D1

Boost Converter Design Project: Modelling and Simulation

In this project you will build a power supply that boosts the voltage provided by a single AA-battery to up to 12 V. You will develop a C-program that runs on `Il Matto` and controls the output voltage through pulse-width-modulation. You will also develop a second C-program that runs on a host PC and interfaces with your embedded program on `Il Matto` by serial communication. The user-interface of the hosted C-program will allow for setting a desired output voltage and for displaying the current output voltage. After building your computer-controlled power supply you will evaluate its performance. This document details the initial modelling and simulation section of the project. [1]

Schedule:	
Preparation time :	5 hours
Lab time :	0 hours
Items provided:	
Tools :	n/a
Components :	n/a
Equipment :	
n/a	
Software :	MATLAB
Items to bring:	
Wet matter:	Brain.

Academic Integrity – *If you undertake the preparation jointly with other students, it is important that you acknowledge this fact in your logbook. Similarly, you may want to use sources from the internet or books to help answer some of the questions. Again, record any sources in your logbook.*

[1] **Revision History**
January 2nd 2014 mcf Version 1

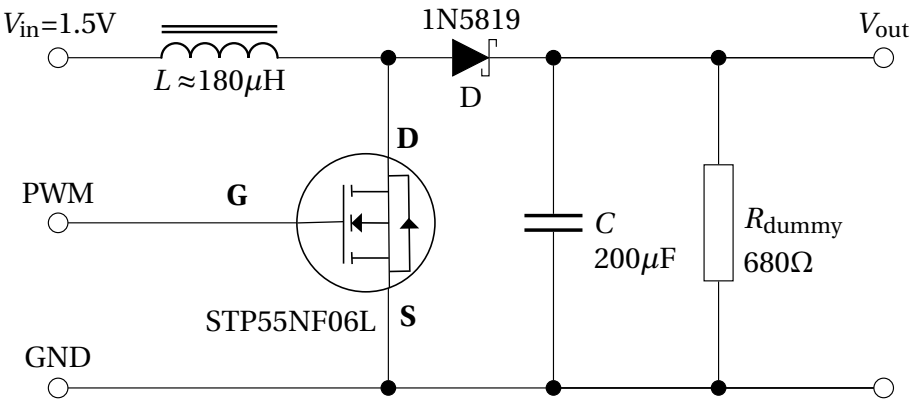

Figure 1: The boost converter

1 Aims, Learning Outcomes and Outline

This section of the design exercise aims to:

- Understand the principles of switch-mode voltage conversion.
- Use circuit theory to develop an idealised model of a switch mode power supply.
- Use MATLAB to develop a hybrid simulation of a boost converter.

The remainder of the design exercise aims to:

- Design and implement an embedded system that is controlled by a PC1 through serial communication.
- Use a microcontroller to continuously monitor an analogue input signal and respond with appropriate output in real-time.
- Utilize a simulation model to inform hardware closed loop control design.
- Implement closed-loop control-schemes with a microcontroller and evaluate their performance.

---

2 Preparation

This document details the preparation for the D1 design exercise. You should expect to complete this preparation in about 5 hours prior to coming to the first lab D1 session, where this preparation will be marked against the standard first year lab marks scheme (excluding the preparation category). Your preparation should be recorded in your lab book.

3 Introduction

The boost converter circuit is shown in Figure 1. It's circuit idealisation is shown in Figure 2. The boost converter is readily and accurately analysed by the ideal components.

