Chapter 16

Lab 6

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

16.1 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))

16.2 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 16.1. 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.

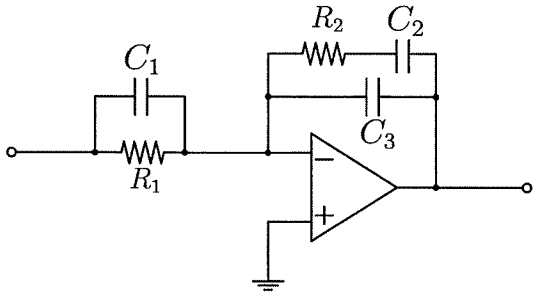

Figure 16.1: 'Dominant pole plus lead' compensator to be designed in this lab.

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

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

$$\omega_1 = \frac{1}{R_2 C_2}$$

$$\omega_2 = \frac{1}{R_1 C_1}$$

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

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.

16.3 Tasks

16.3.1 Pre-Lab

1. **Design of feedback compensator:** Design a 'dominant pole plus lead' compensator for the system. The compensator should achieve a phase margin $\geq 45°$ 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. $\frac{40\ kHz}{8} = 5\ kHz$. 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 & Rytkonen 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.

← YOUR COMPENSATOR COMPONENT VALUES NEED TO BE DIFFERENT THAN THE ONES I USED IN MY LAB 6 QSPICE SCHEMATIC.

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. QSPICE

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*.

16.3.2 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*.

16.3.3 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* (Uncompenstated) (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.