

# **Classical and Modern Control Design: with examples from power electronics**

**Richard Tymerski**

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

**Frank Rytkonen**

Oregon Institute of Technology  
Department of Electrical Engineering and Renewable Energy  
Wilsonville, Oregon, USA

## Acknowledgements

The authors would like to express their sincere gratitude to a number of former students who have greatly contributed to making this book possible. Of particular note, Clint Lieser and Andrew Chuinard ... (To be expounded upon.)

# Preface

This is the preface.



# Contents

Acknowledgements . . . . .	ii
Preface . . . . .	iii
<b>I Classical Control: Fundamentals</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Basic Feedback Configuration . . . . .	3
1.2 Stability - Absolute and Relative . . . . .	4
1.3 Stability Analysis Example . . . . .	8
1.3.1 Matlab Code . . . . .	14
<b>2 System Stability</b>	<b>15</b>
2.1 Stability . . . . .	15
2.2 Stability Analysis by Inspection . . . . .	15
2.2.1 Parameterized Stability Analysis by Inspection . . . . .	16
2.2.2 Examples of Stability Analysis by Inspection . . . . .	16
2.3 Routh-Hurwitz Analysis . . . . .	17
2.3.1 Zero in First Column . . . . .	20
2.3.2 A Row of Zeros . . . . .	20
2.3.3 Parameterized Routh-Hurwitz Analysis . . . . .	21
<b>3 Steady State Error</b>	<b>23</b>
3.1 Unity gain feedback . . . . .	25
3.1.1 Example 1: . . . . .	31
3.2 Non-Unity Feedback . . . . .	33
3.2.1 Example 2: . . . . .	36
3.2.2 Example 3: . . . . .	41
3.2.3 Example 4: . . . . .	43
3.2.4 Example 5: . . . . .	46
3.2.5 Example 6: . . . . .	49
3.2.6 Example 7: . . . . .	51
3.2.7 Example 8: . . . . .	53
3.3 Note: partial fraction containing a complex pole pair . . . . .	58

<b>4</b>	<b>Bode Plots</b>	<b>59</b>
4.1	Simple Gain . . . . .	59
4.2	Pole at Zero . . . . .	60
4.3	Zero at Zero . . . . .	61
4.4	Pole at $\omega_o$ . . . . .	62
4.5	Zero at $\omega_o$ . . . . .	63
4.6	Right Half Plane Zero at $\omega_o$ . . . . .	63
4.7	Complex Pole Pair with Resonant Frequency at $\omega_o$ . . . . .	63
4.8	Complex Zero Pair with Resonant Frequency at $\omega_o$ . . . . .	64
4.9	Composite Transfer Functions . . . . .	65
4.10	Summary of Bode Plots . . . . .	70
<b>5</b>	<b>Compensator Design</b>	<b>73</b>
5.1	Design Procedure . . . . .	73
5.1.1	Introduction . . . . .	73
5.1.2	Uncompensated System . . . . .	74
5.1.3	Proportional Compensated System . . . . .	78
5.1.4	Dominant Pole Compensated System . . . . .	80
5.1.5	Dominant Pole Compensated System with zero . . . . .	82
5.1.6	Dominant Pole Compensated System with zero, improved phase margin . . . . .	84
5.1.7	Lead Compensated System . . . . .	85
5.1.8	Lead Compensated System with integrator and zero . . . . .	87
5.1.9	Summary . . . . .	88
5.1.10	MATLAB Code . . . . .	103
<b>II</b>	<b>Classical Control: Application</b>	<b>109</b>
<b>6</b>	<b>Modelling - Introduction</b>	<b>111</b>
6.1	Introduction . . . . .	111
<b>7</b>	<b>The System</b>	<b>113</b>
7.1	Introduction . . . . .	113
7.2	The Plant: Buck Converter . . . . .	114
7.2.1	Introduction . . . . .	114
7.2.2	Transfer Function Derivations . . . . .	114
7.3	Pulse-width Modulator . . . . .	122
7.4	Summary . . . . .	123
<b>8</b>	<b>Single Loop Voltage Mode Control</b>	<b>127</b>
8.1	Introduction . . . . .	127
8.2	Buck Converter System Models . . . . .	128
8.2.1	General Model . . . . .	128
8.2.2	Simplified System Model . . . . .	128
8.2.3	Design Targets . . . . .	129

8.2.4	Buck Converter Model Analysis . . . . .	129
8.3	Uncompensated System . . . . .	130
8.4	Dominant Pole Compensation . . . . .	134
8.5	Dominant Pole Compensation with Zero . . . . .	142
8.6	Lead Compensation . . . . .	146
8.7	Dominant Pole with Lead Compensation . . . . .	151
8.7.1	Design 1: Zero $f_1 = 500\text{Hz}$ . . . . .	153
8.7.2	Design 2: Zero $f_1 = 150\text{ Hz}$ . . . . .	156
8.8	Extended Bandwidth Design . . . . .	158
8.9	Conclusion . . . . .	166
8.9.1	Compensator Circuits . . . . .	170
8.10	MATLAB Code . . . . .	172
<b>9</b>	<b>Droop and Multi-Loop Control</b>	<b>175</b>
9.1	Introduction . . . . .	176
9.2	Design . . . . .	176
9.2.1	Passive Droop Compensation . . . . .	176
9.2.2	Active Droop Compensation . . . . .	183
9.2.3	Voltage Mode Compensation . . . . .	188
9.3	Summary . . . . .	200
9.4	MATLAB Code . . . . .	200
<b>III</b>	<b>Classical Control: Laboratory</b>	<b>211</b>
<b>10</b>	<b>Introduction to the Labs</b>	<b>213</b>
<b>11</b>	<b>Lab 1</b>	<b>217</b>
11.1	Objectives . . . . .	217
11.2	Circuit #1 . . . . .	217
11.2.1	Tasks . . . . .	218
11.3	Circuit #2 . . . . .	219
11.3.1	Tasks . . . . .	220
11.4	Circuit #3 . . . . .	220
11.4.1	Tasks . . . . .	221
<b>12</b>	<b>Lab 2</b>	<b>223</b>
12.1	Objectives . . . . .	223
12.2	Background . . . . .	223
12.3	Tasks . . . . .	226
12.3.1	PECS simulation . . . . .	226
12.3.2	Matlab simulation – results check . . . . .	227