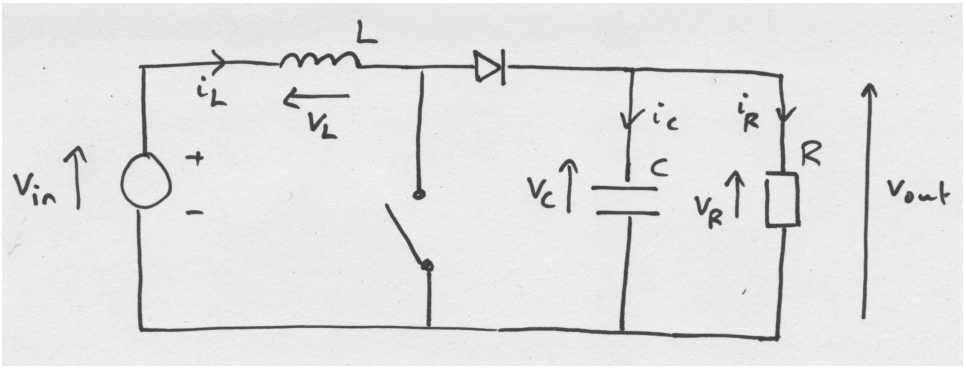

Figure 2: The boost converter idealisation

1. The PWM modulated transistor is represented by a switch which is closed when the PWM output is `high` and open when the PWM signal is `low`.
2. The diode, capacitor, inductor and resistor are represented by their idealisations.

3.1 Static PWM driven switch

The switch is driven by PWM, and is such that the switch is closed when the PWM signal is high and open when the PWM signal is low. We define the PWM signal:

$$\text{pwm}(t,T,d) = \begin{cases} 1 & \text{if } kT \le t < (k+d)T \\ 0 & \text{if } (k+d)T \le t < (k+1)T \end{cases} \tag{1}$$

where t is time, T is the period of the pulse waveform, $0 < d < 1$ is the duty cycle (the proportion of time spent at signal level 1).

- In MATLAB, define a `function v=pwm(t,T,d)` (recall this needs to be in an m-file called `pwm.m`) which represents the above signal. You might find it useful to investigate (`help`) the `mod` operation. Use MATLAB to plot a graph of `pwm(t,T,d)` for a suitable range of t and values of T, d and indicate T and d on the plot.

3.2 Explanation of how a boost converter works

The key to understanding how a boost converter works is to consider steady state operation, and assuming the PWM switching is fast enough that the inductor current never drops to zero. Further assume that the output voltage v_{out} at steady state is constant and larger than v_{in} (this approximately correct if the time constant RC is such that the capacitor does not significantly discharge during the switch off part of the cycle). In that case, the idealised behaviour is straightforward to analyse. The key is to consider the current i_L flowing in the inductor.

1. When the switch is closed, see Figure 3, the inductor current increases due to the voltage from the source:
$$L\frac{di_L}{dt} = v_{in}. \tag{2}$$
Since per cycle of period T, the switch is closed for a time dT, it follows that the change in current i_L whilst the switch is closed is given by:
$$\Delta i_L = dT\frac{v_{\text{in}}}{L}. \tag{3}$$

2. When the switch is open, see Figure 4, the inductor current decreases:

$$L\frac{di_L}{dt} = v_{in} - v_{out}, \tag{4}$$

Since per cycle of period T, the switch is closed for a time $(1-d)T$, it follows that the change in current i_L whilst the switch is closed is given by:

$$\Delta i_L = (1-d)T\frac{v_{\text{in}} - v_{\text{out}}}{L}. \tag{5}$$

In the steady state, the current at the end of the cycle (of period T) will be the same as at the start, so, by equations 3, 5 it follows that:

$$dT\frac{v_{\text{in}}}{L} + (1-d)T\frac{v_{\text{in}} - v_{\text{out}}}{L} = 0 \tag{6}$$

Rearranging we get:

$$v_{\text{out}} = \frac{1}{1-d}v_{\text{in}}, \tag{7}$$

so it can be seen that $v_{\text{out}} > v_{\text{in}}$ since $0 < d < 1$ as consistent with the starting assumption. Further, as $d \to 1$, the output voltage becomes arbitrarily large.

- All this is under the assumption that the inductor current does not drop to zero (the so-called *continuous mode*). If the inductor current does fall to zero during discharging (*discontinuous mode*), then the diode becomes reverse biased, and the inductor current remains at zero for the remainder of the duty cycle. A similar analysis applies, but now v_{out} depends also on the circuit component values, the switching frequency etc. Read the wikipedia entry on boost converters:

  `http://en.wikipedia.org/wiki/Boost_converter`

  and derive the resulting expression for v_{out} in your logbook.

