
```

Problem 8

- 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.

- Answer should be in a file of the form Problem8VeraRubin.pdf