<b>13 Lab 3</b>	<b>231</b>
13.1 Objectives . . . . .	231
13.2 Background . . . . .	231
13.3 System Transfer Functions . . . . .	234
13.4 Tasks . . . . .	238
13.5 Note . . . . .	245
13.6 Results . . . . .	246
13.7 Postscript . . . . .	246
<b>14 Lab 4</b>	<b>249</b>
14.1 Objectives . . . . .	249
14.2 Background . . . . .	249
14.3 Tasks . . . . .	252
14.4 Results . . . . .	254
14.5 Notes - Buck converter components: . . . . .	256
<b>15 Lab 5</b>	<b>257</b>
15.1 Objectives . . . . .	257
15.2 Background . . . . .	257
15.3 Tasks . . . . .	261
15.3.1 Pre-Lab . . . . .	261
15.3.2 In the Lab . . . . .	264
15.3.3 Post-Lab . . . . .	265
15.4 Optional Tasks - Alternative compensators . . . . .	266
15.4.1 Proportional Control . . . . .	266
15.4.2 Lead Control . . . . .	266
15.5 Note . . . . .	267
<b>16 Lab 6</b>	<b>269</b>
16.1 Objectives . . . . .	269
16.2 Background . . . . .	270
16.3 Tasks . . . . .	271
16.3.1 Pre-Lab . . . . .	271
16.3.2 In the Lab . . . . .	273
16.3.3 Post-Lab . . . . .	274
<b>Appendix A List of Parts</b>	<b>277</b>
<b>Appendix B Lab Grading Sheets</b>	<b>279</b>
<b>IV Modern Control</b>	<b>291</b>
<b>17 Introduction</b>	<b>293</b>



<b>18 System Analysis</b>	<b>297</b>
18.1 The Ćuk Converter . . . . .	297
18.2 The Ćuk Converter Model . . . . .	299
18.2.1 Analysis of Inductors with Mutual Coupling and Equivalent Series Resistances . . . . .	299
18.2.2 The State Space Averaged Model . . . . .	300
18.2.3 Component Values . . . . .	303
18.3 Ćuk Converter Open Loop Performance . . . . .	303
18.4 Controllability and Stabilizability . . . . .	307
18.5 Observability and Detectability . . . . .	307
18.6 Controlling the Ćuk Converter . . . . .	308
18.6.1 Time Domain Specifications . . . . .	308
18.6.2 Frequency Domain Specifications . . . . .	308
18.6.3 Control Effort Constraints . . . . .	309
18.7 MATLAB Code . . . . .	311
<b>19 Pole Placement</b>	<b>315</b>
19.1 Pole Placement via Ackermann’s Formula . . . . .	316
19.2 Ćuk Converter with State Feedback Compensator . . . . .	316
19.3 MATLAB Code . . . . .	321
<b>20 Integral Action</b>	<b>325</b>
20.1 Adding Integrators . . . . .	325
20.2 Ćuk Converter with State Feedback and Integral Compensator . . . . .	327
20.3 MATLAB Code . . . . .	331
<b>21 State Estimation</b>	<b>333</b>
21.1 Full-Order State Estimators . . . . .	333
21.2 Full-Order Estimator-Based Compensator . . . . .	334
21.3 Reduced-Order State Estimators . . . . .	338
21.4 Reduced-Order Estimator-Based Compensator . . . . .	341
21.5 MATLAB Code . . . . .	345
<b>22 Linear Quadratic Optimal Control</b>	<b>353</b>
22.1 Linear Quadratic Regulators . . . . .	353
22.2 Ćuk Converter with LQR Compensator . . . . .	354
22.3 Linear Quadratic Gaussian Regulators . . . . .	357
22.4 Ćuk Converter with LQG Compensator . . . . .	358
22.5 Control with LQG/LTR Compensators . . . . .	360
22.6 MATLAB Code . . . . .	363
<b>23 Compensator Order Reduction</b>	<b>369</b>
23.1 Model Reduction of the LQGI/LTR Compensator . . . . .	369
23.2 A Reduced-Order LQGI/LTR Compensator . . . . .	373
23.3 MATLAB Code . . . . .	379

<b>24 Compensator Implementation</b>	<b>385</b>
24.1 MRLQGI/LTR Compensator Construction . . . . .	385
24.2 MRROLQGI/LTR Compensator Construction . . . . .	386
<b>25 Power Electronic Circuit Simulation</b>	<b>391</b>
25.1 Simulating the Controlled Ćuk Converter in PECS . . . . .	391
<b>26 Conclusion</b>	<b>395</b>
<b>Appendix C PECS</b>	<b>397</b>
C.1 PECS Overall Description . . . . .	397
C.2 PECS Usage . . . . .	397
C.2.1 Building a circuit schematic in PECS . . . . .	398
C.2.2 Setting simulation parameters . . . . .	400
C.2.3 Running the simulation . . . . .	401
C.2.4 Selecting the desired output(s) to plot . . . . .	401
C.3 Elements . . . . .	403
C.3.1 Basic passive elements: R (resistor), C (capacitor) and L (inductor) . . . . .	403
C.3.2 Sources: VDC (DC voltage), IDC (DC current), VAC (AC voltage) . . . . .	404
C.3.3 Ports: Vport (voltage port) and Iport (current port) . . . . .	407
C.3.4 Switches: Sw (controlled switch) and D (diode) . . . . .	408
C.3.5 Switch control elements: Clk (Clock) and (MOD) PWM Modulator . . . . .	410
C.3.6 Miscellaneous: OPAMP (operational amplifier), current sensor (no symbol name), ground node (no symbol name) . . . . .	412
C.4 PECS PLOT . . . . .	414
C.5 Terminology: . . . . .	414
C.6 General use . . . . .	414
C.7 Menu items . . . . .	415
C.7.1 File menu commands . . . . .	416
C.7.2 Edit menu commands . . . . .	417
C.7.3 Plots Menu commands . . . . .	417
C.7.4 View menu commands . . . . .	419
C.7.5 Options menu commands . . . . .	419
C.7.6 Help menu commands . . . . .	419
C.8 References . . . . .	420

Part I

Classical Control:  
Fundamentals



# Chapter 1

## Introduction

### 1.1 Basic Feedback Configuration

The standard single loop feedback block diagram configuration is shown in Figure 1.1. There are three basic blocks  $G(s)$ ,  $G_c(s)$  and  $H(s)$ , the plant, compensator and feedback transfer functions, respectively. These are discussed below. But first a straightforward analysis leads to the overall input to output transfer function of this configuration:

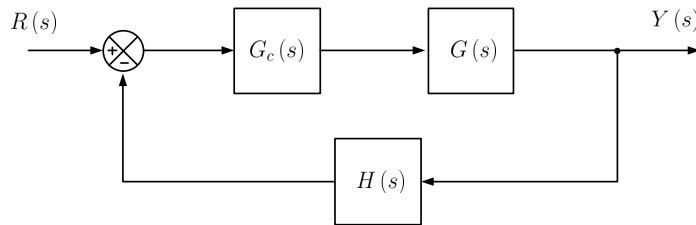


Figure 1.1: Feedback System Block Diagram

$$\frac{Y(s)}{U(s)} = \frac{G(s)}{1 + G(s)H(s)} \quad (1.1)$$

$$= \frac{1}{H(s)} \frac{G(s)H(s)}{1 + G(s)H(s)} \quad (1.2)$$

The product  $G(s)H(s)$  (more strictly,  $-G(s)H(s)$ ) is termed the loop gain as it is the product of the gains around the feedback loop. Equation (1.3) is termed

the closed loop gain of the system. The advantages that feedback provides is derived from the fact that as long as the magnitude of the loop gain is large, that is,  $|G(s)H(s)| \gg 1$ , then the closed loop gain is determined by  $H(s)$ . However, the price to be paid for this is the possibility that the denominator polynomial, known as the characteristic polynomial, i.e.  $1 + G(s)H(s)$  may vanish at some frequency resulting in the closed loop gain becoming unbounded, that is unstable. The purpose of the compensator  $G_c(s)$  therefore is to provide loop shaping to avoid this instability condition.