4 Static Simluator

This section will lead you to the construction of a simulator in MATLAB which will enable you to verify the above assumptions, and see how the circuit performs in transient conditions and in the discontinuous mode.

4.1 Model when switch is closed

When the switch is closed, the circuit splits into an RC circuit, and an inductor driven by the source (see Figure 3), giving:

$$L\frac{di_L}{dt} = v_{in}, \tag{8}$$

and

$$C\frac{dv_{out}}{dt} = C\frac{dv_C}{dt} = i_C = -i_R = -\frac{V_R}{R} = -\frac{v_{out}}{R}. \tag{9}$$

- Write this in state space form:

$$\frac{d}{dt}\begin{bmatrix} i_L \\ v_{out} \end{bmatrix} = \underbrace{\begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \end{bmatrix}}_{A_{\text{closed}}}\begin{bmatrix} i_L \\ v_{out} \end{bmatrix} + \underbrace{\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}}_{B_{\text{closed}}} v_{in}. \tag{10}$$

  where the 2×2 matrix A and the 2×1 matrix B depend on R, L and C.

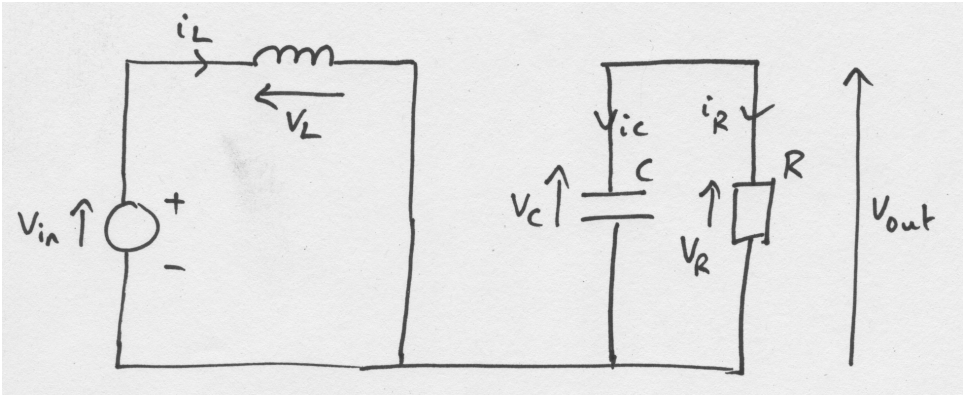

Figure 3: Closed switch

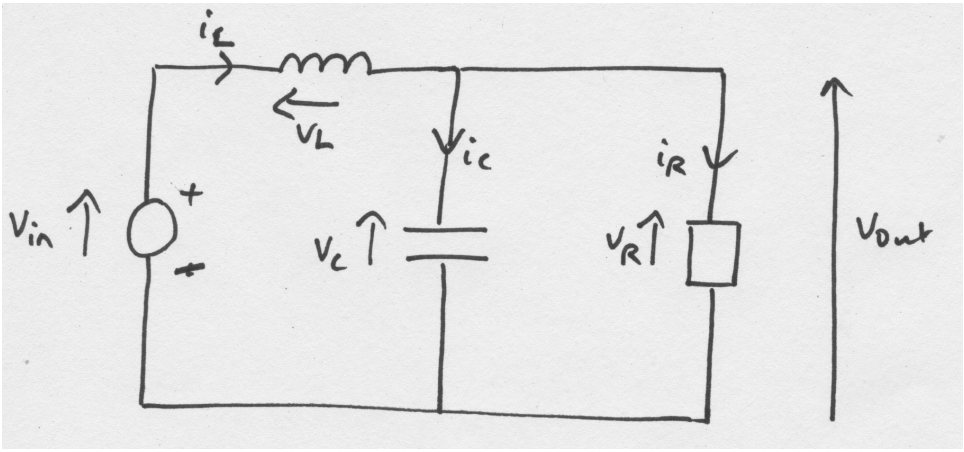

Figure 4: Open switch

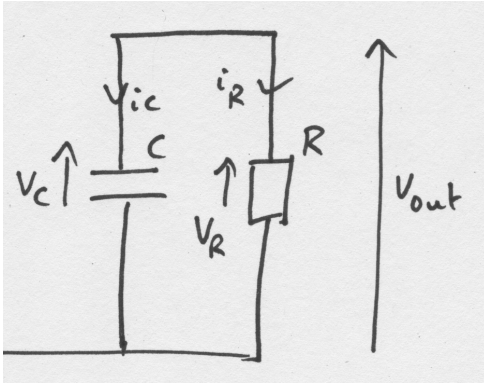

Figure 5: Open switch

4.2 Model when switch is open and diode in forward region

When the switch is open and $i_L > 0$, the diode is in the forward region and is replaced by a short circuit and the circuit is RLC (Figure 4), giving:

$$L\frac{di_L}{dt} = v_{in} - v_{out}, \tag{11}$$

and

$$C\frac{dv_{out}}{dt} = C\frac{dv_C}{dt} = i_C = i_L - i_R = i_L - \frac{V_R}{R} = i_L - \frac{v_{out}}{R}. \tag{12}$$

- Write this in state space form:

$$\frac{d}{dt}\begin{bmatrix} i_L \\ v_{out} \end{bmatrix} = \underbrace{\begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \end{bmatrix}}_{A_{\text{open,forward}}} \begin{bmatrix} i_L \\ v_{out} \end{bmatrix} + \underbrace{\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}}_{B_{\text{open,forward}}} v_{in}. \tag{13}$$

