

ECE317 Laboratory

Classical Control

Richard Tymerski
Portland State University
Department of Electrical and Computer Engineering
Portland, Oregon, USA

Contents

1	Introduction to the Labs	1
2	Lab 1	5
2.1	Objectives	5
2.2	Circuit #1	5
2.2.1	Tasks	6
2.3	Circuit #2	7
2.3.1	Tasks	8
2.4	Circuit #3	8
2.4.1	Tasks	9
3	Lab 2	11
3.1	Objectives	11
3.2	Background	11
3.3	Tasks	14
3.3.1	PECS simulation	14
3.3.2	Matlab simulation – results check	15
4	Lab 3	19
4.1	Objectives	19
4.2	Background	19
4.3	System Transfer Functions	22
4.4	Tasks	26
4.5	Note	33
4.6	Results	34
4.7	Postscript	34
5	Lab 4	37
5.1	Objectives	37
5.2	Background	37
5.3	Tasks	40
5.4	Results	42
5.5	Notes - Buck converter components:	44

6 Lab 5	45
6.1 Objectives	45
6.2 Background	45
6.3 Tasks	49
6.3.1 Pre-Lab	49
6.3.2 In the Lab	52
6.3.3 Post-Lab	53
6.4 Optional Tasks - Alternative compensators	54
6.4.1 Proportional Control	54
6.4.2 Lead Control	54
6.5 Note	55
7 Lab 6	57
7.1 Objectives	57
7.2 Background	58
7.3 Tasks	59
7.3.1 Pre-Lab	59
7.3.2 In the Lab	61
7.3.3 Post-Lab	62
A List of Parts	65
B PECS	67
B.1 PECS Overall Description	67
B.2 PECS Usage	67
B.2.1 Building a circuit schematic in PECS	68
B.2.2 Setting simulation parameters	70
B.2.3 Running the simulation	71
B.2.4 Selecting the desired output(s) to plot	71
B.3 Elements	73
B.3.1 Basic passive elements: R (resistor), C (capacitor) and L (inductor)	73
B.3.2 Sources: VDC (DC voltage), IDC (DC current), VAC (AC voltage)	74
B.3.3 Ports: Vport (voltage port) and Iport (current port)	77
B.3.4 Switches: Sw (controlled switch) and D (diode)	78
B.3.5 Switch control elements: Clk (Clock) and (MOD) PWM Modulator	79
B.3.6 Miscellaneous: OPAMP (operational amplifier), current sensor (no symbol name), ground node (no symbol name)	82
B.4 PECS PLOT	84
B.5 Terminology:	84
B.6 General use	84
B.7 Menu items	85
B.7.1 File menu commands	86
B.7.2 Edit menu commands	87

CONTENTS

v

B.7.3	Plots Menu commands	87
B.7.4	View menu commands	89
B.7.5	Options menu commands	89
B.7.6	Help menu commands	89
B.8	References	90

Chapter 1

Introduction to the Labs

Classical control design is generally undertaken in the frequency domain as opposed to the time domain. Examination of a system in the frequency domain allows for simpler determination of stability, through the use of phase and gain margin metrics.

The classical control design methodology requires that a system model in the form a transfer function be available. This is complicated by the fact that often a suitable model may not be apparent. The following set of labs starts with a simple system consisting of a single-pole, double-throw switch followed by an LC filter, proceeds through the modelling of the system and subsequently to the design of an effective controller. The purpose of this system is to convert a dc voltage at a higher level to a lower level while achieving a high power efficiency, typically in the 90% range. The system examined is known a Buck dc-to-dc power converter and is widely used in industry. The endgame is to enclose this system with negative feedback to produce an effective voltage regulation, in the face of input voltage and output load variations, and so an effective controller needs to be designed.

The apparent simplicity of the buck converter belies the modeling challenges that can be brought to light. The function of the switch in the system is to produce pulses whose width can be varied. The control of the pulse width is undertaken by a pulse width modulator (PWM). This is inherently a nonlinear device since a sinusoidal input produces a sinusoidally modulated pulse train which contains spectral components not present in the input. In this case, a so-called *describing function* analysis is used in modelling the system. This approach determines the magnitude and phase of the spectral component in the output which is at the same frequency as the sinusoidal input.

Furthermore the buck converter itself is a time varying system but from basic considerations we are able to define an average model which features a number of different transfer functions of interest. In essence the system considered has

three inputs and one output (the output voltage). There is one control input and two disturbance inputs, leading to a total of three transfer functions that need to be modelled.

Two sets of software tools are used to help with the design and verification in these labs:

1. Matlab: As compensator design is undertaken in the frequency domain, Bode plots are used to examine magnitude and phase responses of various transfer functions, in particular the loop gain which is important in assessing phase and gain margins. Additionally, Matlab is used to examine time-domain simulations using the derived small-signal transfer function models
2. PECS: This simulator permits circuit level simulation of the physical circuit, rather than just its small-signal model, and so more faithfully represents the operation of the actual system, showing large-signal effects such as ripple voltage.

The software tools will be used to verify the derived transfer functions and, in the process, help to obtain a better understanding of the system. The circuit simulator allows one to see waveforms that closely resemble those that would be observed in a hardware implementation as large signal effects can also be seen in the simulation. Having arrived at transfer function models for a system, these transfer functions are used in the classical control design process and can also be used to produce time domain simulations. Thus it becomes instructive to see how well these time domain simulations conform to the simulations obtained from Matlab resulting from small-signal transfer function models. Becoming further instructive when observing waveforms of the hardware implementation.

It is the aim of these series of labs to start with a simple system, that is widely used, and to go through the whole process from understanding system operation to the end point of reaching a design that incorporates feedback control. A big picture set of aims for this series of labs is the following:

1. Obtain a clear understanding of the buck converter system operation
2. Obtain a suitable model of the system
3. Test the (open loop) system and verify the model.
4. Examine the effectiveness of a simple compensator used in (closed loop) feedback control.
5. Design a more effective and albeit more complicated controller.
6. Test the improved system design and understand how and why the improvement has been achieved.

In terms of the labs themselves, the titles are listed next:

- Lab 1: Introduction to a basic dc-to-dc power conversion system and the PECS circuit simulator
- Lab 2: System Identification of a 2^{nd} Order System through Step Response
- Lab 3: System Identification of the dc-to-dc Buck converter
- Lab 4: Open loop system construction and testing: the Buck dc-to-dc voltage converter
- Lab 5: Closed Loop Feedback System - Analysis: Analysis and performance of a closed loop dc-to-dc Buck converter system
- Lab 6: Closed Loop Feedback System - Design: Design of an effective feedback compensator for a closed loop dc-to-dc converter system

Labs 1 to 3 involve the software tools in helping to understand the system and in the process of system identification i.e. discovering and verifying transfer functions. Labs 4 to 6 involve hardware construction. In particular, Lab. 6 involves controller design using the asymptotic Bode plot construction methodology discussed in class. The design is fully verified using the software tools before implementation.

The labs have been written to be stand-alone. Consequently, sufficient background material has been included in each lab with the aim to provide sufficient understanding required for the lab and hopefully, for the most part, obviating the need to search elsewhere for this information.

Chapter 2

Lab 1

Introduction to a basic dc-to-dc power conversion system and the PECS circuit simulator

2.1 Objectives

1. To introduce a basic dc-to-dc power conversion system.
2. To introduce the PECS simulator which simulates these systems at the circuit level. Specifically to gain experience with the use of the *switch* (SW), *pulse width modulator* (MOD) and *clock* (CLK) elements used in PECS and which appear in later labs.

Before using the PECS simulator, read the documentation which provides an overview of PECS sufficient for this and subsequent labs.

2.2 Circuit #1

The circuit in Figure 2.1 switches a voltage source (V1) ON and OFF to produce a rectangular wave (appearing across voltage port VP1) which is then filtered through an LC network which produces a lower dc voltage across a load (R1). This voltage, monitored by port VP2, will be examined.

This example circuit also illustrates how *clock* elements can be used to control switches.

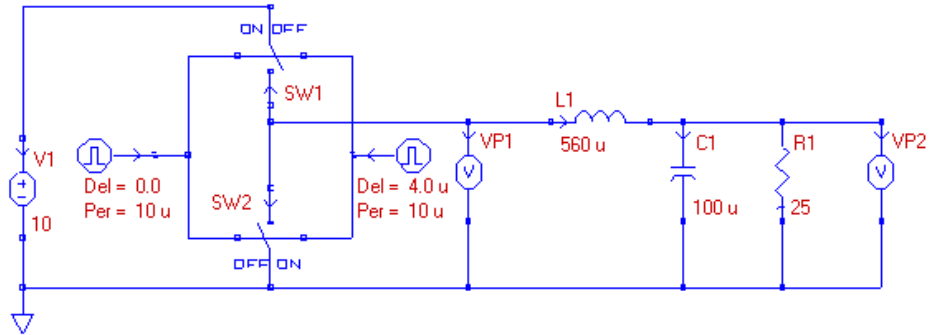


Figure 2.1: A pulse generator followed by a second order (LC) filter, ($L = 560 \mu H$, $C = 100 \mu F$ and $R = 25 \Omega$) constitutes a dc-to-dc voltage converter. (buck_clocks.ckt)

2.2.1 Tasks

1. Construct the circuit in Figure 2.1 in PECS. Leave the initial conditions for L and C at zero and set the initial state of the switches to ON and OFF for the top and bottom switches, respectively. (However, the initial switch state is not important here but must be set to some state prior to trying to run a simulation). Also, be aware when building the circuit that connections to components are only made at the nodes of the components
2. Use the following simulation parameters: *Final Time* = 6e-2, *Step Size* = 2e-7, *Start time* = 0, *End time* = 6e-2. These simulation parameters will provide a simulation of the circuit for 60 ms (as specified by *Final Time*) where recording of the waveform points occur every 200 ns (as specified by *Step Size*) as well as points occurring at switching discontinuities. At the end of the simulation time points occurring between the *Start Time* and *End Time* will be saved to the hard drive, which will subsequently be read by the plotting program (PECS PLOT). For the times chosen here, points from the whole simulation run will be saved. The relatively short *Step Size* value chosen results in many points being saved which will produce smooth output plots.
3. Run the simulation. This can be initiated from the menu items by selecting *Simulation* → *Run*, or more conveniently from the top icon bar by clicking the (left) script R (which appears in red).
4. Obtain a plot of the waveform across VP2, the output voltage. Looking at your plot determine the steady state value, that is, the value of the output voltage at the end of the simulation.
5. We would like to now look more closely at 4 or 5 cycles at the end of the simulation. Use the *Zoom* feature in PECS PLOT to isolate these. Access

this through menu items *View* → *Zoom* and then use the mouse to isolate the time period of interest. Left click the mouse anywhere in the plotting area at the desired starting time and drag to the desired ending time. This can be repeated any number of times to hone into your desired time interval.

6. Under the plot obtained in the previous task add the VP1 waveform. Do this by accessing from the menu items: *Plots* → *Add Plot* then select the desired out from the list of outputs shown.
7. Use the measuring feature (accessed through menu item *Plots* → *Measure*) to find the peak-to-peak voltage ripple of the output voltage (VP2). Use the right (→) and left (←) arrows on the keyboard to precisely pinpoint the maximum and minimum values of the waveform after initially placing the measurement markers by clicking the left and right mouse buttons.
8. For the waveform of VP1, note the peak amplitude, period and pulse width of this waveform. Determine the *duty ratio* of the pulse train. The duty ratio is defined as the length of the high portion of the pulse, i.e. the pulse width, divided by the period.
9. Taking the above plots into consideration, explain why you would expect to get the steady state value you found in task (4).

2.3 Circuit #2

The circuit in Figure 2.2 produces a sawtooth voltage waveform which appears across the capacitor (C1). This is achieved by using a constant current source (I1) to charge the capacitor and having the capacitor rapidly discharged through the resistor (R1) when the voltage has reached a preset level. The capacitor voltage and current are monitored by VP1 and IP1, respectively.

This circuit illustrates the use of the pulse width modulator element. However, in this circuit not all features of the modulator are used. The modulator is basically a comparator with four inputs, two of which connect externally, one connects to a user set constant value and the last connects to an internal sawtooth signal. We will just use one of the external connections and the internal constant input. (The internal sawtooth signal is not used).

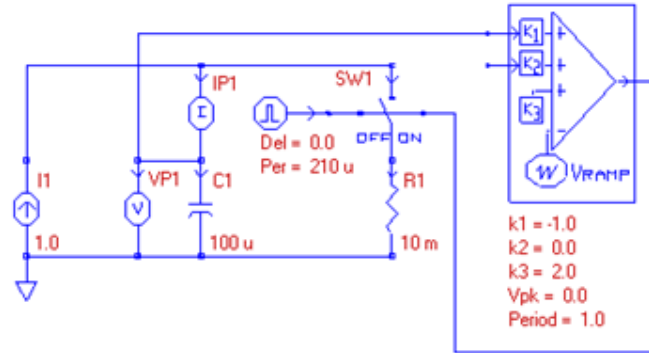


Figure 2.2: Sawtooth generator. A 1 A current source charges a $100 \mu\text{F}$ capacitor which is periodically quickly discharged through a $10 \text{ m}\Omega$ resistor. (sawtooth_3.ckt)

2.3.1 Tasks

1. Construct the circuit in Figure 2.2 in PECS. Note that for the modulator we are setting $K_1 = -1$ and $K_3 = 2$, with the other modulator parameters left untouched. The clock element has a *Delay* = 0 and *Period* = $210\text{e-}6$.
2. Use the following simulation parameters: *Final Time* = $1\text{e-}3$, *Step Size* = $1\text{e-}6$, *Start time* = 0, *End time* = $1\text{e-}3$. Run the simulation.
3. Obtain a plot of the voltage waveform across the capacitor (VP1) and under this have a plot of the capacitor current (IP1).
4. For the waveform of VP1, note the peak amplitude and period. Given that $K_1 = -1$, what other factors in the circuit determines the peak amplitude and why?

2.4 Circuit #3

We will modify the circuit in Figure 2.1 by replacing one of the clocks with a modulator and a DC voltage source. The modulator output is connected to the switch control terminal where the deleted clock had been. The added voltage source's positive terminal is connected to the top input of the modulator. The negative terminal is connected to ground. The configuration we are seeking is for the modulator to turn OFF the upper switch (and turn ON the lower switch).

The modulator parameter K_1 associated with the top modulator input is set to $K_1 = 1$. (Also be sure to have $K_2 = 0$ and $K_3 = 0$). We will also specify the modulator internal sawtooth to have a peak voltage of 5 V, i.e. $V_{pk} = 5$.

Furthermore, we will change the frequency of operation of this circuit to 40 kHz . So the period parameters of the modulator and the remaining clock should be changed to $25\ \mu\text{s}$, i.e. set $Period = 25\ \mu\text{s}$, for both these components.

Also, we will operate the circuit at 50% duty ratio. Determine the value of the voltage source needed to achieve this, given the parameters stated above.

2.4.1 Tasks

1. Obtain a copy of your schematic. (buck_mod.ckt)
2. Repeat the tasks of Circuit #1.
3. Complete the following table.

	Switching Frequency	Duty Ratio	Peak-to-peak Input Voltage to Filter	Steady State Average Output Voltage	Peak-to-peak Output Voltage Ripple
Circuit #1					
Circuit #3					

4. Explain the differences seen in the peak-to-peak ripple voltage values between Circuit #3 and Circuit #1. Are they in line with your expectations? Why?

Chapter 3

Lab 2

System Identification of a 2nd Order System through Step Response

3.1 Objectives

To observe the step response of a second order system exemplified by an *LCR* circuit. By taking note of certain features of the response the parameters of its transfer function will be deduced thus performing a *system identification*.

3.2 Background

In classical control a transfer function is used to mathematically describe the system to be controlled. The *LCR* network examined in this lab is an important component of the system we will be using in subsequent labs for which we will be designing controllers.

System identification is the term used to describe the process by which a system's transfer function is determined by suitable probing of inputs and examining the resulting output response. In this lab we will examine the step response of a second order system to determine the parameters of its transfer function. We will use a *small-signal* approach. Small signal refers to the fact that the response will be examined for a small variation around the DC operating point which is not necessarily zero. Since the *LCR* circuit is linear, all small signal models will be identical. That is, a change of operating point does not alter the transfer function in this case. Nevertheless we will demonstrate the small signal approach here.

Note that for an *LCR* network the transfer function can be easily determined through fundamental circuit analysis, so the student will use the results obtained through this procedure to confirm their identified model.

The *LCR* network examined in this lab is shown in Figure 3.1.

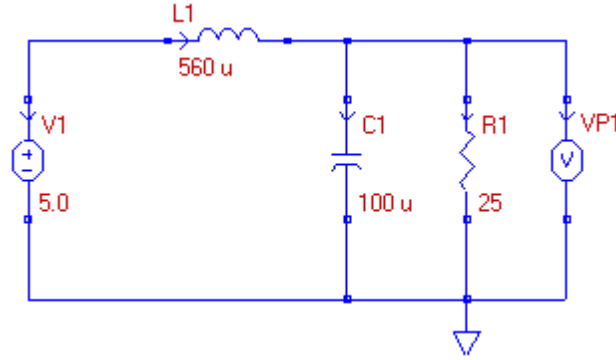


Figure 3.1: PECS schematic of the LCR network used in this lab. The values of the components are $L = 560 \mu H$, $C = 100 \mu F$ and $R = 25 \Omega$. (LCR_step_0.ckt)

The input is V1, the 5 V voltage source, and the output is the voltage which appears across R1 which in PECS is monitored by the voltage port VP1. (In PECS the presence of a voltage port dictates to the simulator the requirement to store the voltage appearing across this port in the output variables to be subsequently plotted).

To obtain the small-signal response, a 10% step input will be applied (V1 changes from 5V to 5.5V) and the consequential voltage changes around the steady state value of the output voltage will be examined.

We will consider a second order transfer function, $G(s)$, expressed as follows:

$$G(s) = K \frac{1}{a_2 s^2 + a_1 s + 1}$$

We see that there are three parameters a_1 , a_2 and K that need to be determined by the identification procedure to fully characterize this system. In the case of a second order underdamped system, that is, a system that features complex poles. The transfer function can also be written as

$$G(s) = K \frac{1}{\left(\frac{s}{\omega_n}\right)^2 + 2\xi \frac{s}{\omega_n} + 1}$$

where ω_n is the undamped natural frequency and ξ is the damping factor where $\xi < 1$.