From this discussion we can therefore summarize the purpose of each of the three blocks:  $G(s)$  is the plant, which implements the operational function that is desired; the feedback block  $H(s)$  (under the condition of high loop gain) sets the value of closed loop gain

$$\frac{Y(s)}{U(s)} \approx \frac{1}{H(s)} \quad \text{for } |G(s)H(s)| \gg 1$$

and the compensator  $G_c(s)$  block improves the stability and performance of the closed loop system.

The performance of the closed loop system will be assessed by examining the response to step inputs. Specifically, with reference to figure 12.2 we will look at the rise time, settling time and percentage overshoot. Rise time,  $t_r$ , is defined as the time it takes for the step response to go from 10% to 90% of the final value. Settling time,  $t_s$ , refers to the time it takes for the response to remain in a band of  $\pm 5\%$  of the final value. The percentage overshoot (OS) is determined by the following formula

$$OS = \frac{B - A}{A} \times 100 \quad (1.3)$$

where  $A$  and  $B$  represent the largest overshoot value and final value, respectively.

## 1.2 Stability - Absolute and Relative

As discussed above, the question of absolute stability of a closed loop system, that is whether it is stable or not, can be answered by whether a frequency exists for which satisfies the characteristic equation:

$$1 + G(s)H(s)|_{s=j\omega} = 0 \quad (1.4)$$

These frequencies are the 'poles' of the closed loop system. The locations of pole frequencies can be mathematically determined by a root finding procedure. Stability can be subsequently assessed by examining whether all of these complex valued quantities have a negative real parts. If so, the system is said to be absolutely stable.

Once absolute stability is assessed, then the question of how stable the system is, that is, its relative stability may be asked. In this regard, the gain (GM) and phase margins (PM) are used. This is generally done in the context of Bode plots, which show the magnitude and phase response of the loop gain over a range of frequencies which include the  $-180^\circ$  phase crossover frequency and the unity gain crossover frequency, as shown in figure 1.3. As a rule of thumb, a gain margin of  $> 10$  dB and phase margin of between  $45^\circ$  to  $60^\circ$  are deemed desirable.

Note also that (under certain mild assumptions concerning the loop gain) the phase margin may be used to determine the stability of the closed loop system. The phase margin test for absolute stability requires that

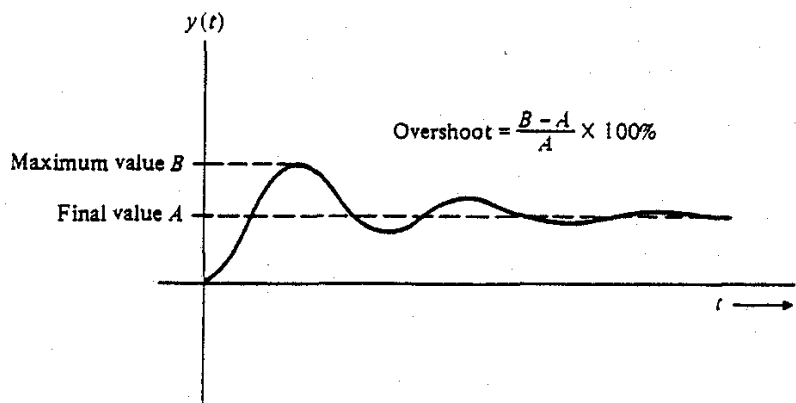
$$\text{Phase Margin Test for Stability: } PM > 0 \implies \text{stability} \quad (1.5)$$

The gain margin does enter into the absolute stability consideration. This may be seen as rather unusual and so an example will be presented in the next section demonstrating this as well as tying up a number of other concepts presented here.

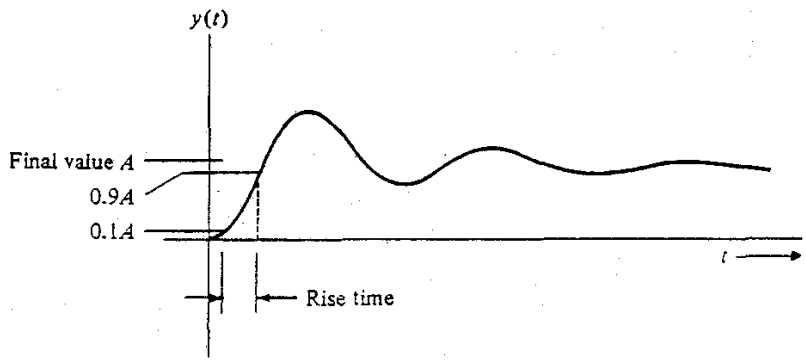
It should be emphasized that the stability of the closed loop system which has transfer function  $\frac{G}{1+GH}$  can be assessed by looking at a property the loop gain which has transfer function  $GH$ , specifically the phase margin.

The design of compensators may be approached in several ways. In this book the use of Bode plots to shape the loop and obtain desirable gain and phase margins is demonstrated.

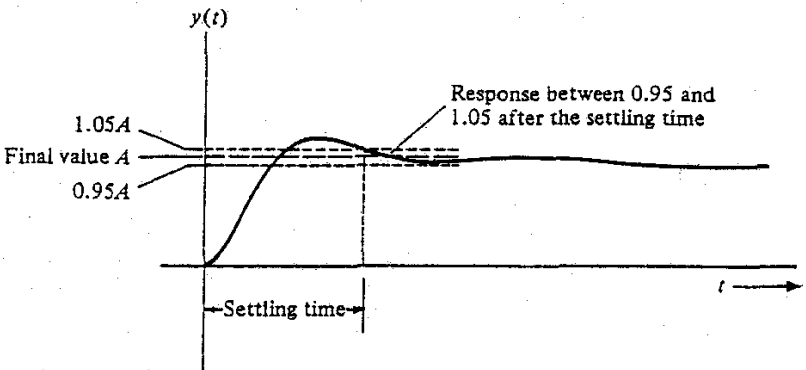
The design of the compensator may be approached several ways. In this book the use of Bode plots to shape the loop is demonstrated



(a)



(b)



(c)