  where the 2×2 matrix A and the 2×1 matrix B depend on R, L and C.

4.3 Model when switch is open and diode in reverse region

When the switch is open and the diode is in the reverse region, the diode is replaced by an open circuit, and the inductor and the source are disconnected (switch is open, and diode in reverse region). No current can flow through the inductor:

$$\frac{di_L}{dt} = 0, \tag{14}$$

The remaining circuit is RC (Figure 4), giving:

$$C\frac{dv_{out}}{dt} = C\frac{dv_C}{dt} = i_C = -i_R = i - \frac{V_R}{R} = -\frac{v_{out}}{R}. \tag{15}$$

- Write this in state space form:

$$\frac{d}{dt}\begin{bmatrix} i_L \\ v_{out} \end{bmatrix} = \underbrace{\begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \end{bmatrix}}_{A_{\text{open,reverse}}} \begin{bmatrix} i_L \\ v_{out} \end{bmatrix} + \underbrace{\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}}_{B_{\text{open,reverse}}} v_{in}. \tag{16}$$

  where the 2×2 matrix A and the 2×1 matrix B depend on R, L and C.

4.4 Model including the switch

At the times when the switch opens and closes the system model changes from being described by $(A_{\text{closed}}, B_{\text{closed}})$ to being described by $(A_{\text{open,forward}}, B_{\text{open,forward}})$ or $(A_{\text{open,reverse}}, B_{\text{open,reverse}})$. In turn the whether the diode is in the forward or reverse region determines the which of the 'open' models is relevant. We need to consider what the initial values of i_L and v_{out} are at the time of switching.

Since the current i_L in an inductor cannot change instantaneously, i_L takes the value immediately after the switch that it had immediately before the switch. Similarly since the voltage in a capacitor cannot change instantaneously, v_C takes the value immediately after the switch that it had immediately before the switch. Further, when the (ideal) diode moves from the forward to reverse region and vice versa, i_L and v_C have no instantaneous jumps in value.