Consequently the output step response, $c(t)$, can be found to be [1]

$$c(t) = 1 - \frac{1}{\sqrt{1-\xi^2}} e^{-\xi\omega_n t} \cos\left(\omega_n \sqrt{1-\xi^2} t - \phi\right)$$

$$\text{where } \phi = \tan^{-1}\left(\frac{\xi}{\sqrt{1-\xi^2}}\right).$$

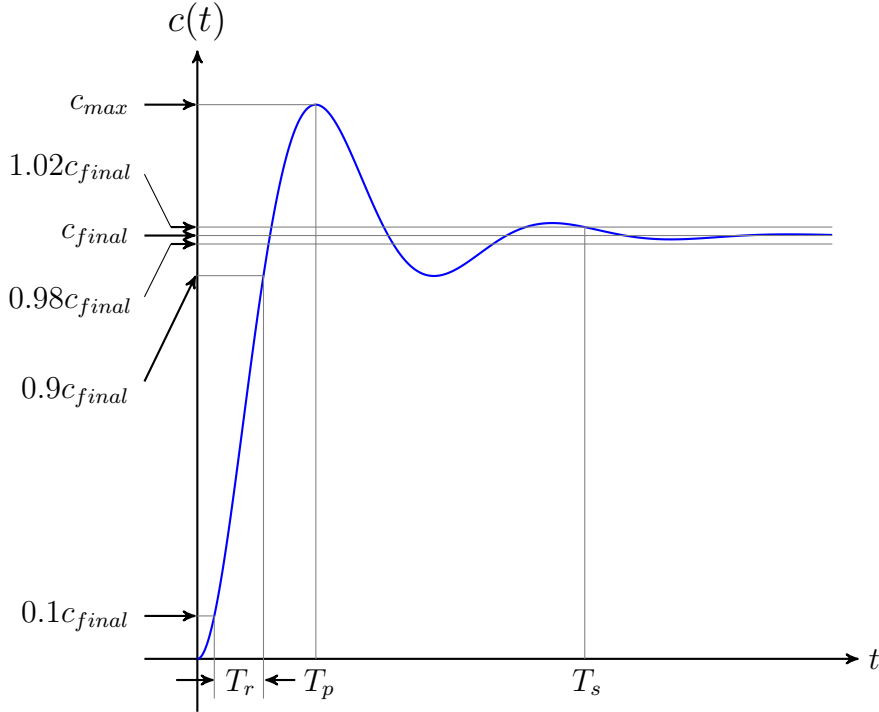


Figure 3.2: Second-order underdamped step response

A typical step response for an underdamped second order system is shown in Figure 3.2.

Of particular note in this response is the *percentage overshoot*, $\%OS$, and *settling time*, T_s .

The *percentage overshoot* is the maximum amount the output waveform overshoots the steady state, or final value, expressed as a percentage of the final value. Therefore, it is given by:

$$\%OS = \frac{c_{max} - c_{final}}{c_{final}} \times 100$$

The variables c_{max} and c_{final} are shown in Figure 3.2.

On the other hand, the *settling time*, T_s , is the time taken from the initiation of the step for the output waveform oscillations to remain within a band of $\pm 2\%$ of the final value.

We will determine these two measures from a simulation and use them to subsequently determine ω_n and ξ as seen next.

Using the expression for the output step response, the percentage overshoot can be found to be given by

$$\%OS = e^{-\left(\frac{\xi\pi}{\sqrt{1-\xi^2}}\right)} \times 100$$

Making ξ the subject of the equation leads to:

$$\xi = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}}$$

To determine the settling time, T_s , we again use the expression for the output step response $c(t)$ given above to find [1]

$$T_s = \frac{4}{\xi\omega_n}$$

or

$$\omega_n = \frac{4}{\xi T_s}$$

Thus from the simulation plot one can determine the variables c_{max} , c_{final} and T_s and then utilize the above formulas to find ω_n and ξ .

The final parameter to be determined is K , which represents the DC (i.e. zero frequency) gain of the transfer function and is given by

$$K = \frac{\Delta c}{\Delta v} = \frac{c_{final} - c_0}{v_{final} - v_0}$$

where c_{final} and c_0 represent the final (steady state) output level after the step has been applied and the (steady state) value of the output before the application of the step input, respectively. Similarly for v_{final} and v_0 which now refers to the input. We will apply a 10% step input $\Delta v = 0.5$ with $v_{final} = 5.5$ and $v_0 = 5$.

3.3 Tasks

3.3.1 PECS simulation

1. Enter the *LCR* schematic shown in Fig. 1 into PECS. Note the values of the components: $L = 560 \mu H$, $C = 100 \mu F$ and $R = 25 \Omega$. Leave the initial conditions for the inductor and capacitor at zero.
2. Setup the input source to achieve a 10% step of the input voltage. As mentioned above we will change the input voltage from $V1 = 5 V$ to $V1 = 5.5 V$. To set this up in PECS we need to bring up the V1 element dialog window by double clicking on the V1 symbol. Enter the initial value of $V1 = 5 V$ in the *Value* parameter space. Next click on the *Steps* button and then enter $t1 = 0.11$ and $v1 = 5.5$. This is the time at which and value to which the input voltage V1 will change.

3. To set up the simulation parameters for the simulation click on the menu item *Simulation* → *Parameters* ... This will bring up a dialog window. Enter the following parameters: *Final time = 0.2*, *Step Size = 1e-6*, *Start Time = 0.101* and *End time = 0.2*.
4. Run the simulation. From the resulting plot obtained from the simulation determine the peak overshoot value and the final steady state value. We will refer to these values as c'_{max} and c'_{final} .
5. To determine the small signal model we will subtract the initial output voltage level c'_0 from the two voltage levels found in Task (4) and also offset the step time to zero. So that in terms of the small signal model we have:

$$\begin{aligned} c_{max} &= c'_{max} - c'_0 \\ c_{final} &= c'_{final} - c'_0 \end{aligned}$$

6. Using the values from Task (5) determine the percentage overshoot, %OS, and subsequently determine the damping factor, ξ , using the formulas given above.
7. We will now determine the settling time. From your step response plot determine the time value T'_s at which further oscillations remain with $\pm 2\%$ of the small signal final value. The small signal settling time T_s is obtained by offsetting the time of the step:

$$T_s = T'_s - \text{step time}$$

8. The final parameter of our model K can next be determined using the appropriate formula from above. The model is now fully specified.

3.3.2 Matlab simulation – results check

We will next check the results obtained above.

9. Derive the transfer function of the *LCR* circuit. This is perhaps most easily achieved using the impedance divider rule. Obtain your transfer function in symbolic form. Identify the transfer function coefficients K , $a1$ and $a2$ which are functions of the component values L , C and R . Complete the following table and include it in your report:

Transfer Function Parameters	PECS simulation derived values (from Tasks 6,7&8)	From transfer function derivation: symbolic form	From transfer function derivation: evaluated
ξ			
ω_n			
K			

10. We will now use a simulation function available in Matlab (*lsim*) to obtain the step response in an analogous fashion to that obtained using PECS. Use the following Matlab code to do this. Be sure to understand each line of code.

Before you can run this code, enter K , a_1 and a_2 as functions of L , C and R (found in the previous task) on the appropriate lines.

```

L = 560e-6;
C = 100e-6;
R = 25;

% enter transfer function parameters K, a1 & a2 as functions of L, C and R
K = ?; % transfer function DC gain
a1 = ?; % coefficient of s^1 in denominator polynomial
a2 = ?; % coefficient of s^2 in denominator polynomial

tf_LCR = tf(K, [a2, a1, 1]); % transfer function of the RLC network

% t is a vector of 1000 time values linearly spaced between 0 and 0.2
t = linspace(0, 0.2, 1000);
u = 5*ones(length(t), 1);
step_time = 0.11; % step the input at step_time
n = find(t >= step_time);
u(n) = u(n) + 0.5; % input containing the 10% step at step_time

y = lsim(tf_LCR,u,t); % simulate the LCR network with the desired input
figure(1)
plot(t,y)
title('Output response including the large-signal start-up transient')

% isolate the small-signal step response
c_prime_0 = y(n(1)-1); % initial output before the step
ys = y(n) - c_prime_0; % small signal output response
ts = t(n) - step_time; % small signal response times

figure(2)
plot(ts, ys)

```

```
title('Step response')

stepinfo(ys,ts) % obtain the step response metrics
```

11. Compare the plots obtained from the Matlab code to that obtained using PECS.
12. Compare the values of percentage overshoot and settling time determined by the Matlab function *stepinfo* in the above Matlab code to the values you determined from the PECS simulation. Complete the following table:

Response Feature	From PECS Simulation	From Matlab function <i>stepinfo</i>
% overshoot %OS		
Settling Time, T_s		

13. In the subsequent labs we will use the *LCR* network as a low pass filter but we will include the losses of the inductor which are modelled as an equivalent series resistance (*ESR*). In this task we'll qualitatively assess the effect of this *ESR* on the transfer function.

Rederive the transfer function of the network where the inductor L is now replaced with a series combination of the inductor L and a resistor r_L . Note: in deriving the new transfer function a short cut can be taken by using the previously obtained transfer function and replacing any term of sL with $sL + r_L$.

To more easily determine the qualitative effect of loss inclusion in the transfer function, the newly obtained transfer function is simplified with the approximation:

$$R + r_L \approx R$$

which is true when $r_L \ll R$.

With the above approximation the qualitative change in the response can be determined by inspection. State your observation on this matter. Hint: Does it (substantially) change any of the following of the transfer function:

- (a) DC gain?
- (b) Undamped natural frequency?
- (c) Damping factor?

Reference:

- [1] Norman S. Nise, "Control Systems Engineering", 7th edition. Pages 173 – 176.

Chapter 4

Lab 3

System Identification of the dc-to-dc Buck converter

4.1 Objectives

The circuit simulator PECS will be used to examine and characterize the steady state operating conditions of the dc-to-dc buck converter and will aid in the confirmation of the system transfer functions. The open loop response to step changes in the input voltage as well as step load changes are also examined using both PECS and Matlab. Matlab will also be used in examining the system transfer functions. Through these simulations the student will gain a better understanding of the operation of the buck converter circuit. The transfer functions verified here will be used later to design effective closed loop feedback control.

4.2 Background

The LCR network examined in the previous lab is used together with a single pole, double throw switch to derive a power processing circuit known as the dc-to-dc Buck converter. This circuit takes a dc voltage source at the input and transforms it to a lower value dc level at the output, whilst achieving a high power efficiency, typically in the 90% range. This is possible through the use of a high frequency switch.

The schematic of the buck converter is shown below in Figure 4.1. The converter schematic has been divided into 3 sections: 1) the input voltage source, V_g , 2) the single-pole double-throw switch, and 3) the LCR network which com-

prises the low-pass filter. To more accurately model the losses in the inductor a series resistor, r_L , is included. The output load to which the power is delivered is represented by resistance R .

The switch is operated in cyclical manner with period, T_s . At the start of each period the switch is in its upper position which connects the input source to the output filter. The switch stays in this position for length of time, DT_s , where D is known as the *duty ratio* which takes a value $0 < D < 1$. At the end of the DT_s subinterval the switch changes position to disconnect the input source.

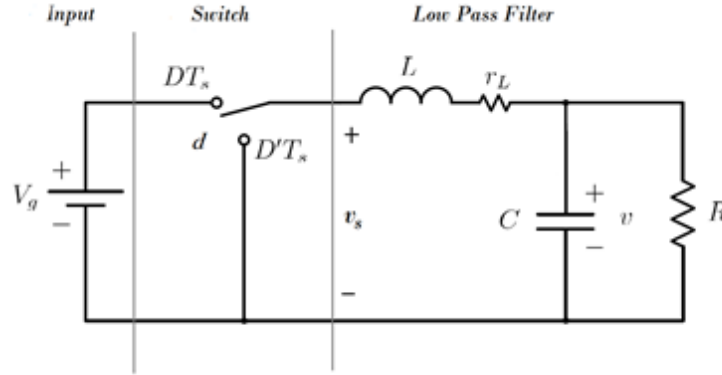
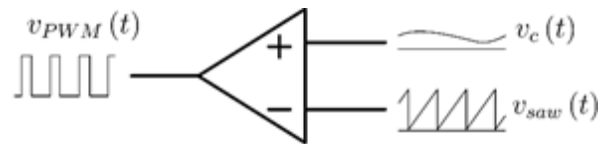


Figure 4.1: Dc-to-dc Buck converter.

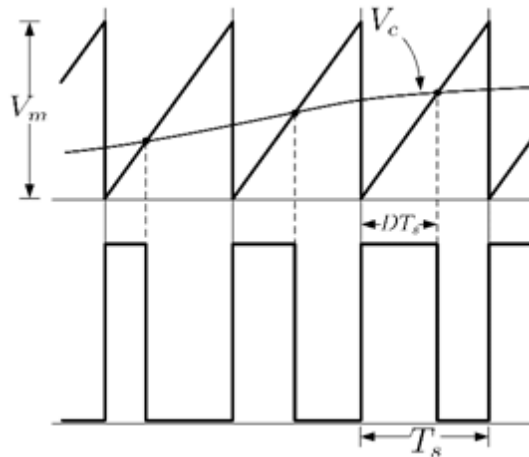
The length of time remaining until the end of cycle is $D'T_s$ where $D' \equiv 1 - D$. So we see that through the switching action the voltage v_s appearing at the input of the low pass filter is a rectangular wave. The output of the filter is predominately a constant dc level, which corresponds to the average of the input waveform, together with a small ripple which represents the unfiltered residuals of the input waveform. Neglecting losses (which generally will be very small by design) the dc output voltage is given by $V = DV_g$. Given the range of D we see that the output can be adjusted from 0 to V_g . Control of the output voltage is achieved by variation of the duty ratio. To denote a time varying quantity a lower case symbol will be used and thus a varying duty ratio is denoted by d . Also, to further highlight a signal that represents a deviation around a steady state average, we will use a caret '^', so that \hat{d} is the small signal deviation around the average duty ratio D . Therefore, the time varying duty ratio $d(t)$ is comprised of an average (DC) value together with a deviation \hat{d} around this average so that $d = D + \hat{d}$ or alternatively, $\hat{d} = d - D$.

To achieve a varying duty ratio given a control voltage, v_c , a pulse width modulator (PWM) circuit is used, see Figure 4.2. This is comprised of an op-

amp comparator to which one of the inputs is connected to a sawtooth waveform, v_{saw} , which has a peak-to-peak amplitude of V_M and period T_s . As shown in Figure (2b) a comparison of the control voltage, v_c , results in a rectangular output waveform of period, T_s , and duty ratio, d . Combining the PWM with the buck converter results in an open loop controlled buck converter shown in Figure 4.3. Through the use of the PECS simulator we will examine the steady state operation of this system.



(a)



(b)

Figure 4.2: (a) PWM comparator, (b) comparison of the control signal, v_c , with the sawtooth waveform v_{saw} , results in a variable pulse width rectangular waveform used to drive the switch in the buck converter.

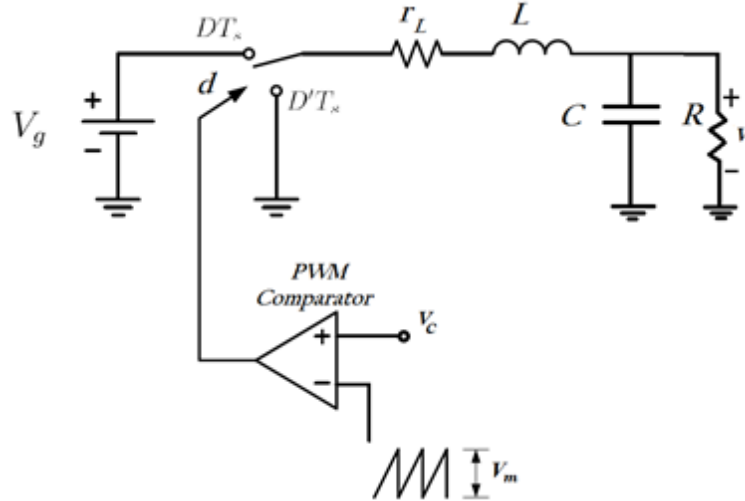


Figure 4.3: Open loop controlled buck converter. The duty ratio d of the converter is set by the control voltage, v_c , via the PWM comparator.

4.3 System Transfer Functions

In a later lab we will design a closed loop controller for this system. To be able to do this using a classical control design methodology we will first need to determine important transfer functions of the system. We will consider three transfer functions. The first is the *control-to-output voltage* transfer function. This transfer function has two components: the *duty ratio to output voltage* transfer function of the buck converter power stage, G_{vd} , and the pulse width modulator transfer function, G_{PWM} .

Control-to-output transfer function:

$$\frac{\hat{v}}{\hat{v}_c} = G_{vd} \cdot G_{PWM} = \frac{\hat{v}}{\hat{d}} \cdot \frac{\hat{d}}{\hat{v}_c}$$

where the caret ($\hat{\cdot}$) has been used to denote a small signal signal quantity. Note that since the modulator is nonlinear, a so-called *describing function* analysis method is used to determine the transfer function which ends up being a (frequency independent) constant gain given by:

$$G_{PWM} = \frac{\hat{d}}{\hat{v}_c} = \frac{1}{V_M}$$

The control-to-output transfer function plays a very important role in control design as it forms part of the loop gain which is important to stability and achieving good stability margins (both phase and gain).

The other two transfer functions of the buck converter we will consider are 1) the *input voltage to output voltage* transfer function and, 2) *output load current to output voltage* transfer function:

1. input voltage to output voltage transfer function: $G_{vg} = \frac{\hat{v}}{\hat{v}_g}$
2. output load current to output voltage transfer function: $\frac{\hat{v}}{\hat{i}_o} = -Z_{out}$

These transfer functions quantify how variations in the input quantity at various frequencies propagate to the output. That is, how much of an affect does input voltage variations or load current variations have on the output voltage. Ideally, in a voltage regulator system as we are considering here, we would like this to be zero. In this lab we will examine the Bode magnitude response to see what level of transmission is achieved in open loop operation. With a properly designed control system incorporating feedback, these input disturbance propagations through the system will be greatly diminished.

A block diagram model of the buck converter transfer functions which will be used in a later lab for controller design is shown in Figure 4.4. In this lab we'll do a partial verification of these transfer functions comparing them with previously derived results.

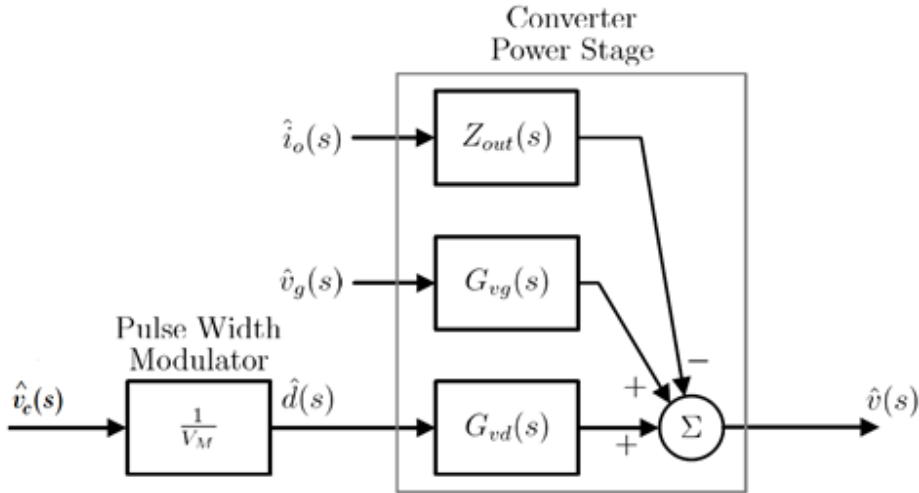


Figure 4.4: Block diagram model of the Buck converter along with the pulse width modulator.