Figure 1.2: Step Response  
 © Richard Tymerski and Frank Rytkonen, 2017



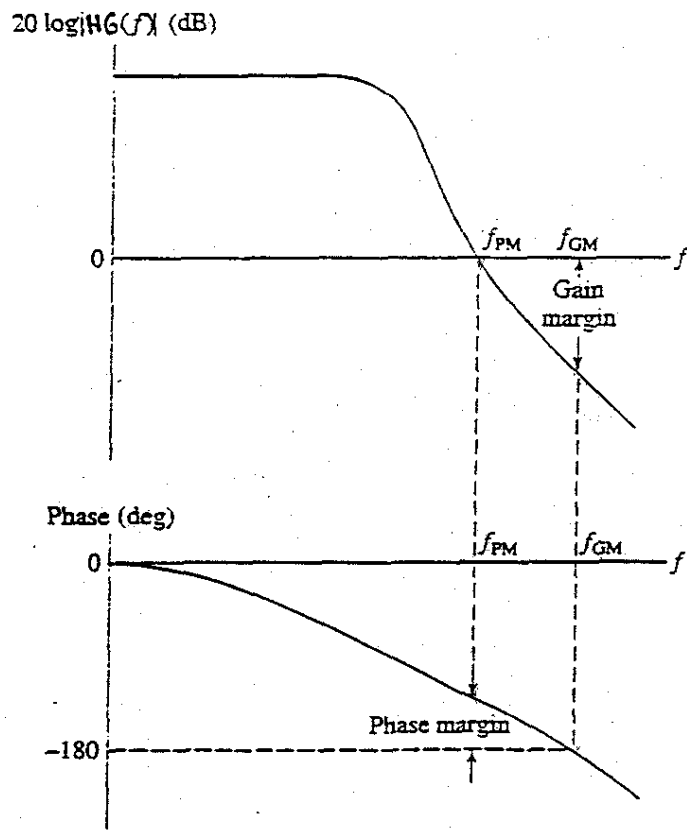


Figure 1.3: Gain and Phase Margins

### 1.3 Stability Analysis Example

Consider a system with the following loop gain

$$T(s) = A \frac{\left(1 + \frac{s}{\omega_z}\right) \left(1 + \frac{s}{\omega_1}\right)}{s^3}$$

where  $A = 300$ ,  $\omega_z = 40\text{rad/s}$ , and  $\omega_1 = 1\text{rad/s}$

Using the methodology that has been developed to this point, the loop gain,  $T(s)$  will be analyzed to determine whether the system is stable.

The stability analysis begins by constructing an asymptotic bode plot of the given system, as shown in Figure 1.4.

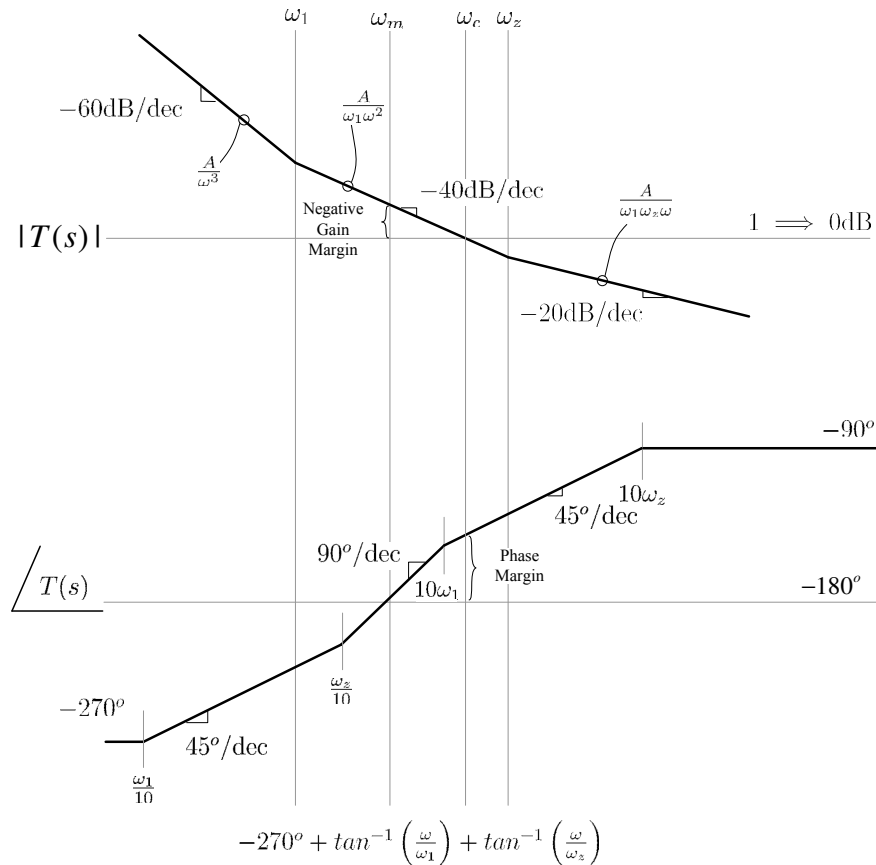


Figure 1.4: Bode Plot: System Loop Gain

Using the asymptotic bode plot, the phase margin is calculated by utilizing the magnitude to determine the crossover frequency:

$$\frac{A}{\omega_c^2} = 1 \text{ or } \omega_c = \sqrt{A} = 17.3 \text{ rad/s}$$

With the crossover frequency defined, the margin is calculated by inserting the frequency value into the loop gain phase equation:

$$PM = 180^\circ - 270^\circ + \tan^{-1} \left( \frac{\omega_c}{\omega_z} \right) + \tan^{-1} \left( \frac{\omega_c}{\omega_1} \right)$$

$$PM = 180^\circ - 270^\circ + \tan^{-1} \left( \frac{17.3}{40} \right) + \tan^{-1} \left( \frac{17.3}{1} \right) = 20^\circ$$

To determine the gain margin, the phase equation is used to determine the frequency in which the phase is  $-180^\circ$ .

$$-180^\circ = -270^\circ + \tan^{-1} \left( \frac{\omega_m}{\omega_z} \right) + \tan^{-1} \left( \frac{\omega_m}{\omega_1} \right)$$

Solving the phase equation for  $\omega_m$ ,  $\omega_m = 6.32 \text{ rad/s}$

Inserting the computed value of  $\omega_m$  into the magnitude equation, the gain margin is calculated as follows:

$$|T(j\omega_m)| = \frac{A}{\omega_m^2} = \frac{300}{6.32^2} = 7.5$$

$$GM = -20 \log_{10} (7.5) = -17.5 \text{ dB}$$

Using MATLAB to verify the asymptotic bode plot, Figure 1.5 confirms the phase and gain margin analysis, with a gain margin and phase margin error of 1.1% and 6.1% respectively, due to the approximations of the asymptotic magnitude plot.

With a positive phase margin, and a negative gain margin, additional criterion are necessary to determine if the system is stable. Another method used to determine the stability of a system is the Routh-Hurwitz stability criterion.

Before the Routh-Hurwitz criterion can be applied to the presented system, a closed loop transfer function must be derived for the system loop gain. One possible realization of the closed loop system is presented in Figure 1.6.