That is both i_L and $v_C = v_{\text{out}}$ are continuous, and hence the complete system is described by:

$$\frac{d}{dt}\begin{bmatrix} i_L(t) \\ v_{out}(t) \end{bmatrix} = A(t)\begin{bmatrix} i_L(t) \\ v_{out}(t) \end{bmatrix} + B(t)v_{in}(t). \tag{17}$$

where

$$A(t) = \begin{cases} A_{\text{closed}} & \text{if } \text{pwm}(t,T,d) = 1 \\ A_{\text{open,forward}} & \text{if } \text{pwm}(t,T,d) = 0 \text{ and } i_L(t) > 0 \\ A_{\text{open,reverse}} & \text{if } \text{pwm}(t,T,d) = 0 \text{ and } i_L(t) \leq 0 \end{cases} \tag{18}$$

$$B(t) = \begin{cases} B_{\text{closed}} & \text{if } \text{pwm}(t,T,d) = 1 \\ B_{\text{open,forward}} & \text{if } \text{pwm}(t,T,d) = 0 \text{ and } i_L(t) > 0 \\ B_{\text{open,reverse}} & \text{if } \text{pwm}(t,T,d) = 0 \text{ and } i_L(t) \leq 0 \end{cases} \tag{19}$$

NB. by means of a check, you should have that $B(t) = B_{\text{open,forward}} = B_{\text{closed}}$ and $B_{\text{open,reverse}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

4.5 Simulation model

Let the state vector $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} i_L \\ v_{\text{out}} \end{bmatrix}$, and hence complete the code below, to define the time-varying differential equation (17)- (19):

```
function dx=boost(t,x)
global v_in R C L T d
dx=zeros(2,1);
if pwm(t,T,d)==1
    dx(1) =
    dx(2) =
end
if pwm(t,T,d)==0 && x(1)>=0
    dx(1) =
    dx(2) =
end
if pwm(t,T,d)==0 && x(1) <0
    dx(1) =
    dx(2) =
end
```

Save this as the file `boost.m`. There is a subtle point about calling simulating this function that we have not dealt with in other simulations for ELEC1200. It is necessary to set the maximum time step size in the integration routine. If you don't do this, then the simulation can 'jump' over a number of duty cycles in a single update, and not 'see' the detail that makes the circuit work. Hence we need simulate using e.g.

```
options=odeset('MaxStep', 1e-5);
[t,x]= ode23(@boost2,[0:1e-7:0.2], [0 0], options);
```

You may need to adjust the max. step size, always review your simulation results for plausibility. Note that the time interval `[0:1e-7:0.2]` is chosen to ensure that the output of the simulations are reported a sufficiently high frequency for accurate graphing.

- Run the simulation using `ode23` from another script, where v_{in}, R, C, L take the circuit values:

Parameter	Value
v_{in}	$1.5V$
R	680Ω
C	$200\mu F$
L	$180\mu H$

  Start at a PWM frequency of $10KHz$, and with $d = 0.1$. Make sure that you declare v_{in}, R, C, L, T, d to be `global` in that script (so that the `boost` function sees these globally defined values).

- Experiment with values of $0 < d < 1$ and periods T. On a single plot with suitable time axes, produce graphs of i_L, v_{out} and pwm against time t.

- The diode can take a maximum current of 1A. What is the best steady state performance (i.e. the highest steady state value of v_{out}) you can achieve for a choice of d and T if the diode current is restricted to 200mA? Evidence your claim. Make sure that the behaviour at steady state is visible, at the level of the individual pulses.

- Look at the steady state behaviour (i.e. after the transient effects have gone). Are the assumptions/approximations made in Section 3.2 valid? Graph $i_L(t)$ and the PWM signal during *continuous* and *discontinuous* modes of operation (see Section 3.2).

5 Dynamic Simulator

A boost converter can achieve better results still if the PWM signal is regulated via feedback. The purpose of this section is to build a simulator which allows simple closed loop control strategies to be implemented. Later in the exercise, you should use this code to trial control strategies for implementation on the `Il Matto`.

Via ADC, the controller is supplied a measurement of v_{out} at discrete time steps. The controller then adjusts the duty cycle d of the PWM signal to achieve a desired performance.