For transfer functions G_{vg} and G_{vd} considering the signal flow from the input through to the output of the converter, one can readily assume a form as follows

$$G_{vg}(s) = \frac{\hat{v}_s}{\hat{v}_g} \cdot \frac{\hat{v}}{\hat{v}_s} = K_{vg} \cdot G_{LPF}(s)$$

and

$$G_{vd}(s) = \frac{\hat{v}_s}{\hat{d}} \cdot \frac{\hat{v}}{\hat{v}_s} = K_{vd} \cdot G_{LPF}(s)$$

where \hat{v}_s represents the small signal voltage variations at the input of the output filter and where K_{vg} and K_{vd} are constant gains and $G_{LPF}(s)$ represents the transfer function of the second order low pass *LCR* filter which, as seen in the previous lab, is given by

$$G_{LPF}(s) = \frac{1}{\left(\frac{s}{\omega_n}\right)^2 + 2\xi\frac{s}{\omega_n} + 1}$$

Note that the damping ratio, ξ , now includes the effect of inductor losses, as considered at the end of Lab. 2, by including r_L , the inductor ESR (equivalent series resistance). Constants K_{vg} and K_{vd} represent the effect of the switching elements in propagating variations of the input voltage level (for G_{vg}) or variations of the duty ratio (for G_{vd}) to the input of the low pass filter. K_{vg} and K_{vd} represent the DC gain of the relevant transfer functions and will be found through simulation below.

Output load current to output voltage transfer function: $\frac{\hat{v}}{\hat{i}_o} = -Z_{out}$:

The final transfer function that we'll consider is related to the output impedance of the buck converter. Despite the presence of switching which affected the two other transfer functions considered (i.e. G_{vd} and G_{vg}), the output impedance is more straightforwardly determined as switching has no effect. The circuit configuration of the buck converter during the first (DT_s) subinterval is seen in Figure 4.5a, and during the remainder of the period (the $D'T_s$ interval) is seen in the Figure 4.5b. As independent sources are nulled for determination of impedances we can see that the output impedance, Z_{out} , are the same during the two subintervals and can be simply seen as a parallel connection of three impedances such that:

$$Z_{out} = (sL + r_L) \parallel \frac{1}{sC} \parallel R$$

You will be asked to evaluate this in a latter task.

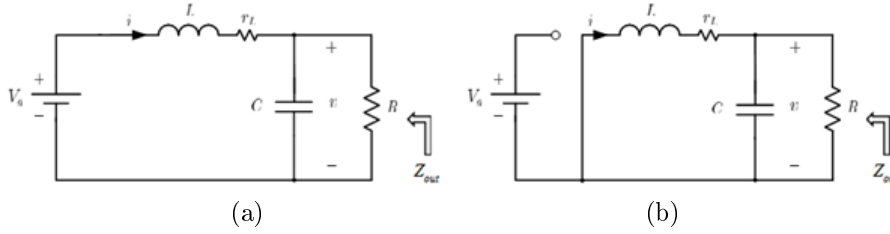


Figure 4.5: Buck converter configuration during (a) the first subinterval, DT_s , and (b) the second subinterval, $D'T_s$.

The actual transfer function of interest for us is that related to how output current variations, \hat{i}_o , lead to output voltage variations, \hat{v} . With reference to Figure 4.6 we see that the output current i_o is given by

$$i_o = I_o + \hat{i}_o$$

where the capitalized symbol refers to the DC steady state value and the term with a caret ($\hat{}$) indicates a small-signal variation. A similar expression can be written for the output voltage: $v = V + \hat{v}$

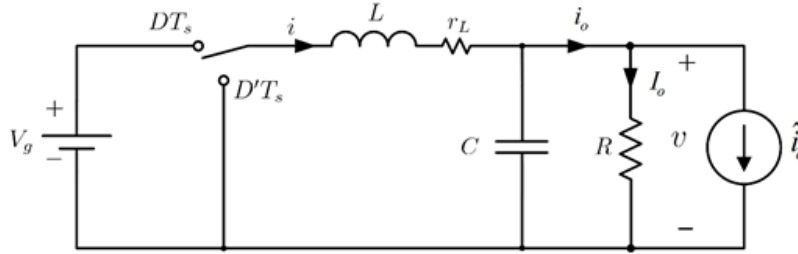


Figure 4.6: The effect of output current variations, \hat{i}_o , causing output voltage variations, \hat{v} , is quantified by transfer function $-Z_{out}$.

Consequently the small signal transfer function for the output impedance Z_{out} is given by

$$Z_{out} = \frac{\hat{v}}{-\hat{i}_o}$$

and consequently the transfer function of interest $\frac{\hat{v}}{\hat{i}_o}$ which we would like to determine is given by

$$\frac{\hat{v}}{\hat{i}_o}(s) = -Z_{out}$$

Note however that in the simulation (and later in the lab with the hardware implementation), in order to achieve output current variations we will be stepping the load, that is, changing the load resistance between two different values.

This however causes the system transfer functions to be modified somewhat. However, as seen in Lab. 2 the major change that occurs is that of varying the damping factor of the circuit. Nevertheless considering the convenience of performing a step load change in the lab, the approximation considered here will be accepted.

4.4 Tasks

1. **Start-up transient:** In this first task we will examine the start-up transient and steady state operation of the buck converter. Build in PECS the buck converter circuit shown in Figure 4.7 below. Use simulation parameters: *final time = 0.06*, *step size = 1e-6*, *start time = 0* and *end time = 0.06*. Run the simulation and obtain the output voltage plot. Note that the large start-up transient is not of much interest to us here as it is a large signal phenomenon and may be easily avoided with a slight redesign. However, note the steady state value of the output voltage.

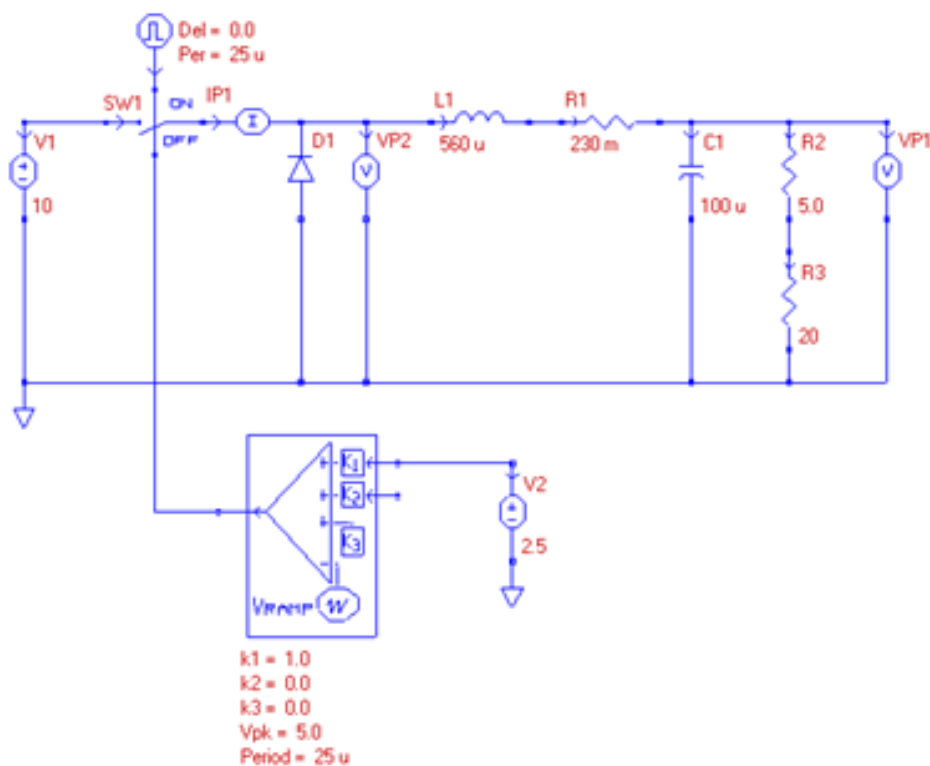


Figure 4.7: PECS schematic of the Buck converter with pulse width modulator. (Lab3_1.ckt)

2. **Steady-state characteristics:** We will now rerun the simulation except we will only display the last few cycles of the response. This can be achieved by setting the *start time* = $5.9881e-2$ in the simulation parameters dialog window and rerunning the simulation. Obtain a plot of the output voltage, the diode voltage and switch current, in three separate plots one above the other so that relative timing relationships can be observed. You can do this by first producing a plot of the output voltage (VP1), then using PECS PLOT to add a plot containing the diode voltage (VP2), this will be added under the current plot, followed subsequently by adding the switch current (IP1) plot. (Lab3_1a.ckt)

Determine the following from the plots:

- (a) The average output voltage (a visual estimate is sufficient)
- (b) The peak-to-peak output voltage (this is the output voltage ‘ripple’)
- (c) The peak-to-peak diode voltage
- (d) The peak inductor current,
- (e) The frequency of the waveforms
- (f) The duty ratio

Explain how the duty ratio is set in the circuit, given the current parameters of the sawtooth waveform.

3. **K_{vg} and Step input voltage change:** We will now determine constant K_{vg} using a step change of the input voltage and monitoring the resulting output voltage change, as was previously done in Lab 2. Recall from Lab 2 that

$$K_{vg} = \frac{\Delta c}{\Delta v} = \frac{c_{final} - c_0}{v_{final} - v_0}$$

where c_{final} and c_0 represent the final (steady state) output level after the step has been applied and the value of the output before the application of the step input, respectively. Similarly for v_{final} and v_0 which now refers to the input. We will apply a unit step input $\Delta v = 1$ with $v_{final} = 11$ and $v_0 = 10$.

To set this up in PECS we will introduce two step changes in the input source. Bring up the input source (V1) parameter dialog window click on the *Steps . . .* button and set $t1 = 0.025$ with $v1 = 11$ and $t2 = 0.04$ with $v2 = 10$. We’ll also change the simulation parameter *Start Time* = $2.001e-002$. Obtain the simulation plot and subsequently determine K_{vg} using the above equation. Also determine the maximum peak-to-peak output

voltage deviation Δv and steady state error, SSE . The SSE is simply the absolute difference between the steady state values before and after the input change, so this is represented by $|c_{final} - c_0|$. For an example of how these are determined see the note below. The SSE is somewhat misnamed in the open loop context we have here but this metric will be used later in comparison with closed loop contexts where the steady state output voltage is desired to remain unchanged in the presence of a disturbance input. (Lab3_1b.ckt)

4. K_{vd} : To determine constant K_{vd} we'll use a different input to that of task (3). A sinusoidal input at a low frequency, much lower than the low pass filter corner frequency, will drive the input of the modulator and the resulting sinusoidal output voltage of the converter will be monitored. To be clear about this, determine the filter corner frequency expressed in Hz. The sinusoidal source is attached as shown in Figure 4.8. We will use sinusoidal source parameters: *Peak Amplitude* = 0.2, *Frequency* = 100 and *Phase (degs)* = 0. Use the same simulation parameters as in the previous task. However, be sure to void the step changes that were introduced in the previous task.

Run the simulation and obtain a plot of the driving sinusoid (VP3). Confirm that the amplitude and frequency are as desired. Now use the add plot feature in PECS PLOT to add the plot of the output voltage (VP1) below the current plot. Obtain peak-to-peak measurements of the two waveforms and determine the voltage gain. Take note of the phase relationship between the two waveforms. Are they in-phase or out-of-phase?

Considering the influence of the PWM gain, determine K_{vd} . Hint: From previous discussions (and also from Figure 4.4) we see that the gain found is given by

$$\frac{\hat{v}}{\hat{v}_c}(s) = G_{PWM} \cdot G_{vd} = G_{PWM} \cdot K_{vd} G_{LPF}(s)$$

With $G_{PWM} = \frac{1}{V_M}$ (with V_M known) and for the frequency used at well below the low pass filter corner frequency such that $G_{LPF} \approx 1$, the parameter K_{vd} may be easily determined.

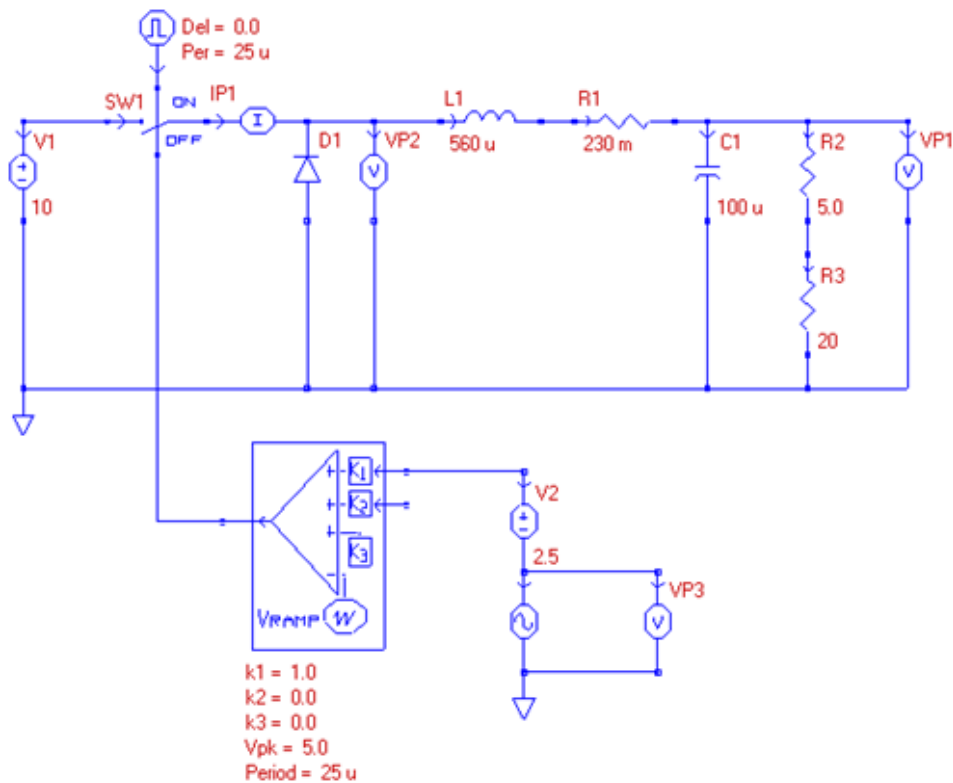


Figure 4.8: Buck converter with pulse width modulator driven by a sinusoidal voltage source. The added sinusoidal source has parameters: *Peak Amplitude* = 0.2 volts, *Frequency* = 100 Hz. and *Phase (degs)* = 0. (Lab3_2.ckt)

5. **Step load change:** We next examine the output voltage change to a step in load. This test can be easily performed for an actual circuit and so will be undertaken subsequently on the hardware in the lab and will form the basis for examining how well the application of feedback improves on the open loop performance.

Alter the buck converter circuit by adding a switch which is controlled by two clock elements as shown in Figure 4.9. Be sure to set the initial switch state of the added switch to be in the OFF state, otherwise the simulation will be incorrect. Also make sure no other step changes are initiated in your circuit. Run the simulation using the simulation parameters used before and obtain a plot of the output voltage (VP1). From this plot determine the maximum peak-to-peak output voltage deviation, Δv , and steady state error, *SSE*.

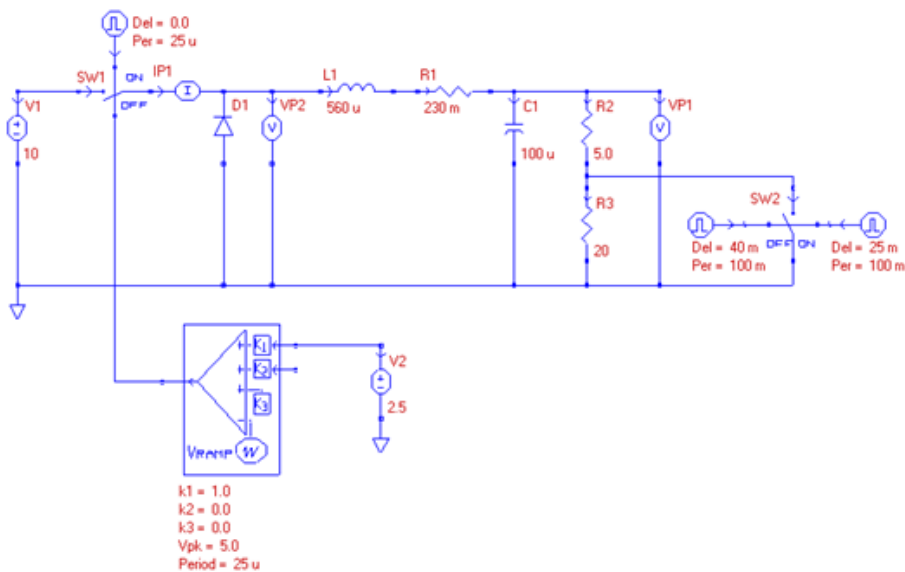


Figure 4.9: Buck converter with step load changes (Lab3_3.ckt)

6. Matlab

In the following tasks we will analyze the open-loop system using Matlab. We'll examine both disturbance input transfer functions and also use Matlab to perform time-domain simulations based on these transfer functions. In the model for the PWM modulator use a peak-to-peak ramp amplitude value of $V_M = 5$.

Transfer functions

- (a) **Loop gain:** Under the condition that there is no compensator, i.e. $G_c = 1$ and the desired output voltage and reference voltages are, $V = 5$ and $V_{ref} = 2.5$, respectively so that $H(s) = \frac{V_{ref}}{V} = \frac{2.5}{5} = 0.5$, determine the loop gain transfer function and use the Matlab *margin* command to obtain the Bode plot of this (uncompensated) loop gain. Have the plot display frequency in Hz. The command will also obtain the unity gain and -180° phase crossover frequencies and the phase and gain margins of the system. Make note of these in your report. Matlab code to do this:

```
s = tf('s');
G_loop = ?; % input your loop gain expression as a function
of s
figure(1)
margin(G_loop)
h = gcr;
```

```

h.AxesGrid.Xunits = 'Hz'; % display the frequency in Hz
h.AxesGrid.TitleStyle.FontSize=12; % increase font size
h.AxesGrid.XLabelStyle.FontSize=12; % for readability
h.AxesGrid.YLabelStyle.FontSize=12;

```

- (b) **Input voltage to output voltage:** Using the model for the (open-loop) input source voltage to output voltage transfer function, G_{vg} , use the Matlab *bodemag* command to obtain the magnitude frequency response of this transfer function (with frequency in Hz).
- (c) **Output current to output voltage:** Using the model for the (open-loop) output current to output voltage transfer function, $-Z_{out}$, use the Matlab *bodemag* command to obtain the magnitude frequency response of this transfer function (with frequency in Hz).