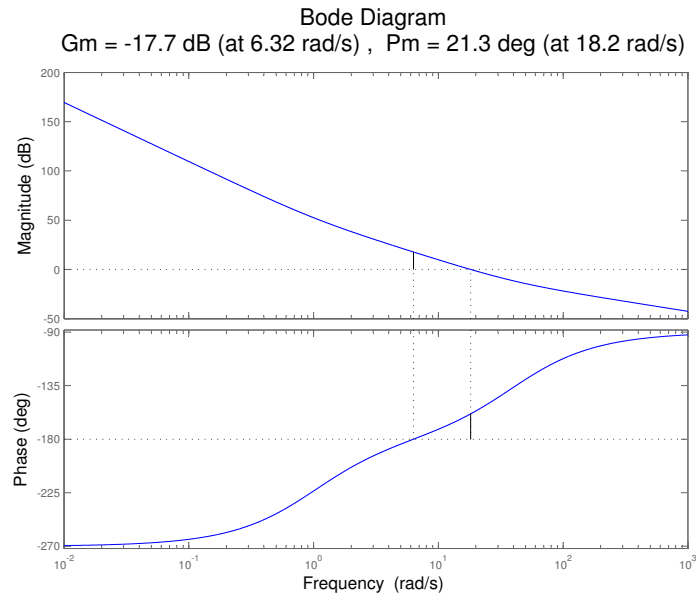


Figure 1.5: Matlab Analysis of phase and gain margin

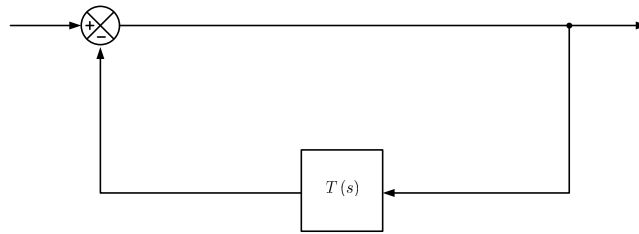


Figure 1.6: Closed-Loop realization of  $T_s$

From Figure 1.6, the closed-loop transfer function is derived as follows:

$$T_{cl}(s) = \frac{1}{1 + T(s)}$$

$$T_{cl}(s) = \frac{1}{\frac{s^3 + A(1 + \frac{s}{\omega_z})(1 + \frac{s}{\omega_1})}{s^3}}$$

$$T_{cl}(s) = \frac{s^3}{s^3 + \frac{A}{\omega_z \omega_1} s^2 + A \left( \frac{1}{\omega_z} + \frac{1}{\omega_1} \right) s + A}$$

With the system defined in a closed-loop form, the denominator polynomial can be used to determine system stability as shown in Figure 1.7

$$\begin{array}{c|cc}
 s^3 & 1 & A\left(\frac{1}{\omega_z} + \frac{1}{\omega_1}\right) \\
 s^2 & \frac{A}{\omega_1\omega_z} & A \\
 s^1 & \frac{A(\omega_1 + \omega_z)}{\omega_z\omega_1} - \omega_z\omega_1 & \\
 s^0 & A & 
 \end{array}$$

Figure 1.7: Routh Hurwitz analysis of closed loop system

$$\begin{array}{c|cc}
 s^3 & 1 & 307.5 \\
 s^2 & 7.5 & 300 \\
 s^1 & 267.5 & \\
 s^0 & 300 & 
 \end{array}$$

Figure 1.8: Routh Hurwitz Values

Applying the system parameters, Figure 1.8 confirms that no sign changes are present in the first column. This confirms that the system is stable, while having negative gain margin.

With the system confirmed as stable, the next item to explore is the possible parameter shifting in the system that could cause the system to become unstable. From the derivations shown in Figure 1.7, the only term that could cause a sign change in the first column is the  $s^1$  term. Setting this term to zero and solving for A:

$$\begin{aligned}
 \frac{A(\omega_1 + \omega_z)}{\omega_1\omega_z} - \omega_z\omega_1 &= 0 \\
 A &= \frac{1}{\omega_z + \omega_1} = 39.024
 \end{aligned}$$

Applying the shifted A parameter to the asymptotic bode plot of Figure 1.4, the updated bode plot is shown in Figure 1.9.

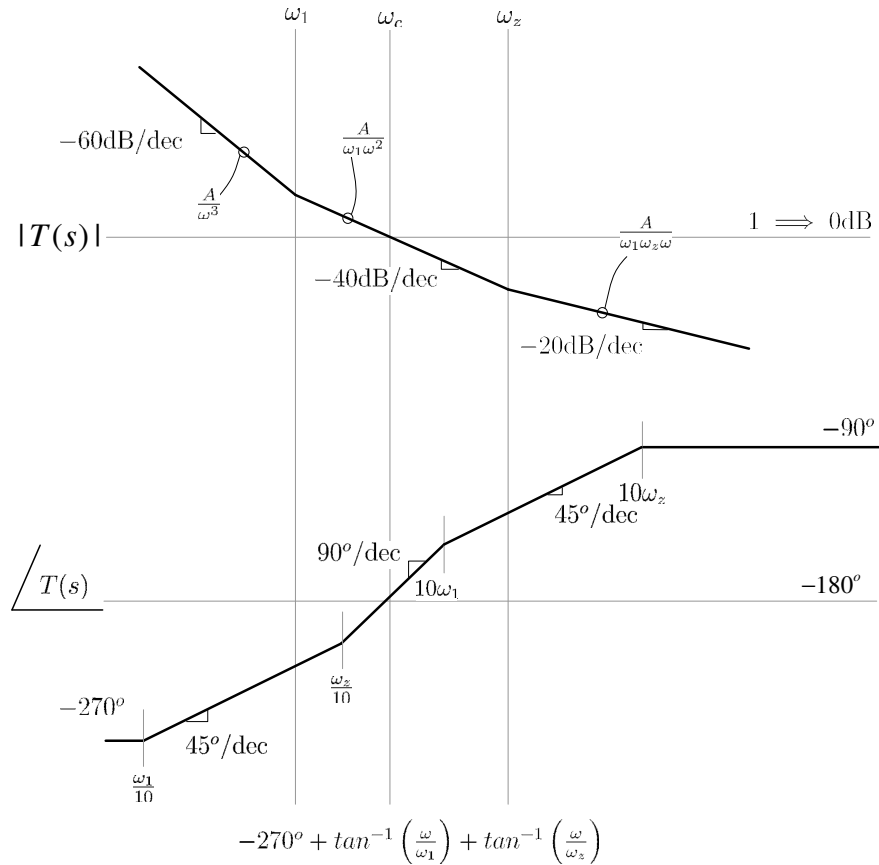


Figure 1.9: Bode Plot: System Loop Gain with parametric shift

Applying the new value of A to the magnitude equation,  $\frac{A}{\omega_c^2} = 1$  or  $\omega_c = \sqrt{A} = 6.25 \text{ rad/s}$ . With this frequency, the phase margin is calculated to be  $-0.2^\circ$ .

As a confirmation of the above margin estimate due to parametric value shift, Figure 1.10 confirms the analysis, with a phase margin of  $-8.82 \cdot 10^{-5}$ .