5.1 Discretisation of the Circuit Model

Since the control strategy will be implemented digitally on the `Il Matto`, the simulator we construct in this section will be entirely implemented in discrete time. That is, rather than utilise `ode23` to integrate the differential equations, we will first discretise the equations ourselves and then implement the fully discretised system (including the controller) in MATLAB.

The discretisation utilised is the Euler scheme: and is based on the approximation of a derivative:

$$\frac{df}{dt} \approx \frac{f(t+h) - f(t)}{h}.$$

We will simulate in time steps of period h. Applied to equation (17) at time step $k \in \mathbb{N}$, i.e. $t = kh$, we therefore have:

$$\frac{1}{h}\begin{bmatrix} i_L((k+1)h) - i_L(kh) \\ v_{\text{out}}((k+1)h) - v_{\text{out}}(kh) \end{bmatrix} = A(kh)\begin{bmatrix} i_L(kh) \\ v_{out}(kh) \end{bmatrix} + B(kh)v_{in}(kh). \tag{20}$$

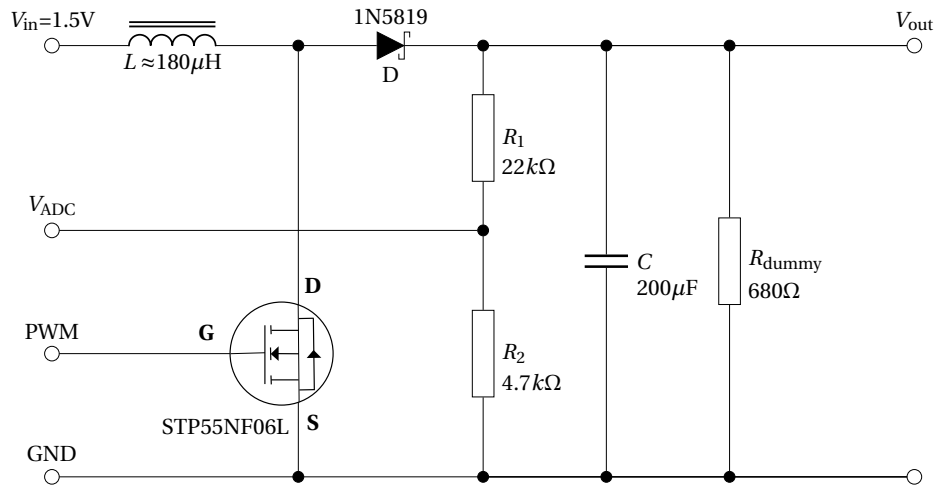

Figure 6: V_{adc} measurement.

which implies:

$$\begin{bmatrix} i_L((k+1)h) \\ v_{\text{out}}((k+1)h) \end{bmatrix} = \begin{bmatrix} i_L(kh) \\ v_{out}(kh) \end{bmatrix} + hA(kh) \begin{bmatrix} i_L(kh) \\ v_{out}(kh) \end{bmatrix} + hB(kh)v_{in}(kh), \tag{21}$$

$$= (I + hA(kh)) \begin{bmatrix} i_L(kh) \\ v_{out}(kh) \end{bmatrix} + hB(kh)v_{in}(kh). \tag{22}$$

5.2 Discrete controller

The controller will adjust the duty cycle d depending on measurements of v_{out}. This section describes the modelling and simulation of this feedback loop. This controller will be implemented on the `Il Matto` based on the voltage level being read in discrete steps via the ADC channel.

A voltage divider is used to take measurements of v_{out} at an appropriate voltage level for the ADC channel of the Il Matto. From figure 6, we have

$$v_{\text{adc}} = \gamma v_{\text{out}}, \quad \text{where } \gamma = \frac{R_2}{R_1 + R_2} = 0.176 \tag{23}$$

and $R_1 = 22\text{k}\Omega$, $R_2 = 4.7\text{k}\Omega$.