Be sure to include your Matlab code for (b) and (c) in your report. These magnitude responses show the gain at various frequencies of input disturbance in propagating to the output. With the subsequent incorporation of feedback these responses will be greatly reduced. Ideally we would like the response to be zero across all frequencies. Needless to say that this cannot be achieved perfectly in practice.

Open-loop simulations:

Note that the following simulations obtained using Matlab are based on the small-signal model only. Thus DC conditions and large signal effects are not modelled and consequently do not show up in the simulations. In order to more easily compare the Matlab simulations with those obtained from PECS we will simply add the average converter output voltage to the response obtained from Matlab.

- (d) **Input voltage step response:** Use the transfer function obtained above for G_{vg} to obtain the response to 10% step input voltage change. Since the nominal input voltage level is 10 V, we therefore will use a unit step input. Use the Matlab *lsim* command to perform this simulation. Determine the maximum peak-to-peak output voltage deviation, Δv , and steady state error, *SSE*. Matlab code to do this:

```

Vg = 10;
D = 0.5;
V = D*Vg;
s = tf('s');
Gvg = ? ; % input your expression for Gvg as a function of s
t = linspace(0.02, 0.06, 1000);

```

```

u = zeros(size(t));
ind = find(t>=0.025 & t<=0.04);% step is between 0.025<t<0.04

Vg_diff = 1;
u(ind) = u(ind) + Vg_diff;% form input vector containing the step

figure(2)
y = lsim(Gvg, u, t); % simulate the step response
plot(t,y+V) % add steady state voltage to the output and plot it

del_v = max(y) - min(y) % peak-to-peak voltage deviation
SSE = y(ind(end)) % steady state error

```

- (e) **Output current step response:** Based on discussion in a prior section of this lab, determine the converter output impedance, Z_{out} . We will use the transfer function $-Z_{out}$ to obtain the response for a step load change. In order for this to mimic the practical circuit as closely as possible you will first determine the value of the current step involved. Taking note of the load switching circuit shown in Fig. 3, we see that the load switches between 25 ohm and 5 ohms. This results in the output current switching between $I_{o_1} = \frac{V}{25}$ and $I_{o_2} = \frac{V}{5}$, where we have assumed that the output voltage does not change appreciably. Thus the current step is $I_{o_diff} = I_{o_2} - I_{o_1}$. Use the Matlab *lsim* command to perform a step response simulation to this load step. Determine the maximum peak-to-peak output voltage deviation, Δv , and steady state error, *SSE*. Matlab code to do this:

```

s = tf('s');
Zout = ? ; % input your expression for Zout as a function of s
Vg = 10;
D = 0.5;
V = D*Vg;
Io_1 = V/25; % load current before step. (25 ohm load)
Io_2 = V/5; % load current after step. (5 ohm load)
Io_diff = Io_2 - Io_1; % current step
t = linspace(0.02, 0.06, 1000);
u = zeros(size(t));
ind = find(t>=0.025 & t<=0.04);% step is between 0.025<t<0.04

u(ind) = u(ind) + Io_diff;
figure(3)

```

```

y = lsim(-Zout, u, t);
plot(t,y+V)
del_v = max(y) - min(y)
SSE = y(ind(end))

```

4.5 Note

Computing maximum peak-to-peak output voltage deviation, Δv , and steady state error, SSE :

With reference to Figure 4.10 we see that the output voltage before the input step is at 5 V (see value at time = 0.2646). The input step occurs at time 0.30 causing the output to oscillate between max and min values of 5.996 and 3.199 before settling to new steady state output value of 4.816. Subsequently the input step reverts to its initial value at time 0.34 resulting in oscillations occurring between max and min values of 6.617 and 3.82. The quantity Δv is determined as the difference between the maximum and minimum deviations in the step response, so that $\Delta v = 6.617 - 3.199 = 3.418$ V. The SSE (steady state error) is determined as the difference in the two steady state values, so that $SSE = 5.0 - 4.816 = 0.1840$ V.

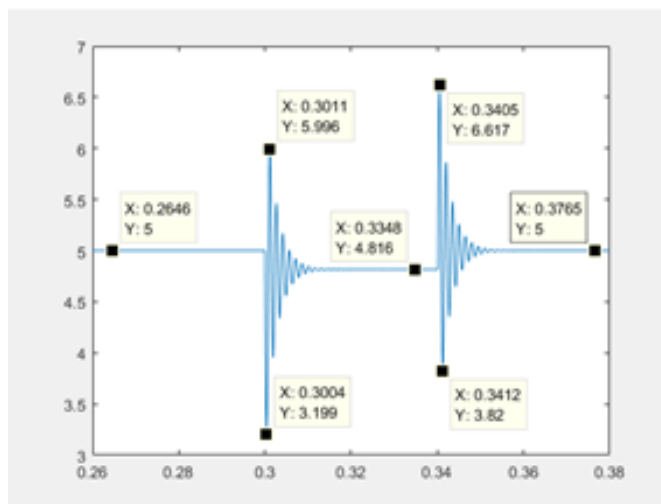


Figure 4.10: The plot shows the output voltage response to an input step change. This response is used to determine the maximum peak-to-peak output voltage deviation, Δv , and steady state error, SSE .

4.6 Results

Be sure to include a summary of your results by completing the following tables and including them in your report.

Transfer function DC gains:

	From PECS: Tasks (3) and (4)	From model: Symbolic Model	From model: Symbolic formula evaluated
K_{vg}			
K_{vd}			

Uncompensated stability margins and associated frequencies, from Task 6a:

Gain Margin	Phase Crossover Frequency	Phase Margin	Gain Crossover Frequency

Step response characteristics:

	PECS Tasks (3) and (5)	Matlab Tasks (6d) and (6e)
v_g step: Δv		
v_g step: SSE		
i_{out} step: Δv		
i_{out} step: SSE		

Provide answers to the following questions regarding the step response results:

1. Comparison of the results for Δv for i_{out} step changes obtained from PECS and Matlab differ more than one might expect. Provide a reason why. (Hint: look closely to how this is modelled).
2. From basic circuit considerations provide a simple symbolic expression for the SSE quantity found for a step in input voltage, v_g .

4.7 Postscript

In this lab we have developed a linear model for the buck converter system which resulted in deriving three transfer functions: 1) G_{vd} , 2) G_{vg} and 3) $-Z_{out}$. These transfer functions quantify how the duty ratio control input, \hat{d} , and other (disturbance) inputs, \hat{v}_g and \hat{i}_o , cause output voltage variations, \hat{v} . The pulse width modulator ‘describing function’ is much more involved to determine despite the simplicity of the final result being a constant, which was given here without

further discussion. Together with transfer function, G_{vd} , the control voltage, \hat{v}_c , to output, \hat{v} , transfer function can be determined.

Through simulations, both at a circuit level (using PECS) and at a transfer function level (using Matlab), the model has been confirmed.

In subsequent labs a hardware implementation of these circuits will be examined with the final goal of designing an effective compensator for closed loop feedback control.

Chapter 5

Lab 4

Open loop system construction and testing: the Buck dc-to-dc converter

5.1 Objectives

To build and examine the operation of an open loop dc-to-dc buck converter. This system converts a dc voltage level to a lower dc voltage level at high power efficiency. We will examine the performance of this circuit to step load changes. This will establish a baseline level of performance by which the closed loop designs of later labs will be judged.

5.2 Background

The open loop dc-to-dc buck converter circuit to be constructed is shown in Figure 5.1 where the individual function blocks are identified. There are several function blocks that are discussed below:

1. **Buck converter power stage:** This contains the LC filter and load resistance as well as the switches. The single-pole, double-throw switch of the buck converter is implemented using a mosfet and diode combination.

To minimize losses and damping in the buck converter, the components used there should all be high quality. This translates for each component to mean the following:

- (a) Buck converter mosfet: high power mosfet with low $R_{DS(on)}$ value.
- (b) Diode: this needs to be fast reverse recovery power diode. The diode that has been specified is a Schottky diode which features a very

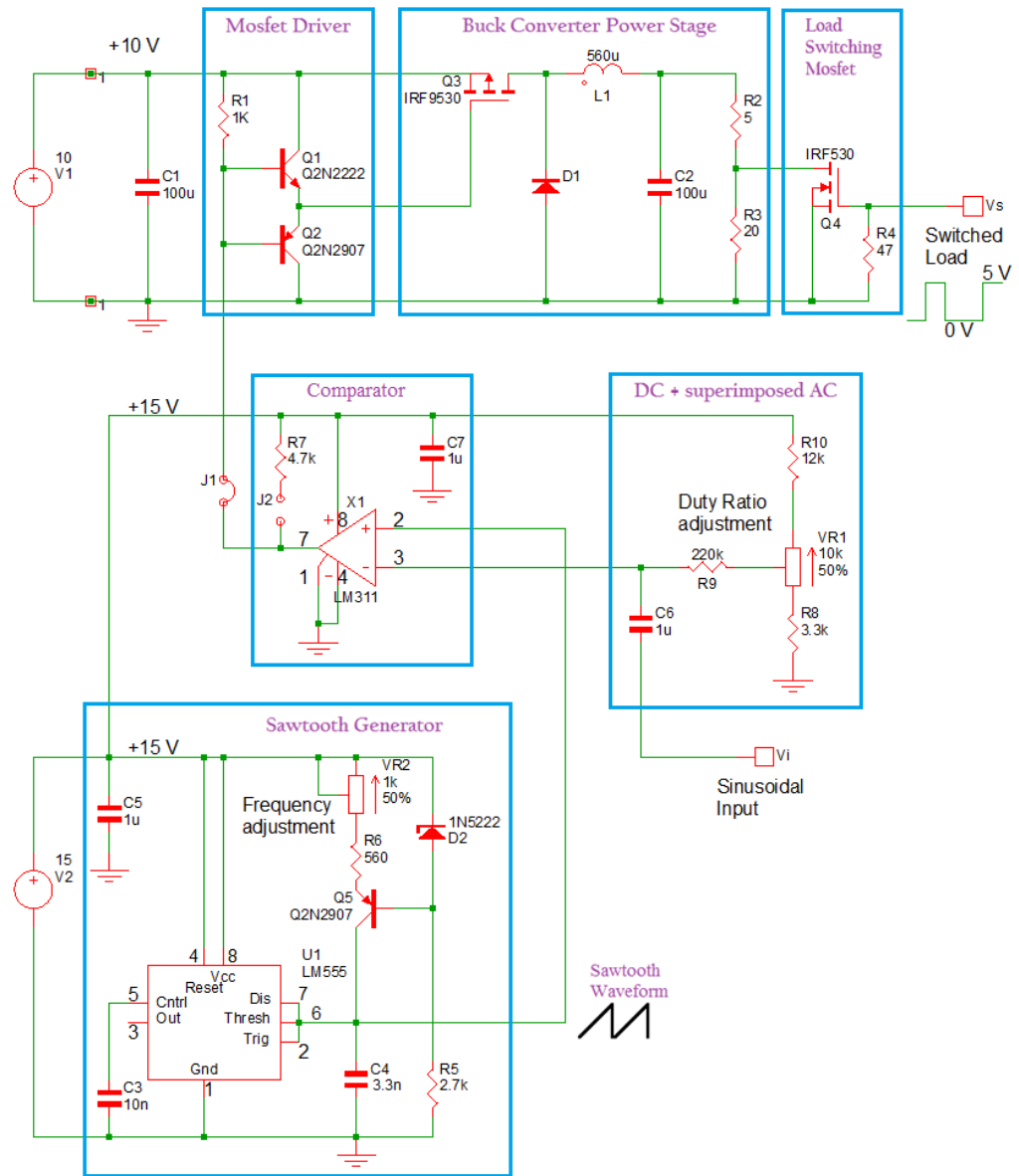


Figure 5.1: Schematic of the open-loop dc-to-dc buck converter.

fast recovery time as well as a lower voltage conduction drop than a silicon diode.

- (c) Inductor: this needs to be a power inductor which can handle an appreciable DC current flow. It should also have a low equivalent series resistance (ESR).
- (d) Capacitor: this should be a high quality capacitor with a low ESR.

Note also that the load resistances need to handle the power that they will dissipate, so the specified power rating needs to be observed.

Further discussion concerning components appears at the end of this lab. Bearing the above in mind, arbitrary substitution of components is to be avoided.

2. **Load switching mosfet:** The load will be switched between two different values in order to examine the stepped-load performance of the converter. Switching of the load is done by turning ON and OFF a mosfet which is in parallel with a portion of the load. This mosfet is controlled by an external square wave generator.
3. **Mosfet driver:** this is required in order to drive the mosfet with sharp step transitions at a high frequency. It interfaces between the comparator and the buck converter power mosfet.
4. **DC + superimposed AC block:** this is simply a variable resistive voltage divider which provides an adjustable DC voltage level. This is fed to the comparator (discussed next).

The adjustable DC allows for the duty ratio, which controls the buck converter, to be set. An external AC signal may be fed into the circuit which superimposes the AC on top of the DC level. The resulting AC modulated duty ratio will be seen in the buck converter output voltage.

5. **Comparator:** The comparator and the sawtooth generator (discussed next) together form the pulse width modulator (PWM). As discussed above, in the present circuit a DC voltage is applied to the inverting input of the comparator and a sawtooth waveform is applied to the non-inverting input. As the sawtooth transitions past the DC voltage level, the comparator will change state producing a rectangular output waveform. The duty ratio of this waveform can be changed by adjusting the DC level using the potentiometer voltage divider.
6. **Sawtooth generator:** A constant current source (comprising transistor Q5, zener diode D2 and resistors R5, R6 and VR2 in Figure 5.1 is used to charge a capacitor (C4 in Figure 5.1) resulting in a voltage ramp appearing across the capacitor. At a predetermined voltage across the capacitor

the LM555 circuit discharges it. This process is ongoing producing a sawtooth waveform. The level of the constant current can be adjusted (by varying VR2 in Figure 5.1) which adjusts the rate at which the threshold discharge voltage is reached thus varying the frequency of the resulting sawtooth.

5.3 Tasks

We will construct the circuit in a step-by-step fashion and test each function block as it is constructed. Note that we will be using two different power supply sources. A 10 V high current source is needed for the buck converter and a 15 V low current source will power the rest of the circuit. A number of tasks below ask that you take a screenshot (or a photo) of the waveform(s) appearing on the oscilloscope. If this is not possible you'll need to neatly sketch the waveform providing voltage level and time period annotations.

1. Construct the sawtooth generator:
 - (a) After you assemble the circuit, use an oscilloscope to monitor the sawtooth voltage waveform across the charging capacitor, C4. Adjust the potentiometer (VR2) to obtain a frequency of 40 kHz .
 - (b) Take a screenshot of the sawtooth waveform. What are the minimum and maximum voltage levels of the sawtooth? The peak-to-peak voltage difference represents V_M in our modelling of the PWM modulator.
2. Construct the comparator and DC + superimposed AC circuits:
 - (a) Assemble the circuit. At this time we will open jumper J1 and close jumper J2. Also the just-constructed sawtooth generator signal should be connected to the non-inverting input of the comparator.
 - (b) Based on the sawtooth waveform you've just observed, what range of duty ratios are you able to achieve by varying the Duty Ratio Adjustment potentiometer (VR1) from one extreme to the other? Show how you can calculate this. Confirm your results in the lab. Note that the duty ratio is defined by the length of the high time of the pulse that appears at the input of the output filter as a ratio of the period. Due to use of a p-channel mosfet in the buck converter power stage a low comparator output corresponds to the mosfet turning ON, which produces the high portion of the pulse, the time interval of which determines the duty ratio, as was just mentioned. Therefore, use this time interval of the low voltage level at the comparator output divided by the square wave period as the duty ratio value.
 - (c) Use an oscilloscope to monitor the output of the LM311 (pin 7). Vary the duty-ratio adjustment potentiometer (VR1) to set a 50% duty

ratio. Check that you have 15 V peak-to-peak square wave. (If so your comparator circuit is functioning properly). Take a screenshot of this waveform.

3. Construct the mosfet driver:

- (a) Connect the two transistors and resistor of the mosfet driver to the 10 V power supply. We will connect the just-constructed comparator circuit by closing jumper J1 and opening jumper J2.
- (b) Use an oscilloscope to monitor the voltage at the common emitter connection of the two transistors. Check that you are seeing a 10 V peak-to-peak square wave. (If so, your driver circuit is functioning properly). Take a screenshot of this waveform.

4. Construct the Buck converter:

- (a) Assemble the Buck converter. (Do not add the load switching mosfet at this time). At this point the whole circuit should be fully functional (without load switching).
- (b) Use an oscilloscope to observe the waveform across the switching diode. Check that you are seeing a 10 V peak-to-peak square wave. Also, you should see sharp rising and falling transitions. (If so, your power mosfet is likely switching properly). Take a photo of this voltage waveform.
- (c) To further verify proper functioning of your power stage, use the oscilloscope to monitor the voltage across the output load, the series combination of the 5 Ω and 20 Ω resistors. Determine its DC value. Is this in accordance with the duty ratio value that you had previously set up?
- (d) For even further verification of the proper operation of the circuit we will inject a small sinusoidal waveform that will modulate the duty ratio, the effect of which will be seen in output voltage. Recall that this was done previously via PECS simulation in Lab 3. You'll be now able to confirm the results obtained there.

To proceed, connect a small-amplitude, low-frequency sinusoidal voltage (100 Hz or lower) to the V_i terminal which feeds into the DC + superimposed AC block. This will superimpose an AC variation on top of the DC voltage appearing at the inverting terminal of the comparator. The frequency is chosen to be much below the LC resonant frequency of the output filter, $f_o = \frac{1}{2\pi\sqrt{LC}} = 673\text{Hz}$. The amplitude should be chosen to be large enough to facilitate measurement of the resulting signal across the output, and no larger so as to not violate the small-signal assumption. Display the input and output signals (as defined next) on the oscilloscope and take a screenshot.

Using the oscilloscope measure the peak-to-peak amplitude of the sinusoidal variation appearing across the converter load. This is the output signal. Now with the oscilloscope, measure the peak-to-peak amplitude of the sinusoid at the inverting input of the comparator (pin 3 of the LM311). This is the input signal. Determine the ratio of the amplitudes of the sinusoids: output amplitude/input amplitude. Can you say what this number represents with respect to the control to output transfer function and what its components are?