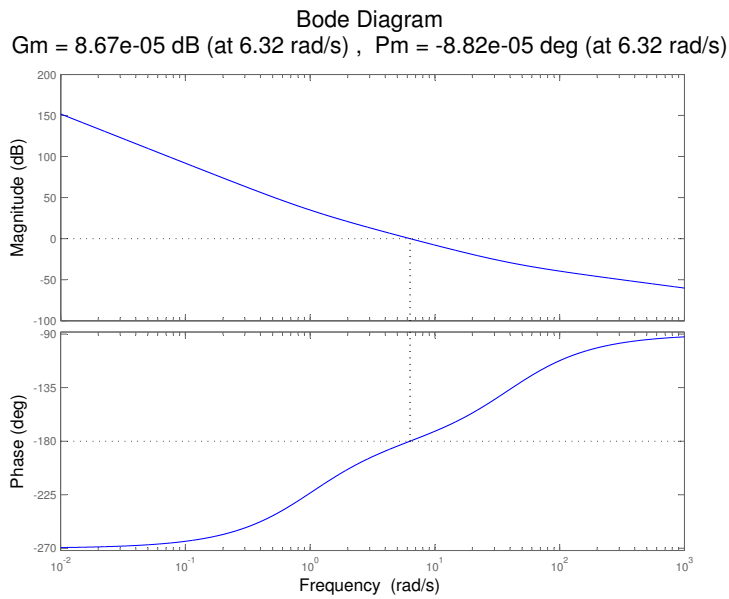


Figure 1.10: Matlab Analysis of phase and gain margin with parametric shift

## Summary

This chapter has shown the reader how to apply asymptotic bode plots to determine the stability of a system. The reader has also learned that in some scenarios it may be necessary to use asymptotic bode plot analysis in conjunction with the Routh-Hurwitz stability criterion to determine the stability of a system.

### 1.3.1 Matlab Code

```
1 clear all;
2 close all;
3
4 f = logspace(-3,3,10000);
5 w = 2*pi*f;
6 s = tf('s');
7
8 A = 300;
9 w1 = 1;
10 wz = 40;
11
12 %=====
13 %System Loop Gain
14 %=====
15 sys = A*(1+s/w1)*(1+s/wz)/(s^3);
16
17 figure(1)
18 [mag, phase] = bode(sys,w);
19 margin(mag, phase, w)
20
21 h = gcr;
22 xlim([10^-2 10^3]);
23 h.AxesGrid.TitleStyle.FontSize = 16;
24 h.AxesGrid.XLabelStyle.FontSize = 12;
25 h.AxesGrid.YLabelStyle.FontSize = 12;
26 %=====
27
28
29 %=====
30 %Plot Marginal Stability Per Routh Hurwitz
31 %=====
32 A=39.024;
33
34 sys = A*(1+s/w1)*(1+s/wz)/(s^3);
35
36 figure(2)
37 [mag, phase] = bode(sys,w);
38 margin(mag, phase, w)
39
40 h = gcr;
41 xlim([10^-2 10^3]);
42 h.AxesGrid.TitleStyle.FontSize = 16;
43 h.AxesGrid.XLabelStyle.FontSize = 12;
44 h.AxesGrid.YLabelStyle.FontSize = 12;
45 %=====
```



## Chapter 2

# System Stability

### 2.1 Stability

Asymptotic stability means that, in the absence of energy coming into the system's input, any stored energy in the system will asymptotically decay to zero over time. A system is considered asymptotically stable if all of the proper transfer function's poles have real components in the open left-half of the complex plane (LHP). This means that the real part of any complex pole  $s = \sigma + j\omega$  must be negative, i.e.,  $\sigma < 0$ , for stability (since the *open LHP* means that the  $j\omega$  axis is excluded). Therefore, a stable system cannot have poles on the  $j\omega$  axis or in the right-half of the complex plane (RHP).

Bounded-input, bounded-output (BIBO) stability means that, in the absence of stored energy in the system, any input signal that is bounded in magnitude, i.e.,  $-\infty < u_{min} \leq u(t) \leq u_{max} < \infty$  will produce an output signal that is bounded in magnitude, i.e.,  $-\infty < y_{min} \leq y(t) \leq y_{max} < \infty$ . BIBO stability exists if the integral of the absolute value of the impulse response is finite, i.e.,  $\int_0^\infty |g(\tau)|d\tau < \infty$ . In order to test BIBO stability, the impulse response function  $g(t)$  must be determined for a given transfer function  $G(s)$  by applying the inverse Laplace transform, an action that is not always trivial.

This text will focus on determining the asymptotic stability of a system, so when a system is declared *stable*, it is referring to asymptotic stability.

### 2.2 Stability Analysis by Inspection

First-order and second-order systems can be analyzed for stability by inspecting their characteristic equations. The following requirements must be met for inspection to be valid:

- The characteristic equation must be second-order or less
- All powers of  $s$  must have non-zero coefficients

- All coefficients of powers of  $s$  must have the same sign

If these three criteria are met, the characteristic equation is stable by inspection. If even one of these criteria is violated, the system is unstable. Third-order and higher-order systems cannot have their stability evaluated by inspection.

### 2.2.1 Parameterized Stability Analysis by Inspection

The stability analysis by inspection technique can be used with parameterized systems to determine the range over which a parameter can vary and still allow the system to remain stable. This can be useful when considering controller gains or uncertainty.

### 2.2.2 Examples of Stability Analysis by Inspection

Given a transfer function characteristic polynomial of:

$$s^2 + 13s + 8$$

determine whether or not system is stable by inspection. Verify the result with the MATLAB *roots* command.

The characteristic equation is second-order. The coefficients of powers of  $s$ , in descending order, are  $\{1, 13, 8\}$ , and are all non-zero. The sign on all of the coefficients of  $s$  is positive. Since all three criteria are met, the transfer function is stable.

Check: since the characteristic polynomial was known, the *roots* command was used in MATLAB to determine the poles were at  $[-12.3523 -0.6477]$ , which not only shows that all poles have real parts in the open left-half of the complex frequency plane, but gives their locations as well.

Given a transfer function characteristic polynomial of:

$$s - 9$$

determine whether or not system is stable by inspection. Verify the result with the MATLAB *roots* command.

The characteristic equation is less than second-order. The coefficients of powers of  $s$ , in descending order, are  $\{1, -9\}$ , and are all non-zero. The sign on the first coefficient is positive, but the second sign is negative indicating the third requirement for stability by inspection is violated, therefore the transfer function is unstable.

Check: since the characteristic polynomial was known, the *roots* command was used in MATLAB to determine the poles were at  $[9]$ , which shows that the pole is real and in the right-half of the complex frequency plane.

Given a transfer function characteristic polynomial of:

$$4s^2 + 3$$

determine whether or not system is stable by inspection. Verify the result with the MATLAB *roots* command.

The characteristic equation is second-order. The coefficients of powers of  $s$ , in descending order, are  $\{4, 0, 3\}$ . Since one of the coefficients is zero, the transfer function is unstable.

Check: since the characteristic polynomial was known, the *roots* command was used in MATLAB to determine the poles were at  $[0 \pm j0.8660]$ , which shows that the poles are on the  $\omega$  axis of the complex frequency plane, producing an oscillator.

A closed-loop transfer function has a characteristic polynomial of:

$$s + 4 - K$$

where  $K$  is the controller gain. Determine, by inspection, the range of non-negative values that  $K$  can assume and still have the closed-loop system remain stable. Check the stability at the upper and lower bounds of the range determined.

The characteristic equation is first-order. The coefficients of powers of  $s$ , in descending order, are  $\{1, 4 - K\}$ . For the transfer function to meet the second and third criteria above,  $4 - K > 0$ . This occurs when  $0 \geq K < 4$  (remember, the question asked for the range of non-negative values  $K$  could assume, thus establishing a lower bound on  $K$  of 0).