The analogue voltage is sampled at frequency f_{control}, where the period $\tau = \frac{1}{f_{\text{control}}}$ is an integer multiple of the PWM period T, so that there are a fixed number of PWM cycles between controlled adjustments of the duty cycle d. The duty cycle d then becomes time-varying, updated at $t = k\tau$ $k \in \mathbb{N}$ and held constant in between. Here we illustrate this with a so-called *proportional controller* which adjusts the duty cycle according to the 'error' between v_{adc} and a desired value r, (say $r = 10V$). The amount the duty cycle is changed at time step $k \in \mathbb{N}$ is determined by the proportional gain $k_p > 0$.

$$d(k\tau) = d((k-1)\tau) - k_p(v_{\text{adc}}(k\tau)/\gamma - r). \tag{24}$$

- The load resistance now includes the voltage divider. Determine the percentage change in load resistance between the circuits in Figure 1 and 6. Is this significant?
- Examine the supplied code in the Appendix. Explain what each section of the code does in the light of the above explanations to demonstrate you clearly understand how the method and the code works.

- Add in the omitted data, e.g. the state space matrices etc, and run the code to explore the effect of changing different values of k_p and the voltage setpoint r. Document your findings.
- This simulation method produces small deviations of i_L below zero. Why does this differ from the simulation you created in Section 4.5, using `ode23`? Is it reasonable to argue that this doesn't matter too much?

Now you have two working simulators and a good understanding of how a boost converter works. Use the simulators in conjunction with your coding and circuit construction in the lab. to inform your design. The dynamic simulator will be useful to trial control strategies before implementation, and to tune up the parameters for best performance. You should find the simulators primarily useful as qualitative guides.

5.3 Optional Additional work

The above discrete time closed loop simulator has been set up with a proportional (P) controller.

- Investigate PID controllers, and implement a PID control design.
- Implement a current limiter in the code, i.e. replicate the effect e.g. a 200mA power supply limit.

A Discrete simulator code

```
%Circuit data
v_in=
R=
C=
L=

%Voltage divider
R_1=
R_2=
gamma=R_2/(R_1+R_2);

%Total Resistance:
R_T=1/(1/R+1/(R_1+R_2));

%Iniital duty cycle:
d=0.8;

%system matrices:
A_closed=
A_open_forward=
A_open_reverse=
B_closed=
B_open_forward=
B_open_reverse=

%pwm frequency:
f_pwm = 10e3;
T = 1/f_pwm;

%simulation sample length:
h = T/1e3;

%control update sample length:
f_control =  f_pwm/10;
tau = 1/f_control;

%Simulation paramters:
final_time = 1;

%Initialisation
x=zeros(2,round(final_time/h));
duty_cycle=zeros(1,round(final_time/h));

I=[1 0; 0 1];

% x(1) = i_L
% x(2) = v_out

%initial condition
x(:,1)=[0 0.5];
```

```
%control setpoint
r=10;

%controller gains
k_p=0.0015;

for t=2*h:h:final_time

    % measured voltage via voltage divider:
    v_adc = gamma* x(2,round(t/h));
    % voltage error to desired setpoint value r:
    v_error=v_adc/gamma-r;

    %duty cycle update:
      if rem(t,tau)==0
      d=d-k_p*(v_error);
  end
      if pwm(t,T,d)==1
        A=A_closed;
        B=B_closed;
    end
    if pwm(t,T,d)==0 && x(1,round(t/h))>0
            A=A_open_forward;
            B=B_open_forward;
    end
    if pwm(t,T,d)==0 && x(1,round(t/h)) <=0
        A=A_open_reverse;
        B=B_open_reverse;
    end
    x(:,round(t/h) +1)= (I+h*A)*x(:,round(t/h)) + v_in*h*B;
    duty_cycle(round(t/h) +1)= d;
end

t= [2*h:h:final_time+2*h];
plot(t,x)
hold on
 plot(t,pwm(t,T,duty_cycle),'r')
 plot(t,duty_cycle,'m')
 xlabel('Time (s)','FontSize',16)
 ylabel('Voltage (V), Current (A) ','FontSize',16)
 legend('i_L','v_{\rm out}','pwm')
hold off
```