Also take note of the phase difference between the output and the input. Is the output in-phase or out-of-phase with the input at the chosen frequency? In this task we simply want to verify whether the signal is being inverted in this signal path. In the next lab we will be closing the loop with negative feedback via a compensator so we will need to determine whether the compensator will need to provide the 180° phase inversion. What is your assessment of this?

5. Switched load performance:

At the present point you should have a fully functional dc-to-dc converter which is converting the 10 V DC input to a 5 V DC output across the series resistor combination of $5\ \Omega$ and $20\ \Omega$ (for a total of $25\ \Omega$) load.

- (a) We will now attach the load switching mosfet circuit and further connect it to a square wave generator which provides a 5 V amplitude signal at a low frequency. The square wave lower voltage level should be 0 V and the higher voltage level should be 5 V. The frequency should be adjusted so as to see the full settled output response of the converter. (A square wave frequency in the range of $50\ Hz$ to $100\ Hz$ should be fine). The mosfet will be turning ON and OFF shorting out the $20\ \Omega$ resistor in the load. Thus the output current will be pulsing between $(\frac{5\ V}{25\ \Omega} =)\ 0.2\ A$ and $(\frac{5\ V}{5\ \Omega} =)\ 1\ A$ (ignoring output voltage changes).
- (b) The varying load current will cause the output voltage to vary. Take a screenshot of the output voltage response. Determine the maximum peak-to-peak output voltage deviation, Δv , and the steady state error, SSE . If necessary, please see the note in Lab. 3 concerning how to determine these quantities.

5.4 Results

- (a) Complete the following tables to summarize your waveform observations in the previous tasks.

Tabulate your results from Task 1b:

	Minimum Value	Maximum Value	V_M = pk-pk voltage
Sawtooth (Task 1b)			

Tabulate your results from Task 2b:

Task 2b	Duty Ratio (D)
Min. D: formula to determine min. D	
Min. D: formula evaluated	
Min. D: measured in lab	
Max. D: formula to determine max. D	
Max. D: formula evaluated	
Max. D: measured in lab	

Tabulate your results from Task 4d:

Input Sine Voltage (Task 4d)	pk-to-pk =
Output Sine Voltage (Task 4d)	pk-to-pk =
Output/Input Ratio (Task 4d)	

- (b) To see how well our prior simulations conform to practice complete the following table:

	From Hardware from Lab 4 (this lab), Task 5b	Using PECS from Lab 3, Task 5	Using Matlab from Lab 3, Task 6e
Δv			
SSE			

- (c) Write your observations concerning the i_o step response results and provide explanations for any discrepancies you see.

5.5 Notes - Buck converter components:

As mentioned earlier, in order to produce a high level of performance, and also to be able to use a system model that is not overly complicated by the need to include many parasitic elements, the components of the buck converter need to be of high quality. We'll take a look at the specific components used in our circuit and take note of some important characteristics seen in their datasheets.

Capacitor: The 100 μF filter capacitor is manufactured by Kemet and has part number A758EK107M1AAAE016. It is polymer aluminum and features a very low ESR (equivalent series resistance) of just 16 $m\Omega$. Generally, the ESR together with the capacitor forms a zero in the transfer functions; however, given these characteristics for the capacitor the zero is positioned at a frequency of approximately $\left(\frac{1}{2\pi R_{ESR}C} \approx \right)$ 100 kHz . This is well beyond the loop bandwidth of our system and consequently may be neglected, as we have done in these labs. The low ESR also minimizes the voltage ripple appearing in the output.

Inductor: The 560 μH inductor is manufactured by Bourns Inc. and has part number 2200HT-561-V-RC. It has a maximum DC resistance of 230 $m\Omega$. We have used this value in our modeling. This is the only parasitic element that is included in our system model. This was done as it has appreciable effect on load regulation, i.e. output voltage variation due to load changes.

Diode: The diode used is manufactured by Sanken and has part number RA13V1. It is a Schottky diode with a fast switching recovery time of less than 500 ns , and has a forward conduction drop of just 360 $mV@2 A$. This compares very favorably compared to the 0.6 V to 0.7 V conduction drop of the average silicon diode. Thus further minimizing parasitic effects.

Mosfet: The IRF9530 mosfet is available from a number of different manufacturers. It has a maximum $R_{DS(on)}$ of 0.30 Ω . Its n-channel complement, the IRF530 mosfet, used to switch the load, has a maximum $R_{DS(on)}$ of 0.16 Ω .

Load resistors: The values and power ratings of the load resistances are 5 Ω , 5 W and 20 Ω , 2 W.

Chapter 6

Lab 5

Closed-Loop Feedback System - Analysis: Analysis and performance of a closed loop dc-to-dc Buck converter system

6.1 Objectives

To build and examine the performance of a closed-loop dc-to-dc buck voltage regulator. A simple compensator is provided. (In the next lab this will be replaced by a better performing, albeit more complicated, compensator designed by the student). First an analysis of the system using the closed loop transfer functions is examined. Subsequently a simulation of the closed loop system is performed both at the transfer function level, using Matlab, and at the circuit level, using PECS. Having first gained an appreciation of the operation of the circuit and the performance of the system through simulation, next a hardware implementation is built and performance is examined in the laboratory and results are compared with those previously obtained by simulation.

6.2 Background

The dc-to-dc converter together with the PWM modulator, introduced in Lab. 3, is now operated in closed loop as depicted in Figure 6.1. As seen in this figure, the output voltage (the controlled variable) is scaled (via the resistive divider comprising R_a and R_b), fed back and compared with a reference voltage producing an error signal. This signal is next processed through the feedback compensator (also known as a frequency compensator) which produces the control signal to the converter via the PWM modulator.

The output voltage of the system, v , is a function of its three inputs: 1) duty ratio, d , 2) input source voltage, v_g , and, 3) load current, i_o . So that

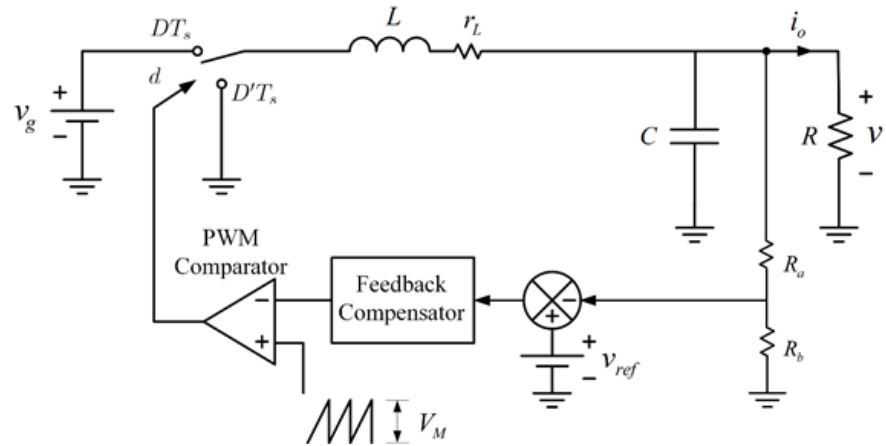


Figure 6.1: Closed loop dc-to-dc buck converter voltage regulator

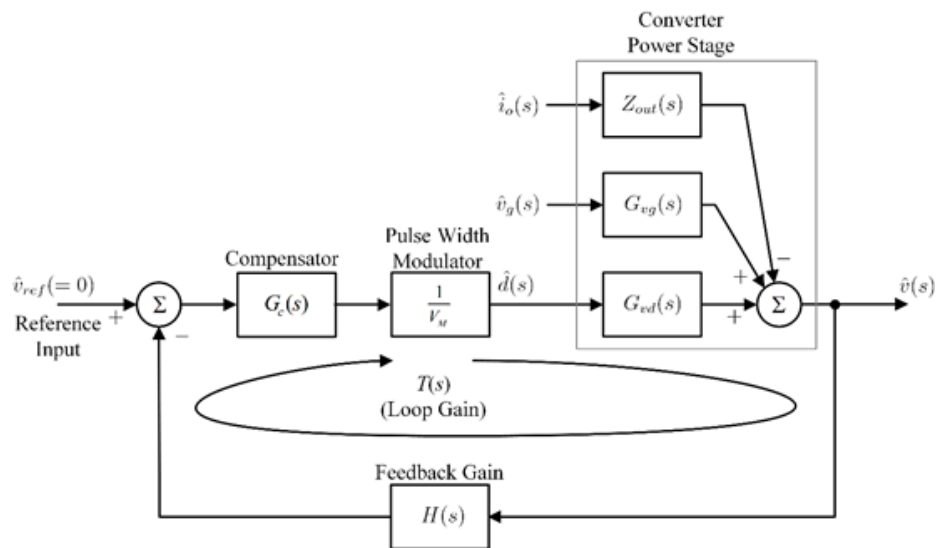


Figure 6.2: Closed loop dc-to-dc buck converter voltage regulator block diagram

$$v = f(d, v_g, i_o)$$

As this function is nonlinear, a linearized model is developed through a Taylor series expansion around a steady state operating point given by $V = f(D, V_g, I_o)$, where capitalized variables refer to steady state values. This results in the following small signal model:

$$\hat{v} = G_{vd}(s) \hat{d} + G_{vg}(s) \hat{v}_g - Z_{out}(s) \hat{i}_o$$

where the caret ‘ $\hat{}$ ’ indicates a small variation around the relevant steady state variable. This represents the power stage small-signal model, and is shown in block diagram form in Figure 6.2. Also shown in this figure are the blocks representing the modulator, compensator and feedback gain. This forms a closed loop feedback system. With the feedback loop closed we have

$$\hat{d} = \frac{G_c(s)}{V_M} (\hat{v}_{ref} - H(s) \hat{v})$$

where \hat{v}_{ref} is the reference voltage (which in a regulator system is constant, so later we will set $\hat{v}_{ref} = 0$). When substituted and rearranged, the above results in:

$$\hat{v} = G_{vref_CL}(s) \hat{v}_{ref} + G_{vg_CL}(s) \hat{v}_g - Z_{out_CL}(s) \hat{i}_o$$

where

$$G_{vref_CL}(s) = \frac{1}{H(s)} \frac{T(s)}{1 + T(s)}$$

$$G_{vg_CL}(s) = \frac{G_{vg}(s)}{1 + T(s)}$$

$$Z_{out_CL}(s) = \frac{Z_{out}(s)}{1 + T(s)}$$

Where $T(s)$ is the loop gain given by:

$$T(s) = \frac{1}{V_M} \cdot G_c(s) \cdot G_{vd}(s) \cdot H(s)$$

Alternatively, these closed loop transfer functions could also have been straightforwardly derived by a block diagram reduction of Figure 6.2.

The open loop transfer functions for the converter and modulator are summarized in Table 6.1.

Table 6.1: Transfer Functions of the buck converter and modulator

$$\Delta(s) = 1 + \frac{s}{\omega_0 Q} + \left(\frac{s}{\omega_0}\right)^2, \quad Q = \frac{\sqrt{LC}}{r_L C + \frac{L}{R}}, \quad \omega_0 = \frac{1}{\sqrt{LC}}$$

$G_{vd} \triangleq \frac{\hat{v}}{\hat{d}}$	$\frac{V_g}{\Delta(s)}$
$G_{vg} \triangleq \frac{\hat{v}}{\hat{v}_g}$	$\frac{D}{\Delta(s)}$
$-Z_{out} \triangleq \frac{\hat{v}}{\hat{i}_o}$	$-\frac{r_L \left(1 + \frac{sL}{r_L}\right)}{\Delta(s)}$
$G_{PWM} \triangleq \frac{\hat{d}}{\hat{v}_c}$	$\frac{1}{V_M}$
$H(s)$ — feedback gain	$\frac{R_b}{R_a + R_b}$

The system to be analyzed and built in this lab is shown in Figure 6.4. This is a modification of the open loop circuit examined in Lab. 4. A number of elements have been added to the open-loop circuit:

1. **Compensator:** this is added to shape the loop gain to assure stability and to obtain adequate phase and gain margins.
2. **Divider:** this is a resistive divider which takes the output converter voltage and divides it to produce a lower level voltage which will be compared to a suitable reference voltage. From a control theoretic point of view, this is denoted as the feedback gain in the standard non-unity gain feedback block diagram.
3. **Voltage reference:** this is the input of a feedback system. Since the system here is a voltage regulator, the input is a constant voltage. This is achieved using a Zener diode.

The compensator used in this lab is the integral compensator, which is shown in its usual form in Figure 6.3. The transfer function for this compensator is given by:

$$G_c(s) = -\frac{1}{RCs}$$

We will add this transfer function and that of the resistive divider, (which corresponds to the $H(s)$ block in Figure 6.2), in closing the loop around the system. The resulting compensated loop gain transfer function of the system will be examined.

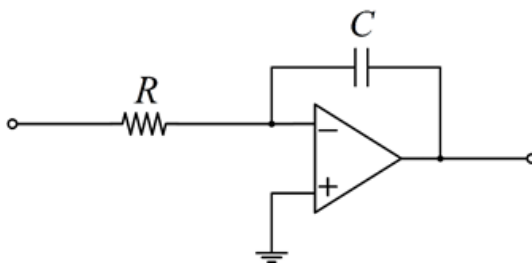


Figure 6.3: Integral compensator implementation, $G_c(s) = -\frac{K_i}{s}$, where $K_i = \frac{1}{RC}$

6.3 Tasks

6.3.1 Pre-Lab

In the following tasks we will analyze the closed-loop system. For the PWM modulator model use the peak-to-peak ramp amplitude value V_M that you measured as part of Lab. 4.

1. Loop gain Analysis:

- (a) **Asymptotic Bode plots:** Sketch and annotate the asymptotic Bode plot (magnitude and phase) for each of the following. The annotations should comprise labeling of break frequencies, slopes of sloping lines, gains of sloping lines and gain and phase levels of zero slope lines.
 - i. Open-loop system, i.e. product of all the transfer functions in the loop, except set $G_c(s) = 1$. Be sure to use a Q value that takes account of the ESR of the inductor. Don't forget to annotate the sketch.
 - ii. Compensator transfer function. Here the integral controller has transfer function, $G_c(s) = -\frac{K_i}{s}$, where $K_i = \frac{1}{RC}$. Don't forget to annotate the sketch.
 - iii. Construct the asymptotic Bode plot using (i) and (ii) for the compensated loop gain. Do this by graphically combining the plots of (i) and (ii). Don't forget to annotate the sketch. Using this plot determine the phase margin and unity gain crossover frequency, and the gain margin (in dB) and the associated -180° phase crossover frequency.

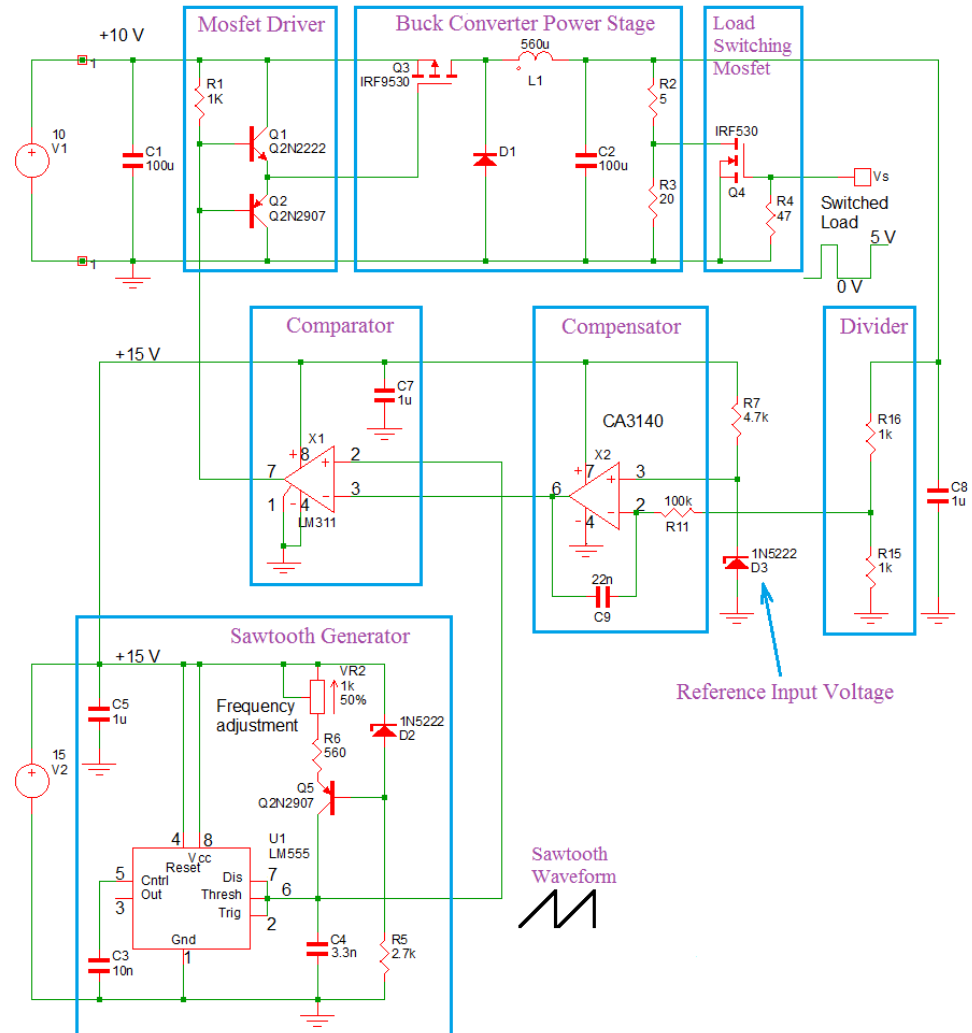


Figure 6.4: Schematic of the closed-loop voltage regulator system.

- (b) **Routh-Hurwitz:** Derive the compensated loop gain, T , with an integral controller with transfer function, $G_c(s) = -\frac{K_i}{s}$. Use the Routh-Hurwitz test to find the range of values of K_i for which the loop is stable. To determine the closed loop stability using Routh-Hurwitz you'll need to examine the coefficients of the denominator polynomial of transfer function $\frac{1}{1+T}$ or $\frac{T}{1+T}$. These transfer functions appear as part of the closed loop transfer functions previously derived. State the value of K_i used in the schematic of Fig. 3.