Check: when  $K = 0$ , the characteristic equation will be  $s + 4$ , which has a pole at  $-4$ , and is therefore stable. When  $K = 4$ , the characteristic equation will be  $s$ , which has a pole at the origin of the complex plane and is therefore not stable.

## 2.3 Routh-Hurwitz Analysis

Given a transfer function, the Routh-Hurwitz stability analysis technique can be performed using the characteristic equation (i.e., the denominator of the transfer function). It is typically used as a go/no-go test - it can identify the number of poles that exist in the right half of the complex plane or on the imaginary axis, thus indicating stability, but it doesn't directly provide the values of the poles - though they can be determined indirectly. Since even simple numerical analysis software can find the roots of the characteristic polynomial of a system, it is easy to see if any poles are outside of the open left-half of the complex plane. Thus, the Routh-Hurwitz stability analysis technique is no longer necessary for systems where the transfer function characteristic equation or the system matrix are known. However, the technique is useful when a coefficient in the characteristic equation is unknown, or if a coefficient is known only to a degree of uncertainty. For these cases, the stability of the system can be determined for a range of the unknown/uncertain parameter. This section will first show how to develop the Routh-Hurwitz array to demonstrate the principle, then show how it can be used to determine stability margins for uncertain systems.

Given the denominator polynomial:

$$d(s) = a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0$$

the Routh-Hurwitz array of numbers may be established:

$$\begin{array}{c|ccc}
 s^n & a_n & a_{n-2} & a_{n-4} \\
 s^{n-1} & a_{n-1} & a_{n-3} & a_{n-5} \\
 s^{n-2} & b_1 & b_2 & b_3 \\
 s^{n-3} & c_1 & c_2 & c_3 \\
 \vdots & \vdots & \vdots & \vdots \\
 s^0 & h_1 & & 
 \end{array}$$

By examination, the first two rows are created using by alternating the coefficients of the characteristic polynomial, starting with the coefficient of the highest order of  $s$  in the first column of the first row. The entries in the third row and beyond are calculated by using the coefficients of previous rows. For example,  $b_1$  is determined using the following calculation:

$$b_1 = \frac{- \begin{vmatrix} a_n & a_{n-2} \\ a_{n-1} & a_{n-3} \end{vmatrix}}{a_{n-1}} = \frac{a_{n-1}a_{n-2} - a_n a_{n-3}}{a_{n-1}}$$

The entry  $b_2$  is determined by:

$$b_2 = \frac{- \begin{vmatrix} a_n & a_{n-4} \\ a_{n-1} & a_{n-5} \end{vmatrix}}{a_{n-1}} = \frac{a_{n-1}a_{n-4} - a_n a_{n-5}}{a_{n-1}}$$

The entry  $c_1$  is determined by:

$$c_1 = \frac{- \begin{vmatrix} a_{n-1} & a_{n-3} \\ b_1 & b_2 \end{vmatrix}}{b_1} = \frac{b_1 a_{n-3} - b_2 a_{n-1}}{b_1}$$

Therefore, the general rule is:

*“The entry in row  $x + 2$ , column  $y$  is calculated by forming the negative of the determinant formed using the entries from the previous two rows in column 1 and the previous two rows in column  $y + 1$ , then dividing it by the entry from the previous row in column 1.”*

Once the Routh-Hurwitz array has been formed, the number of sign changes in the leftmost column of the array indicates the number of poles in the open right-half of the complex frequency plane, i.e., the number of unstable poles.

Given a transfer function with a characteristic polynomial of  $s^4 + 3s^3 + 4s^2 + 6s + 1$ , form the Routh-Hurwitz array and determine whether or not the system is stable. Verify the result with the MATLAB *roots* command.

The initial part of the array is formed as:

$$\begin{array}{c|ccc}
 s^4 & 1 & 4 & 1 \\
 s^3 & 3 & 6 & 0 \\
 s^2 & & & \\
 s^1 & & & \\
 s^0 & & & 
 \end{array}$$

The first element of the  $s^2$  row is calculated by:

$$b_1 = \frac{- \begin{vmatrix} 1 & 4 \\ 3 & 6 \end{vmatrix}}{3} = \frac{3 \cdot 4 - 1 \cdot 6}{3} = 2$$

The second element of the  $s^2$  row is calculated by:

$$b_2 = \frac{- \begin{vmatrix} 1 & 1 \\ 3 & 0 \end{vmatrix}}{3} = \frac{3 \cdot 1 - 1 \cdot 0}{3} = 1$$

The first element of the  $s^1$  row is calculated by:

$$c_1 = \frac{- \begin{vmatrix} 3 & 6 \\ 2 & 1 \end{vmatrix}}{2} = \frac{2 \cdot 6 - 3 \cdot 1}{2} = 4.5$$

The second element of the  $s^1$  row is calculated by:

$$c_2 = \frac{- \begin{vmatrix} 3 & 0 \\ 2 & 0 \end{vmatrix}}{2} = \frac{2 \cdot 0 - 3 \cdot 0}{2} = 0$$

Further calculations reveal the other  $s^1$  row elements are all zeros.

The first element of the  $s^0$  row is calculated by:

$$d_1 = \frac{- \begin{vmatrix} 2 & 1 \\ 4.5 & 0 \end{vmatrix}}{2} = \frac{4.5 \cdot 1 - 2 \cdot 0}{4.5} = 1$$

Further calculations reveal the other  $s^0$  row elements are all zeros.

Thus, the full array is:

$$\begin{array}{c|ccc} s^4 & 1 & 4 & 1 \\ s^3 & 3 & 6 & 0 \\ s^2 & 2 & 1 & 0 \\ s^1 & 4.5 & 0 & 0 \\ s^0 & 1 & 0 & 0 \end{array}$$

Since no sign changes occur in the leftmost column, the system is stable.

Check: since the characteristic polynomial was known, the *roots* command was used in MATLAB to determine the poles were at [-2.312 -0.251±j1.501, -0.187], which not only shows that all poles have real parts in the open left-half of the complex frequency plane, but gives their locations as well.

Once the Routh-Hurwitz array has been created, there are some special cases that may occur that require handling.

### 2.3.1 Zero in First Column

Because the calculation of elements in the Routh-Hurwitz array requires division by an element in the first column, if that element happens to be zero, the calculation breaks down. For arrays where only the leftmost element in a row is zero, a new row must be formed with a nonzero first element in order to complete the array. Identifying the row with the leading zero as Row A, form Row B by shifting Row A to the left until a zero is no longer the first element, padding on the right with zeros, and multiplying the row formed by -1 for each shift. Then, add Row A and Row B to each other using element-by-element addition to form Row C. Finally, replace Row A in the original array with Row C and continue calculating the remainder of the original array terms.

Given a transfer function with a characteristic polynomial of  $2s^4 + 3s^3 + 4s^2 + 6s + 7$ , form the Routh-Hurwitz array, then determine the number of poles in the open right-half of the complex frequency plane. Verify the result with the MATLAB *roots* command.