2. Closed-loop transfer functions - Matlab:

- (a) **Loop gain:** Use the Matlab *margin* command to obtain the Bode plot of the compensated loop gain. Have the plot display frequency in *Hz*. This command will also determine the phase and gain margins of the system, and the associated frequencies, i.e. unity gain crossover frequency and the -180° phase crossover frequency, respectively. These should confirm the numbers that you determined in Task (1.a.iii). Make note of these in your report.
- (b) **Input voltage to output voltage:** Determine the closed-loop input source voltage to output voltage transfer function which we'll denote as: G_{vg_CL} . Use the Matlab *bodemag* command to obtain the magnitude frequency response of this transfer function. On the same plot show the open loop transfer function, G_{vg} . Show the response in *Hz* over a frequency range of *1 Hz* to *10 kHz*.
- (c) **Output current to output voltage:** Determine the closed-loop output current to output voltage transfer function. Since this is the negative of the output impedance, we will denote it as $-Z_{out_CL}$. Use the Matlab *bodemag* command to obtain the magnitude frequency response of this transfer function. On the same plot show the open loop transfer function, $-Z_{out}$. Show the response in *Hz* over a frequency range of *1 Hz* to *10 kHz*.

3. Closed-loop simulations – Matlab:

Note that the following simulations obtained using Matlab are based on the small-signal model only. Thus DC conditions and large signal effects are not modelled and so consequently do not show up in the simulations. However it is recommended that the steady state output voltage be added to the responses below (as was done in Lab. 3), so that an easy comparison can be done with the PECS simulations in Task 4.

- (a) **Input voltage step response:** Use the transfer function previously obtained for G_{vg_CL} to obtain the output step response for a 10% input voltage step, which for a nominal input voltage of *10 V* corresponds to a unit step. If required, refer to Lab. 3 to see how to use the Matlab *lsim* command to perform a unit step response simulation. Obtain the step response plot and determine the maximum

peak-to-peak output voltage deviation, Δv , and steady state error, SSE .

- (b) **Output current step response:** Use the transfer function previously obtained for $-Z_{out_CL}$ to determine the step response for a step load change. If required, refer to Lab. 3 to see how to determine the current load step value and to see how it can be used with the Matlab *lsim* command to perform step response simulation. Obtain the step response plot and determine the maximum peak-to-peak output voltage deviation, Δv , and steady state error, SSE .
4. **Closed-loop simulations – PECS:** We will now use the circuit simulator to obtain the same two closed loop responses discussed in Task (3) above. The PECS simulations will show DC conditions and large signal effects, such as ripple, not available using the Matlab small-signal models.
- (a) **Input voltage step response:** Configure a PECS schematic to obtain the output voltage step response for the input voltage stepping from 10 V (the nominal input voltage level) to 11 V. Obtain the step response plot and determine the maximum peak-to-peak output voltage deviation, Δv , and steady state error, SSE .
 - (b) **Output current step response:** Configure a PECS schematic to obtain the output step response to a switched load, as was previously undertaken in Lab. 3. Obtain the step response plot and determine the maximum peak-to-peak output voltage deviation, Δv , and steady state error, SSE .

6.3.2 In the Lab

5. Build the circuit:

- (a) **Check circuit operation:** Now that the simulations of the previous tasks have been performed, hopefully you've obtained some greater appreciation for the functioning and the performance of the closed loop regulator system. We will now build the system as shown in Figure 6.4. Before performing the load switching test (to be discussed next) make sure your system is functioning properly by checking various voltage waveforms in the circuit. These may include the sawtooth waveform, the voltage waveform across the switching diode (i.e. at the input of the LC filter) as well as the output voltage. The output voltage level should be at a nominal value of 5 V.
- (b) **Output current step response:** For simplicity in the following we will restrict ourselves in the lab to only obtaining the step load response.

Connect a square wave generator which produces a 5 V amplitude signal to the load switching mosfet as shown in the schematic. While

monitoring the output voltage response adjust the frequency of the applied signal to a maximum required to fully see the evolving response before the next switching event occurs. A frequency in the range of 50 Hz to 100 Hz should be adequate. Take a screen shot of this response. Determine the maximum peak-to-peak output voltage deviation, Δv , and steady state error, SSE .

6.3.3 Post-Lab

6. In your report include the following tables to succinctly summarize your results.
- (a) In the following table, summarize results of your loop analysis obtained by
- i. the asymptotic Bode plot method, and,
 - ii. Matlab confirmation.

	Phase Margin ϕ_{PM} (degrees)	Unity Gain Crossover, f_c (kHz)	Gain Margin G_{GM} (dB)	Phase Crossover, f_{GM} (kHz)
Asymptotes				
Matlab				

- (b) To compare the step response results obtained by the three approaches summarize them in the following table:

	Matlab Simulation	PECS Simulation	Lab
v_g : step Δv			Unavailable
v_g : step SSE			Unavailable
i_{out} : step Δv			
i_{out} : step SSE			

- (c) Write your observations concerning these results.
7. The next two tasks (i.e. Tasks (8) and (9)) are optional. Consider doing either one or both. They examine the performance of the system using alternative compensators. In Task (8), the performance of a proportional compensator i.e. $G_c(s) = -K_p$, where K_p is a constant, is examined. In Task (9), a lead compensator, where $G_c(s) = -K \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{\omega_p}}$ where ω_z and ω_p are zero and pole frequencies, respectively, such that $\omega_z < \omega_p$, is examined.

The performance with these controllers feature one striking qualitative difference with that of the integral controller for the stepped input tests

undertaken here. Can you state what that is? (Hint: do a quick simulation using PECS, using the compensator circuits below along with the R and C values suggested there, and see what that is.)

6.4 Optional Tasks - Alternative compensators

Having already tackled tasks (1) to (6) for the case of integral control, it is straightforward to examine the case of proportional and lead control.

6.4.1 Proportional Control

8. The proportional controller is shown in Figure 6.5. In PECS and in the hardware, one needs only to swap out the capacitor of the integral controller with an appropriately sized resistor. To achieve a gain of ten, we will set $R_2 = 1M\Omega$, (given that $R_1 = 100k\Omega$).

Repeat tasks (1) to (6), to examine the performance of a proportional controller.

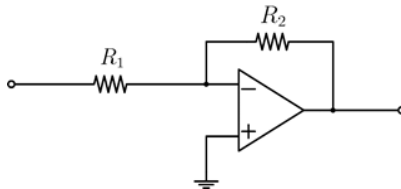


Figure 6.5: Proportional compensator implementation, $G_c(s) = -K_p$, where $K_p = \frac{R_2}{R_1}$

6.4.2 Lead Control

9. The lead controller is shown in Figure 6.6.

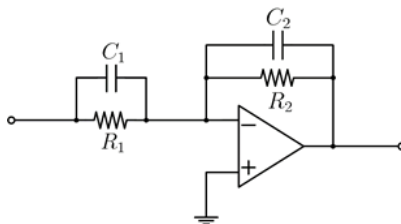


Figure 6.6: Lead compensation implementation, $G_c(s) = -K \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{\omega_p}}$, for $\omega_z < \omega_p$.

The zero and pole frequencies are given by $\omega_z = \frac{1}{R_1 C_1}$ and $\omega_p = \frac{1}{R_2 C_2}$ and $K = \frac{R_2}{R_1}$. Use $R_1 = 22 \text{ k}\Omega$, $R_2 = 820 \text{ k}\Omega$, $C_1 = 4.7 \text{ nF}$ and $C_2 = 12 \text{ pF}$. Repeat tasks (1) to (6), to examine the performance of the lead controller.

6.5 Note

The op-amp used in the compensator, i.e. the CA3140, was chosen since, as stated in the datasheet, it has a “common mode input voltage capability down to 0.5 V below the negative supply terminal” which is more than adequate for our use of a 2.5 V reference voltage at one of its input terminals. Other op-amps do not fair so well so one needs to bear this in mind in seeking a substitute for this device.

Chapter 7

Lab 6

Closed Loop Feedback System - Design: Design of an effective feedback compen- sator for a closed loop dc-to-dc converter system

7.1 Objectives

This is the final in this series of six labs culminating in the design of an effective frequency compensator for a practical feedback system. There are four main objectives:

1. Design of a feedback compensator using the asymptotic Bode plot methodology.
2. Verification of the design before implementation in the lab. This is done both at the transfer function level using Matlab, and at the circuit level using PECS.
3. Building the complete feedback system in the lab and subsequently testing it.
4. To appreciate the effectiveness of the design, a comparison with previous implementations is made. Three implementations are compared:
 - (a) Open loop (Lab 4)
 - (b) Closed loop with integral compensation (Lab. 5)
 - (c) Closed loop with 'dominant pole plus lead' compensation (Lab. 6 (this lab))

7.2 Background

For this lab we start with the circuit implementation of Lab. 5. The integral compensator in the Lab 5 circuit will be replaced with a ‘dominant pole plus lead’ compensator that will be designed in this lab. The compensator has a transfer function given by:

$$G_c(s) = -\frac{\omega_0 \left(1 + \frac{s}{\omega_1}\right) \left(1 + \frac{s}{\omega_2}\right)}{s \left(1 + \frac{s}{\omega_3}\right)}$$

A circuit implementation of this transfer function is shown in Figure 7.1. This compensator enables the loop gain to exhibit a very high gain at low frequencies as well as an extended bandwidth while achieving desired phase and gain margins. The design of the compensator is to be undertaken using the asymptotic Bode plot method which straightforwardly approaches the problem of effective loop shaping.

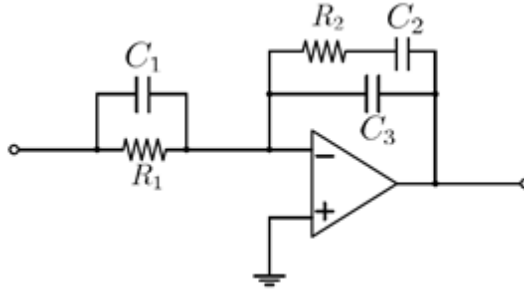


Figure 7.1: ‘Dominant pole plus lead’ compensator to be designed in this lab.

The poles and zeros of the compensator in terms of component values are given by:

$$\omega_0 = \frac{1}{R_1(C_2 + C_3)}$$

$$\omega_1 = \frac{1}{R_2C_2}$$

$$\omega_2 = \frac{1}{R_1C_1}$$

$$\omega_3 = \frac{1}{R_2 \frac{C_2C_3}{C_2+C_3}}$$

In Lab. 5 we implemented the integral controller where the compensator transfer function is given by (ignoring sign) $G_c(s) = \frac{K_i}{s}$. Also, (as an optional task), an opportunity was given to examine proportional control where $G_c(s) = K_p$.

These compensators may be seen as specific components of the three term PID (Proportional – Integral – Derivative) compensators. If we consider the general PID compensator we have

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s$$

where the third (derivative) term has now been added. Note however that this third term, on its own, is non-causal and therefore not realizable. To remedy this, it is customary to associate a high frequency pole with this term. This results in the following for the total PID compensator

$$G_c(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{1 + \frac{s}{a}}$$

where a represents the added high frequency pole. The above may be arranged in quotient form

$$G_c(s) = \frac{(K_p + aK_d)s^2 + (aK_p + K_i)s + aK_i}{s(s + a)}$$

Thus the general PID compensator is comprised of two real poles (with one at zero frequency, the integral term) and two zeros, which are not necessarily real. Comparing this with the ‘dominant pole plus lead’ compensator transfer function, we see the same pole/zero features except that the ‘dominant pole plus lead’ compensator has real valued zeros.

7.3 Tasks

7.3.1 Pre-Lab

1. **Design of feedback compensator:** Design a ‘dominant pole plus lead’ compensator for the system. The compensator should achieve a phase margin $\geq 45^\circ$ and gain margin ≥ 10 dB. The loop gain bandwidth (i.e. the unity gain crossover frequency) should be set at $\frac{1}{8}$ of the switching frequency, i.e. $\frac{40 \text{ kHz}}{8} = 5 \text{ kHz}$. Note that use of this compensator, which features a pole at zero, assures a performance characteristic of zero steady state error to step inputs. Fully document your design procedure.

Use the asymptotic Bode plot method to design your compensator. This methodology involves using asymptotic Bode plot construction of the desired loop gain from which simplified equations may be derived for use in the design process. It does not inherently rely on trial and error iteration and so you are requested not to use such an approach. In the text by Tymerski & Rytkenon the methodology is discussed for the suggested compensator for the buck regulator in a section entitled “Dominant Pole with Lead Compensation”. There are two design

variations discussed there; one being better than the other. (So choose wisely).

Determine the compensator parameters from simplified equations derived from your asymptotic plots. As a final step in the design, implement the compensator as a circuit. The needed equations for this are given above. For the model of the PWM modulator use a peak-to-peak ramp amplitude value that you measured in Lab. 4. Clearly note this value in your report.

2. **Compensator design verification:** With the compensator design completed in Task (1) we will now verify the design before implementing it in the lab. This verification is similar to that previously done in Lab. 5 for a different compensator. The Matlab code developed there can be reused here.

In the following tasks we will analyze the closed-loop system using Matlab and PECS.

Closed-loop transfer functions - Matlab:

- (a) **Loop gain:** Determine the loop gain transfer function and use the Matlab *margin* command to obtain the Bode plot of this loop gain. Have the plot display frequency in *Hz*. The command will also obtain the unity gain crossover and -180° phase crossover frequencies and the phase and gain margins of the system. Make note of these in your report.
- (b) **Input voltage to output voltage:** Determine the closed-loop input source voltage to output voltage transfer function which we'll denote as: G_{vg_CL} . Use the Matlab *bodemag* command to obtain the magnitude frequency response of this transfer function. On the same plot show the open loop transfer function, G_{vg} . Display frequency in *Hz* over a range of *1 Hz* to *10 kHz*.
- (c) **Output current to output voltage:** Determine the closed-loop output current to output voltage transfer function. Since this is the negative of the output impedance, we will denote it as $-Z_{out_CL}$. Use the Matlab *bodemag* command to obtain the magnitude frequency response of this transfer function. On the same plot show the open loop transfer function, $-Z_{out}$. Display frequency in *Hz* over a range of *1 Hz* to *10 kHz*.

3. Closed-loop simulations – Matlab:

Note that the following simulations obtained using Matlab are based on the small-signal model. The DC conditions and large signal effects are not modelled and consequently do not show up in the simulations.

- (a) **Input voltage step response:** Use the transfer function obtained above for G_{vg_CL} to obtain the step response for the 10% input voltage step. As the nominal input voltage 10 V, this implies a unit step change. If required, refer to Lab 3 to see how to use the Matlab *lsim* command to perform step response simulation. As shown in Lab 3 add the steady state average voltage to this simulation so as to ease the comparison with the PECS simulation to follow. Obtain the step response plot and determine the maximum peak-to-peak output voltage deviation, Δv , and steady state error, SSE .
- (b) **Output current step response:** Use the transfer function obtained above for $-Z_{out_CL}$ to obtain the step response for a step load change. If required, refer to Lab. 3 to see how to determine the current load step value and to see how it can be used with the Matlab *lsim* command to perform step response simulation. As with the input voltage step simulation add the average output voltage to the simulation to simplify the comparison with the PECS simulation which follows. Obtain the step response plot and determine the maximum peak-to-peak output voltage deviation, Δv , and steady state error, SSE .

4. Closed-loop simulations – PECS:

We will now use the circuit simulator to obtain the same two closed loop responses discussed in task 3 above. The PECS simulations will show DC conditions and large signal effects, such as ripple, not available using the Matlab small-signal models.

- (a) **Input voltage step response:** Configure a PECS schematic to obtain the input voltage step response to steps from 10 V to 11 V back to 10 V, as done in previous labs. Obtain the step response plot and determine the maximum peak-to-peak output voltage deviation, Δv , and steady state error, SSE .
- (b) **Output current step response:** Configure a PECS schematic to obtain the output voltage response for step load changes from 25 Ω to 5 Ω back to 25 Ω . Obtain the step response plot and determine the maximum peak-to-peak output voltage deviation, Δv , and steady state error, SSE .

7.3.2 In the Lab

Now that you've confirmed your design with both Matlab and PECS, we can confidently build the circuit.

5. Build and test the new compensator:

- (a) **Add new compensator:** Replace the integral compensator of Lab 5 with your newly designed compensator and confirm that the circuit is functioning properly with this compensator. Check that the output is at a constant 5 V level. Do not operate load switching at this time.
- (b) **Output current step response:** Apply the load switching signal and observe the output voltage response. Adjust the frequency of load switching, if necessary. Take a screen shot of the output voltage response. Determine the maximum peak-to-peak output voltage deviation, Δv , and steady state error, SSE .

7.3.3 Post-Lab

Results:

- 6. In your report include the following tables to succinctly summarize your results.
 - (a) In the following table, summarize results of your loop design obtained by
 - i. the asymptotic Bode plot method, and,
 - ii. Matlab confirmation

	Phase Margin ϕ_{PM} (degrees)	Unity Gain Crossover, f_c (kHz)	Gain Margin G_{GM} (dB)	Phase Crossover, f_{GM} (kHz)
i) Asymptotes				
ii) Matlab				

- (b) Summarize the performance of your design. We will also take the opportunity to compare the results with those obtained from previous labs.

Three implementations will be compared:

- i. Open loop (Lab 4)
- ii. Closed loop with integral compensation (Lab. 5)
- iii. Closed loop with ‘dominant pole plus lead’ compensation (Lab. 6, this lab)

The results will be summarized by completing the following table:

ϕ_{PM} : phase margin obtained from Matlab *margin* command

f_c : unity gain crossover frequency obtained from Matlab *margin* command

Δv : maximum peak-to-peak output voltage variation, result may be obtained from the LAB, PECS or MATLAB.

SSE : steady state error voltage result obtained from the LAB, PECS or MATLAB.

	Open loop* (Uncompensated) (from Lab. 4)	Integral Compensator, (from Lab. 5)	Integral + lead Compensator, (from Lab. 6, this lab)
Compensator Transfer Function $G_c(s)$	No compensator		
ϕ_{PM} (degrees) MATLAB			
f_c (kHz) MATLAB			
i_{out} step: Δv (mV) LAB			
i_{out} step: Δv (mV) PECS			
i_{out} step: Δv (mV) MATLAB			
i_{out} step: SSE (mV) LAB			
i_{out} step: SSE (mV) PECS			
i_{out} step: SSE (mV) MATLAB			
v_g step: Δv (mV) PECS			
v_g step: Δv (mV) MATLAB			
v_g step: SSE (mV) PECS			
v_g step: SSE (mV) MATLAB			

*This is the open loop (uncompensated) system. That is, unlike the other two systems feedback is *not* applied. It is included here to see how well feedback control is able to improve on open loop control.

- (c) Write your observations concerning these results.

Appendix A

List of Parts

No.	Item
C1	100 μ F capacitor electrolytic
C2	100 μ F capacitor Kemet A758EK107M1AAAE016
C3	10 nF capacitor
C4	3.3 nF capacitor
C5	1 μ F capacitor
C6	1 μ F capacitor
C7	1 μ F capacitor
C8	1 μ F capacitor
C9	22 nF capacitor
D1	Power Schottky Diode Sanken RA 13V1
D2	1N5222 2.5 V Zener Diode
D3	1N5222 2.5 V Zener Diode
L1	560 μ H inductor Bourns Inc. 2200HT-561-V-RC
R1	1 $k\Omega$ $\frac{1}{16}$ W Resistor
R2	5 Ω 5 W Power Resistor
R3	20 Ω 2 W Power Resistor
R4	47 Ω $\frac{1}{16}$ W Resistor
R5	2.7 $k\Omega$ $\frac{1}{16}$ W Resistor
R6	560 Ω $\frac{1}{16}$ W Resistor
R7	4.7 $k\Omega$ $\frac{1}{16}$ W Resistor
R8	3.3 $k\Omega$ $\frac{1}{16}$ W Resistor
R9	220 $k\Omega$ $\frac{1}{16}$ W Resistor
R10	12 $k\Omega$ $\frac{1}{16}$ W Resistor
R11	100 $k\Omega$ $\frac{1}{16}$ W Resistor
R15	1 $k\Omega$ $\frac{1}{16}$ W Resistor
R16	1 $k\Omega$ $\frac{1}{16}$ W Resistor
Q1	Q2N2222 NPN Transistor
Q2	Q2N2907 PNP Transistor
Q3	IRF 9530 P-Channel MOSFET
Q4	IRF 530 N-Channel MOSFET
Q5	Q2N2907 PNP Transistor
U1	LM555 Timer
VR1	10 $k\Omega$ Variable Resistor
VR2	1 $k\Omega$ Variable Resistor
X1	LM311 Voltage Comparator
X2	CA3140 Op Amp

Appendix B

PECS

Power Electronics Circuit Simulator

B.1 PECS Overall Description

PECS is a circuit simulator which features high simulation accuracy as well as fast simulation times. It is particularly well suited to circuits which contain ideal switches, as is the case with power electronic circuits. It was developed at Portland State University [1], [2], [3].

The PECS software actually consists of two distinct programs:

1. PECS, provides graphical schematic capture and simulation functions. At the end of a successful simulation run an output file is written to the hard drive.
2. PECSPLOT, provides plotting and graphical post processing of selected outputs designated in the PECS schematic. A file previously written to the hard drive is read by this program. Note that PECSPLOT is automatically called by the PECS program at the end of a simulation and, as such, the user need not call this program directly. However, a user may use this program directly if so desired on a previously stored PECS output file.

Figure B.1 shows the PECS user interface with a displayed circuit schematic. PECSPLOT is discussed more fully in the next section..

B.2 PECS Usage

The following steps are taken to simulate a circuit and produce an output plot:

1. Build the desired circuit schematic in PECS.

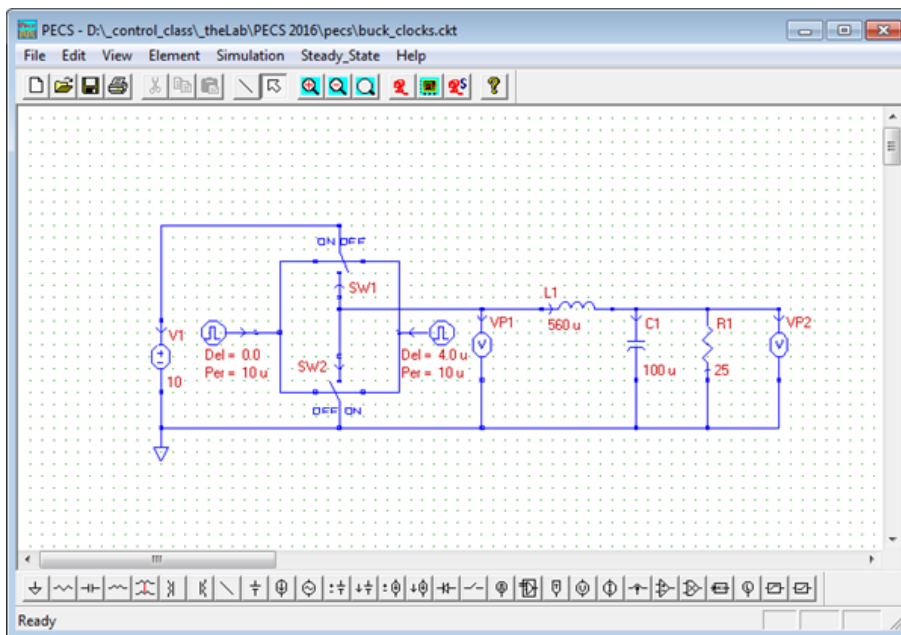


Figure B.1: The PECS program showing a circuit schematic.

2. Set simulation parameters.
3. Run the simulation.
4. Select the desired output(s) to plot.

These will be expanded on next:

B.2.1 Building a circuit schematic in PECS

Element selection: Select the desired component to place from the component selection bar at the bottom by left mouse clicking on the element or from the *Element* menu at the top. Drag the element to the appropriate position on the page and then left click again to place it. This action will also bring up another copy of the element which may subsequently also be placed. Clicking the right button will eliminate the copy (or the original selected component) if not needed.

Component Rotation: Rotation of components in 90 degree clockwise increments can be achieved by pressing the *space bar* on the keyboard.

Wiring between components: Connections between components is made by first left clicking on the *wire* icon available in both the bottom component selection bar and in the top menu bar of PECS. This action causes the mouse cursor to change to a pen shape. Vertical and/or horizontal lines can now be

drawn between element nodes to connect them. Note that only connections between nodes of an element will constitute a valid connection. To exit this mode of operation one can either right click the mouse or left click on the *arrow* icon in the top menu icon bar.

Selection of previously placed components: After a component has been placed in the schematic it can be selected by either left clicking on it or by depressing the left mouse button in the vicinity of the component and then, with the left button depressed, dragging the mouse over the component which will bring up a rectangle. Releasing the button will select any component which was fully enclosed in the rectangular area. The selected components will now appear in red.

Movement of previously placed components: First select components to be moved using the procedure discussed immediately above. These selected components may now be moved by depressing the left button on any of the selected components and, while depressed, dragging to the new desired position. After the components have been moved, deselect them as discussed next.

Deselection of components: Previously selected components can be deselected by either right clicking the mouse or left clicking anywhere away from the area occupied by the selected components.

Deletion of selected components: A component or a number of components can be simultaneously deleted from a schematic by first selecting them and then right clicking on the scissors icon in the top component icon bar or pressing the *Delete* button on the keyboard.

Outputs: Output variables to be plotted can be any voltage or current in the schematic. These are identified in the schematic by an appropriately placed *voltage port* element or a *current port* element. A voltage port is placed in parallel across the two points in the schematic for which the voltage will output. On the other hand a current port is placed in series with line in which the current will be monitored.

Ground node: PECS requires that a ground node be placed in the circuit. This is the left most icon in the bottom element icon bar. Output ports: PECS requires that at least one voltage or current port appear in the circuit. Ports indicate to PECS which variables are to be stored for latter plotting.

Components available: The following are the most used components available in PECS. (This does not cover the complete set of components available in PECS).

- *Basic passive elements:* R (resistor), C (capacitor) and L (inductor)
- *Sources:* VDC (DC voltage), IDC (DC current), VAC (AC voltage)

- *Ports*: Vport (voltage port) and Iport (current port)
- *Switches*: Sw (controlled switch) and D (diode)
- *Switch control elements*: Clk (Clock) and (MOD) PWM Modulator
- *Miscellaneous*: OPAMP (operational amplifier), current sensor (no symbol name), ground node (no symbol name) and wire (no symbol name)

These available components will be further discussed in greater detail below. There are range of other components that PECS features, such the four controlled sources, but these will not be discussed here as they will not be used.

Assigning component parameter values: Any time after a component has been placed it may be assigned with specific values. Double left clicking on the placed component will bring up the component dialog window. Note that PECS will not run a simulation if any component values haven't been adequately assigned. An error message will be given alerting to this fact and unassigned components will be highlighted in red. There is no need to specify component names as these are automatically assigned.

Saving a schematic: A schematic can be saved to disk at any time. A default name of *Untitled.ckt* is assigned and so a more appropriate named should be substituted. Note that during the construction/editing of a schematic, PECS periodically saves a backup file which is given the name *backup.ckt*. If need be, this file can be read back into PECS by using the *File* → *Open ...* menu item.

B.2.2 Setting simulation parameters

Before a simulation is run, appropriate simulation parameters should be assigned. This is achieved by bringing up the appropriate dialog window through the menu item *Simulation* → *Parameters ...* . This dialog window appears as shown in Figure B.2.

The simulation parameters are:

- i *Final Time*: the total length of time of the simulation.
- ii *Step Size*: this should be increased/decreased to decrease/increase, respectively, the number of points at which the response will be determined. It is important to use a small enough value here so as to smoothen the output. Note that plotting between response points is achieved by using straight line segments. The points in time in the simulation interval that are actually saved to hard disk is specified by the *Save interval* parameters:
- iii *Start time*: this is the time from which the output is saved to disk.
- iv *End time*: this is the time to which the output is saved to disk. Generally this should be the same as the simulation *final time*.

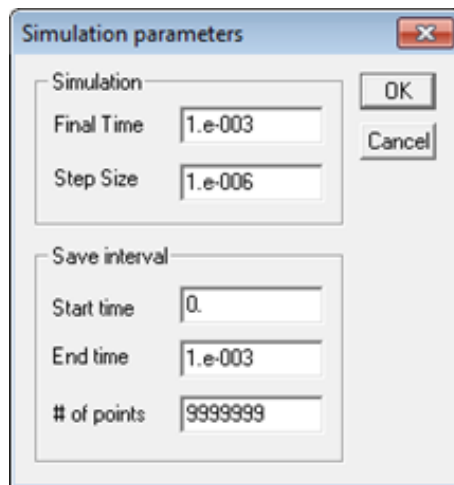


Figure B.2: Simulation parameters dialog window showing the default values.

v # of points: This specifies the number of evenly-spaced points that are saved in the save interval. This value rarely needs to be changed from its default as generally it is best to maximize the number of points saved which will lead to a smoother looking plot.

B.2.3 Running the simulation

After the above two steps have been completed, one can run the simulation by left clicking on the script R icon in the top icon bar. Alternatively, this can be initiated through the menu by selecting *Simulation* → *Run*.

Assuming the schematic has been constituted properly the simulation will start and a progress bar at the bottom of PECS will appear. At the end of the simulation a dialog window appears indicating the total real time used to perform the simulation. When this dialog is dismissed PECS PLOT will automatically be called.

B.2.4 Selecting the desired output(s) to plot

When PECS PLOT appears a number of outputs will appear in the left column. These output having previously been specified by the voltage and current ports in the PECS schematic. Moving the output port names from the left to the right column specifies the plot to be viewed. This is achieved by double clicking the left mouse button on the selected output variable. Alternatively, one can move an output variable from the left column by selecting it (or a number of outputs) and then left clicking on the *Add* button between the columns. Any number of output variables can be displayed in one plot but generally it is best that

different variables appear in their own plot so as to accommodate scaling for the plot, which is automatic. This is achieved by using the *Add* menu item in PECSPLPOT.

B.3 Elements

B.3.1 Basic passive elements: R (resistor), C (capacitor) and L (inductor)

Resistor:

The schematic symbol of a resistor is shown in Figure B.3. The resistance value is the only parameter value that needs to be specified the dialog window shown in Figure B.4.



Figure B.3: Resistor schematic symbol

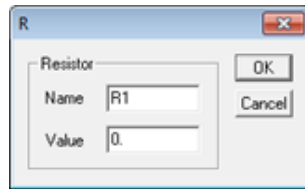


Figure B.4: Resistor dialog window

Capacitor:

The schematic symbol of a capacitor is shown in Figure B.5. The capacitor value and an initial voltage are the parameters that need to be specified in the dialog window shown in Figure B.6.



Figure B.5: Capacitor schematic symbol

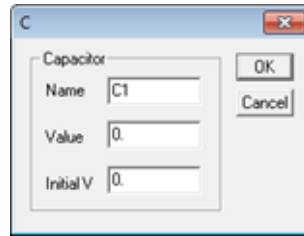


Figure B.6: Capacitor dialog window

Inductor:

The schematic symbol of an inductor is shown in Figure B.7. The inductor value and an initial current are the parameters that need to be specified in the dialog window shown in Figure B.8.



Figure B.7: Inductor schematic symbol

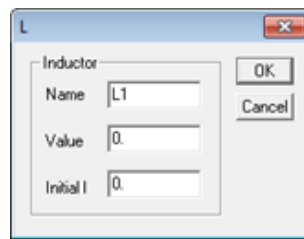


Figure B.8: Inductor dialog window

B.3.2 Sources: VDC (DC voltage), IDC (DC current), VAC (AC voltage)

DC Voltage Source:

The schematic symbol of a DC voltage source is shown in Figure B.9. The value of the voltage is a user specified parameter that is entered in the dialog window, shown in Figure B.10.



Figure B.9: DC Voltage Source schematic symbol

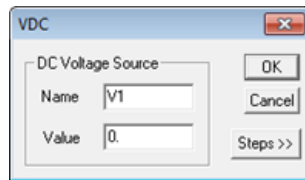


Figure B.10: DC Voltage Source

Voltage sources can be stepped in value at a limited number of user specified times. Clicking on the Steps button will open the bottom section of the dialog window where one is able to enter up to five time and associated step values which the voltage source will assume during the simulation. This is shown in Figure B.11. If step values are present, the dialog window will appear fully expanded on initiation.

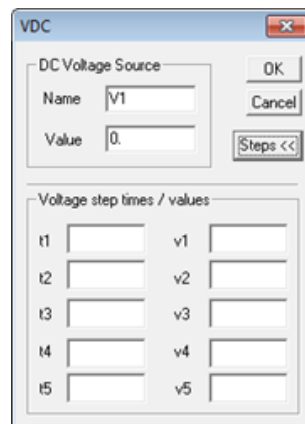


Figure B.11: DC Voltage Source, showing voltage step times and values

DC Current Source:

The schematic symbol of a DC current source is shown in Figure B.12. The value of the current is a user specified parameter that is entered in the dialog window, shown in Figure B.13.



Figure B.12: DC Current Source schematic symbol

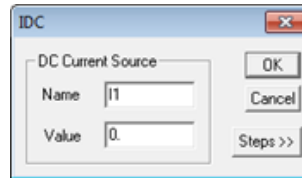


Figure B.13: DC Current Source dialog window

Current sources can be stepped in value at a limited number of user specified times. Clicking on the *Steps* button will open the bottom section of the dialog window where one is able to enter up to five time and associated step values which the current source will assume during the simulation. This is shown in Figure B.14. If step values are present, the dialog window will appear fully expanded on initiation.

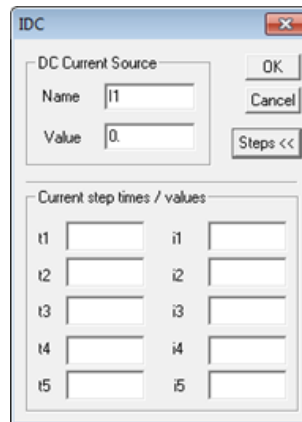


Figure B.14: DC Current Source dialog window, showing voltage step times and values

AC Voltage Source:

The schematic symbol of an AC voltage source is shown in Figure B.15. This element produces a sinusoidal voltage waveform. The peak amplitude (volts), frequency (Hz) and phase (degrees) comprise the parameters that are entered into the dialog window, see Figure B.16, which will specify its characteristics.



Figure B.15: AC Voltage Source schematic symbol

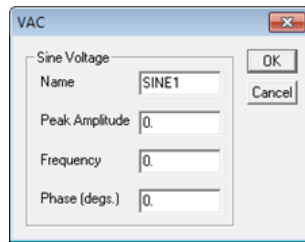


Figure B.16: AC Voltage Source dialog window

B.3.3 Ports: Vport (voltage port) and Iport (current port)

Voltage Port:

The schematic symbol of a voltage port is shown in Figure B.17. The purpose of voltage port is to monitor the voltage across two points in the circuit being simulated. This voltage will be saved in the output file which will be read by the plotting program. There are no parameters required for this element, however as with most of the other elements the name may be changed from its default via the dialog window, see Figure B.18.



Figure B.17: Voltage port schematic symbol

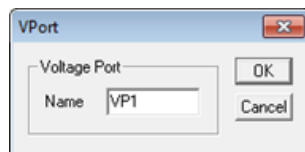


Figure B.18: Voltage Port dialog window

Current Port:

The schematic symbol of a current port is shown in Figure B.19. The purpose of a current port is to monitor the current in a branch in the circuit being simulated. This current will be saved in the output file which will be read by the plotting program. There are no parameters required for this element, however as with most of the other elements the name may be changed from its default via the dialog window, see Figure B.20.

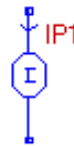


Figure B.19: Current Port schematic symbol

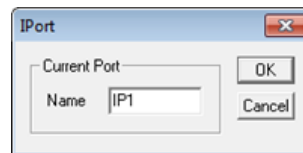


Figure B.20: Current Port dialog window

B.3.4 Switches: Sw (controlled switch) and D (diode)

Switch:

The schematic symbol for a switch is shown in Figure B.21.

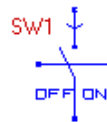


Figure B.21: Switch schematic symbol

A switch features two electrical terminals and two control terminals which are labeled *ON* and *OFF*. The connection to the control terminals is not electrical. Only specific switch control elements may be connected to these terminals. The switch control elements that are discussed here are:

1. Clock
2. PWM Modulator

Note that once a switch turns ON or OFF by the (conceptual) impulses driving it, it remains in this state until the time instant of a counter-acting control signal.

Double clicking on the switch symbol will bring up the dialog window which is shown in Figure B.22. An initial state can be selected and is provided to define the starting point for a cycle for use in the accelerated steady state determination feature available in PECS. (This feature is not discussed here.) If the selected initial state is not correct, the program will alter the state.

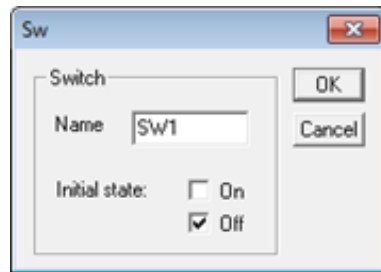


Figure B.22: Switch dialog window

Diode:

The schematic symbol of a diode is shown in Figure B.23, with the dialog window shown in Figure B.24. As with the switch element, there is facility to stipulate the initial state of the diode which is used to define the starting state at the beginning of a switching cycle. Should this state be incompatible with current circuit conditions the simulator will change the diode state.



Figure B.23: Diode schematic symbol

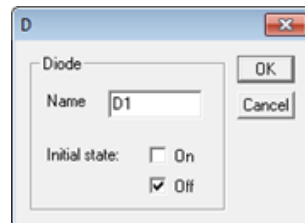


Figure B.24: Diode dialog window

B.3.5 Switch control elements: Clk (Clock) and (MOD) PWM Modulator**Clock:**

The schematic symbol for a clock is shown in Figure B.25.



Figure B.25: Clock schematic symbol

A clock has only one terminal which can only be connected to a switch ON or OFF node. The parameters of a clock element are:

1. Delay time
2. Period

With reference to Figure B.26, we see that after an initial delay of Delay seconds the switch is then turned *ON (OFF)*. Following this initial signal, subsequent signals are issued every Period seconds. Thus, if the clock element is connected to the *ON (OFF)* terminal of the switch, these subsequent signals repeatedly turn the switch *ON (OFF)*.

The Delay and Period parameters are entered into the dialog window as seen in Figure B.27.

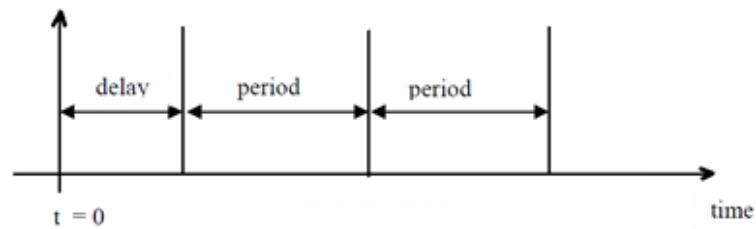


Figure B.26: Delay and period parameters shown pictorially

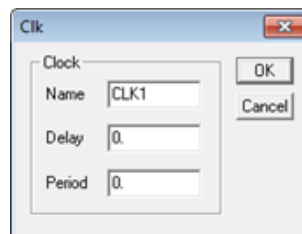


Figure B.27: Clock dialog window

Pulse Width Modulator:

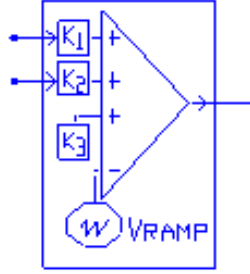


Figure B.28: Pulse Width Modulator schematic symbol

The schematic symbol of a PWM modulator is shown in Figure B.28. This symbol pictorially shows a four input comparator and a sawtooth generator. Whenever the sum of the four inputs (three are added and one is subtracted) becomes negative, the output (which can only be connected to a switch control terminal) will issue an appropriate turn ON or OFF signal. Two external inputs are provided to the modulator. These inputs are fed through gain blocks K_1 and K_2 . The third input to the comparator is a constant K_3 . K_1 , K_2 and K_3 are all input parameters of the modulator element. The last comparator input is from a (conceptual) sawtooth generator where the peak-to-peak voltage amplitude, V_{pk-pk} , period of the sawtooth are the final two parameters of this element.

If we consider signals K_1 and K_2 to be connected to the K_1 and K_2 inputs, respectively, then an output switching signal is issued from the modulator whenever

$$K_1 \cdot v_1 + K_2 \cdot v_2 + K_3 - V_{RAMP} < 0$$

The PWM dialog window is shown in Figure B.29 where the input gains, K_1 , K_2 and K_3 , and sawtooth parameters, the peak to peak voltage (V_{pk-pk} and *period*) can be entered.

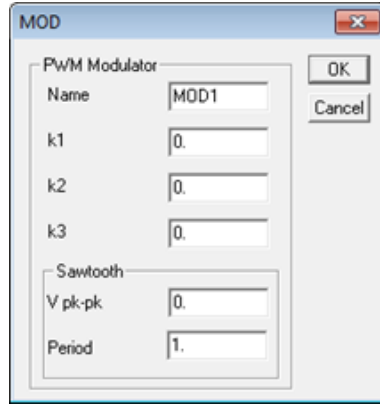


Figure B.29: Pulse Width Modulator dialog window

B.3.6 Miscellaneous: OPAMP (operational amplifier), current sensor (no symbol name), ground node (no symbol name)

Operational Amplifier:

The schematic symbol of an op-amp is shown in Figure B.30.

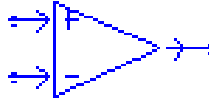


Figure B.30: Operational Amplifier schematic symbol

The model of the op-amp used in PECS is shown in Figure B.31. There are three parameters that characterize this model: 1) input resistance, R_i , 2) output resistance, R_o , and 3) op-amp gain. The values of these parameters can be changed via the element dialog window which is shown in Figure B.32. Generally the default values will need not be changed.

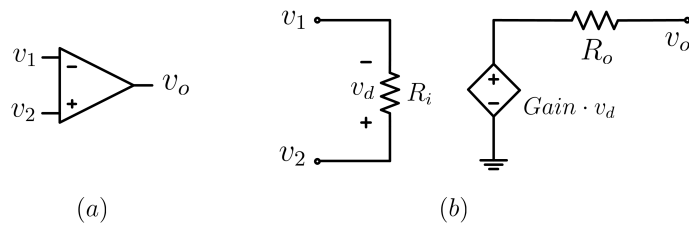


Figure B.31: (a) Op-amp, and (b) its corresponding model

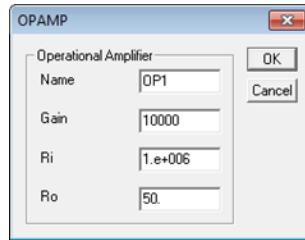


Figure B.32: Operational Amplifier dialog window, showing the default parameter values.

Current sensor:

The schematic symbol for a current sensor element is shown in Figure B.33. There are no user defined parameters for this device and so there is no associated dialog window.



Figure B.33: Current sensor schematic symbol

Ground node:

The schematic symbol for a ground node element is shown in Figure B.34. There are no user defined parameters for this device and so there is no associated dialog window.



Figure B.34: Ground node schematic symbol

B.4 PECS PLOT

This section provides a comprehensive overview of PECSLOT, which is the program that PECS calls to plot the waveforms that were generated by simulation in PECS. PECSLOT is automatically called by PECS at the end of a simulation. However, it may also be user initiated which subsequently requires the user to choose the desired data file (which has extension *.plt) which contains the waveform data.

PECSLOT has the following features:

1. Multiple plots, stacked vertically, are possible.
2. Multiple waveforms can appear in each plot.
3. Zooming into a desired time interval is possible.
4. Accurate measurements of time and amplitude are possible.
5. Auto-scaling of the amplitude axis for single and multiple waveform plots.
6. Full customization of font size, waveform and background colors is provided.

B.5 Terminology:

1. Waveform - a signal that is shown within a plot. Multiple waveforms can be shown within the same plot. Necessarily all waveforms share the same scaling of the vertical axis.
2. Plot - A time and amplitude axis pair within which waveforms appear. Waveforms of widely differing amplitudes are best graphed in separate plots with their own individual vertical scaling.

To illustrate these terms, let us consider Figure B.35 which shows PECSLOT displaying a number of waveforms. Specifically there are two plots. The top plot shows the waveforms VP1 in red and VP2 in green. The bottom plot shows only the VP2 waveform. Because the scaling in the top plot is dominated by the amplitude of the VP1 waveform not much detail can be seen for the VP2 waveform. However, this detail is available in the bottom plot where the vertical scaling accommodates for the amplitude of the VP2 alone.

B.6 General use

Before delving into the functionality offered by PECSLOT under the different menu items, we'll first look at the overall general use.

After PECS completes a simulation it stores the data points to the hard drive and subsequently calls PECS PLOT which reads the file. The waveform names are then displayed to the user who then selects how they are to be plotted, that is, the number of plots desired and the grouping of waveforms. This waveform selection dialog window is shown in Figure B.36.

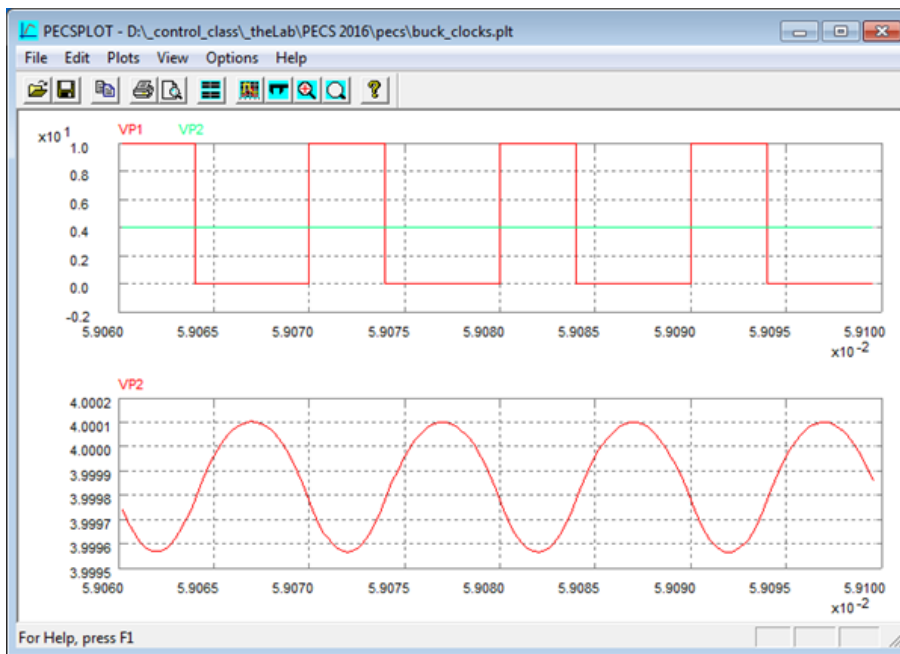


Figure B.35: PECS PLOT displaying two plots. The top plot show two waveforms and the bottom show only one.

B.7 Menu items

As seen in Figure B.35 the following menu items appear across the top of the application:

1. File menu
2. Edit menu
3. Plots menu
4. View menu
5. Options menu
6. Help menu

The functionality provided under each of these menu items will be presented next.

B.7.1 File menu commands

The File menu offers the following commands:

1. *Open*: Opens a PECS PLOT file.
2. *Reload*: Reloads an opened PECS PLOT file.
3. *Save*: Saves an opened PECS PLOT file using the same file name. Only currently displayed waveforms will be saved, other waveforms appearing in the file will be deleted.
4. *Save As*: Saves a PECS PLOT file to a specified file name.
5. *Print*: Prints the currently displayed plot(s).
6. *Print Preview*: Displays the current plot(s) seen on the screen as it would appear printed.
7. *Print Setup*: Selects a printer and printer connection.
8. *Exit*: Exits PECS PLOT

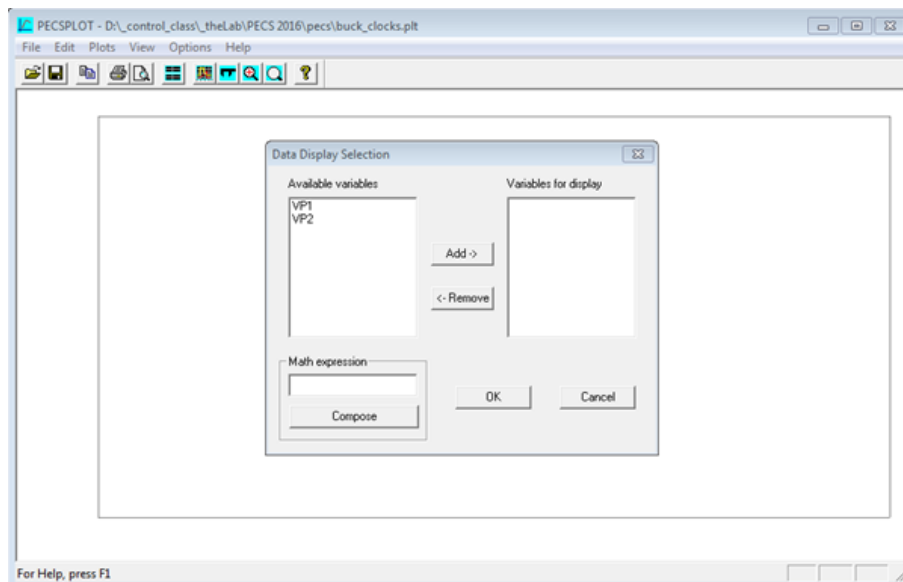


Figure B.36: Selection of waveforms dialog window.

B.7.2 Edit menu commands

The Edit menu offers the following commands:

1. *Copy to Clipboard*: Copies data from the currently displayed plot(s) to the clipboard.
2. *Edit Title*: Two captions that appear on the top left and top right positions of the printout may be edited. The default are the PECSLOT file name (top left) and the current date and time (top right).

B.7.3 Plots Menu commands

The Plots menu offers the following commands:

1. *Add/Delete Waveform*: Add or delete a waveform to the chosen plot.
2. *Add Plot*: Add an extra plot.
3. *Delete Plot*: Delete a plot.
4. *Measure*: Enable measurement of displayed waveforms.
5. *Max*: Show the maximum value of all waveforms (as well as the times at which they occur) in the chosen plot. Measure mode must first be chosen to un-gray this menu item.
6. *Min*: Show the minimum value of all waveforms (as well as the time at which they occur) in the chosen plot. Measure mode must first be chosen to un-gray this menu item.

Selecting the *Plots* → *Measure* menu item (or alternatively clicking on the *calipers* icon in the icon bar) will bring up the measurement display screen as shown in Figure B.37. Measurement is undertaken by first left and/or right clicking inside a plot at the desired position(s). Accurate placement of the measurement lines may be obtained using the left and right keyboard arrow keys. This affects movement of the measurement lines on a data point by point basis.

Waveforms may be manipulated mathematically. This is done through the waveform selection dialog which was shown in Figure B.36. The following operators may be used: +, -, *, /, and ^ where '^' refers to exponentiation. The following functions are provided: *sin*, *cos*, *tan*, *sqrt*, *fabs*, *exp*, *atan*, *log* and *log10*. These functions use the standard C language implementations. Each requires a single argument and caution should be exercised for the range of values that the functions will see as some functions may not be defined for some range of values. For example, *sqrt*, *log* and *log10* are not defined for negative values. The constant *pi* (= 3.141592654) is recognized. Also engineering suffixes may be used and are listed in Table B.1.

Table B.1: Suffixes and corresponding values.

Suffix	M	k	m	u	n	p
Value	1E+6	1E+3	1E-3	1E-6	1E-9	1E-12

For example, $10k = 10,000$. Be sure not to use any waveform names that correspond to these suffixes.

Any level of parenthetical expression with "(" and ")" may be used. An example of a valid expression is:

$(I(L2) + \sin(2 * \pi * 10k * v_o)) - 1$, where "I(L2)" and "vo" are valid waveform names.

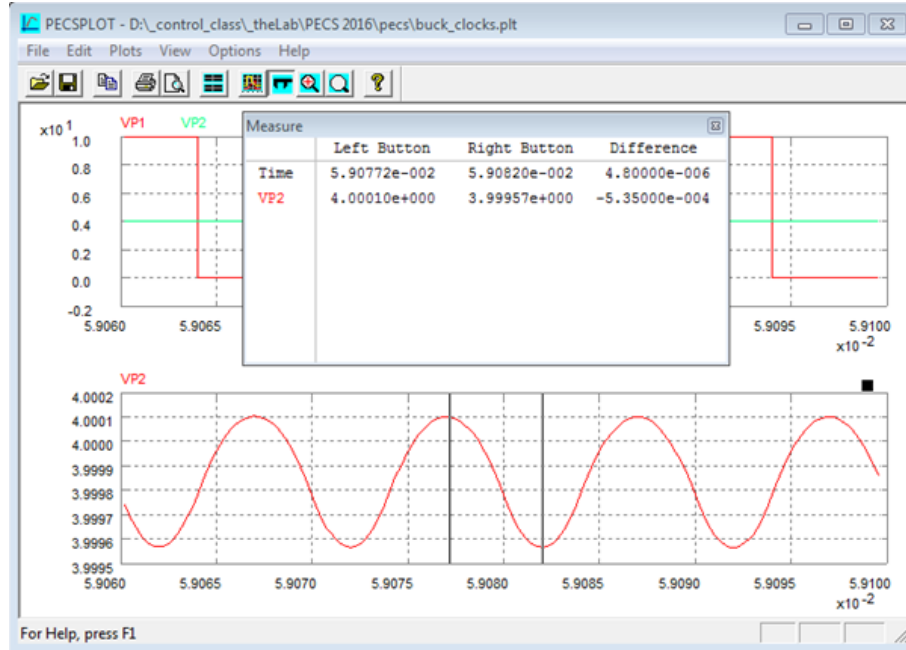


Figure B.37: The measure display dialog usually appears at the lower left corner of the computer screen. Here it has been moved to the position shown for display purposes. Clicking the left and right mouse buttons in the bottom plot area at the appropriate positions results in the vertical lines appearing and the value of the waveform(s), as well as the time values, being displayed. The third column indicates the difference between the two values, which in this case results in a measurement of the peak-to-peak amplitude of the displayed waveform.

B.7.4 View menu commands

The View menu offers the following commands:

1. *Zoom*: Zooms in on the x-axis limits. These limits are established by the user left clicking inside a plot and then dragging to another position. Only the x values of the initial and final positions are used. The y-axis limits of the zoomed plot(s) are automatically scaled commensurate with the currently displayed waveforms in the selected x value range.
2. *Zoom All*: Use maximum x-axis limits given by the data in the PECSLOT file.
3. *Redraw*: Redraw without altering the zoom factor (if any).
4. *Toolbar*: Shows or hides the toolbar.
5. *Status Bar*: Shows or hides the status bar.

B.7.5 Options menu commands

The Options menu offers the following commands, which enable you to customize the plots appearance:

1. *Text Settings*: Change the font style, font size, font color.
2. *Background Color*: Change the background color.
3. *Waveform Color*: Change individual waveform colors. To un-gray this menu item, first select a waveform by left clicking on a waveform title from those appearing above each plot. Note that all plots using the selected waveform color will be changed to the new color.
4. *Grid*: Toggle grid on or off.

B.7.6 Help menu commands

The Help menu offers the following commands, which provide assistance with PECSLOT:

1. *Index*: Offers an index to topics on which help can be obtained.
2. *About*: Displays the version number of PECSLOT as well as contact information.

B.8 References

1. *PECS - Power Electronics Circuit Simulator*, Duwang Li, R. Tymerski, and T. Ninomiya, IEEE Workshop on Computers in Power Electronics (COMPEL '00), Blacksburg, Virginia, July, 2000, pp. 159-165.
2. *PECS - An Efficacious Solution for Simulating Switched Networks with Nonlinear Elements*, Duwang Li, R. Tymerski, and T. Ninomiya, IEEE Power Electronics Specialists Conference (PESC '00), Galway, Ireland, June, 2000, pp. 274-279.
3. *PECS - An Efficient Solution for Simulating Switched Networks with Nonlinear Elements*, D. Li, R.P.E. Tymerski and T. Ninomiya, IEEE Transactions on Industrial Electronics, April 2001, pp. 367-376.