The initial part of the array is formed as:

$$\begin{array}{c|ccc} s^4 & 2 & 4 & 7 \\ s^3 & 3 & 6 & 0 \\ s^2 & 0 & 7 & 0 \\ s^1 & & & \\ s^0 & & & \end{array}$$

The  $s^2$  row becomes Row A. Shifting this once to the left, padding with zeros, and multiplying by -1 (due to only one shift to the left) forms Row B = [-7 0]. Adding Rows A and B together forms Row C = [-7 7]. This now goes into the original array in place of Row A and the remaining elements are calculated as indicated below.

$$\begin{array}{c|ccc} s^4 & 2 & 4 & 7 \\ s^3 & 3 & 6 & 0 \\ s^2 & -7 & 7 & 0 \\ s^1 & 9 & 0 & 0 \\ s^0 & 7 & 0 & 0 \end{array}$$

Since there are two sign changes in the first column, there are two poles in the open right-half of the complex frequency plane, and the system is unstable.

Check: since the characteristic polynomial was known, the *roots* command was used in MATLAB to determine the poles were at  $[0.384 \pm j1.337, -1.134 \pm j0.722]$ , which not only shows two poles whose real parts are in the open right-half of the complex frequency plane, but gives all of the pole locations as well.

### 2.3.2 A Row of Zeros

If a row in the Routh-Hurwitz array consists entirely of zeros, there will be poles that are located symmetrically about the origin in the complex frequency plane. If using the Routh-Hurwitz array as an go/no-go test of stability, the row of zeros indicates a condition where poles exist symmetrically along either

the real axis or the imaginary axis. This is enough to indicate that the system is unstable. However, when a parameterized characteristic polynomial occurs, it is of interest to be able to determine all of the elements in the Routh-Hurwitz array. Since division by zero is not allowed, the row of zeros must be replaced with a nonzero row. This is accomplished by creating a divisor polynomial  $p(s)$  from the row above the row containing the zeros and finding its derivative with respect to  $s$ , then replacing the row of zeros with the coefficients in the derivative. The divisor polynomial is created by multiplying the first term in the row by the power of  $s$  associated with the row, then multiplying each additional term by decreasing the power of  $s$  by a factor of 2 each time, and adding all of the terms. For example, assume the  $s^2$  row was [3 3] and the  $s^1$  row was [0 0]. The first term in the divisor polynomial is created by multiplying 3 by  $s^2$  to get  $3s^2$ . The second term in the divisor polynomial is created by multiplying 3 by  $s^0$  to get 3. Therefore, the divisor polynomial is  $p(s) = 3s^2 + 3$ . Taking the derivative with respect to  $s$  gives  $6s$ , therefore the new  $s^1$  row is [6 0].

### 2.3.3 Parameterized Routh-Hurwitz Analysis

As mentioned previously, the Routh-Hurwitz analysis technique can be used with parameterized systems. One of the most popular parameterized systems to analyze involves the DC gain of the controller. The Routh-Hurwitz analysis can be used to determine the range of DC gain for which the system will be stable.

Given a unity gain feedback system with proportional controller  $K$ , and a system transfer function of:

$$\frac{10}{s^3 + 9s^2 + 23s + 15}$$

form the Routh-Hurwitz array, then determine the range of  $K$  for which the system is stable. Verify the result with the MATLAB *roots* command.

The closed-loop transfer function is:

$$T(s) = \frac{10K}{s^3 + 9s^2 + 23s + (15 + 10K)}$$

The Routh-Hurwitz array is formed from the characteristic equation:

$$\begin{array}{c|cc} s^3 & 1 & 23 \\ s^2 & 9 & 15 + 10K \\ s^1 & \frac{9 \cdot 23 - 1 \cdot (15 + 10K)}{9} & \\ s^0 & 15 + 10K & \end{array}$$

Evaluation of the first column terms indicates that  $15 + 10K < 207$  and  $15 + 10K > 0$ . These two inequalities lead to a range for stable operation that is  $-1.5 < K < 19.2$ .

Check: The characteristic polynomial with  $K = -1.5$  is  $s^3 + 9s^2 + 23s$ . The *roots* command was used in MATLAB to determine the poles were at

$[-4.5 \pm j1.6583, 0]$ , which shows a pole at the origin of the complex plane, and therefore the system is unstable. The characteristic polynomial with  $K = 19.2$  is  $s^3 + 9s^2 + 23s + 207$ . The *roots* command was used in MATLAB to determine the poles were at  $[0 \pm j4.7958, -9]$ , which shows a complex pole pair on the  $j\omega$  axis, and therefore the system is unstable. Thus, the range for stable operation is verified to be  $-1.5 < K < 19.2$ .

Another use of the Routh-Hurwitz analysis is evaluating the range of a parameter in the characteristic equation that is uncertain. Given a unity gain feedback system with uncertainty  $\Delta$  and a characteristic equation of:

$$s^3 + 2s^2 + 3(1 + \Delta)s + 6$$

form the Routh-Hurwitz array, then determine the range of  $\Delta$  for which the system is stable. Verify the result with the MATLAB *roots* command.

The Routh-Hurwitz array is formed from the characteristic equation:

$$\begin{array}{c|cc} s^3 & 1 & 2 \\ s^2 & 3(1 + \Delta) & 6 \\ s^1 & 1 + 3\Delta & \\ s^0 & 6 & \end{array}$$

Evaluation of the first column terms indicates that  $1 + 3\Delta > 0$ . This inequality leads to a range for stable operation that is  $\Delta > \frac{-1}{3}$ .

Check: The characteristic polynomial with  $\Delta = \frac{-1}{3}$  is  $s^3 + 2s^2 + 2s + 4$ . The *roots* command was used in MATLAB to determine the poles were at  $[-2, 0 \pm j\sqrt{2}]$ . Since there are a complex pole pair on the  $j\omega$  axis, the system is unstable for this value of  $\Delta$ . Thus, the range for stable operation is verified to be  $\Delta > \frac{-1}{3}$ .



## Chapter 3

# Steady State Error

The output of a system,  $y(t)$  can be decomposed into two components: 1) steady state response,  $y_{ss}(t)$ , and 2) transient response,  $y_{tr}(t)$ , so that

$$y(t) = y_{ss}(t) + y_{tr}(t)$$

For a stable system the transient component will eventually die out, leaving just the steady state response.

In this chapter we will examine the steady state response of feedback systems. The desired steady state response will dictate the type of frequency compensation that can be used for a specific system. Generally, the input(s) used in assessing the steady state response is one or all of the following:

1. Step:  $r(t) = u(t)$ , with  $R(s) = \mathcal{L}\{r(t)\} = \frac{1}{s}$
2. Ramp:  $r(t) = t$ , with  $R(s) = \mathcal{L}\{r(t)\} = \frac{1}{s^2}$
3. Parabola:  $r(t) = \frac{1}{2}t^2$ , with  $R(s) = \mathcal{L}\{r(t)\} = \frac{1}{s^3}$

The motivation for using these inputs may be seen by observing a Taylor series expansion of a general input signal,  $r(t)$  (around point  $t = a$ ):

$$\begin{aligned} r(t) &= r(a) + \left. \frac{dr}{dt} \right|_{t=a} (t-a) + \frac{1}{2!} \left. \frac{d^2r}{dt^2} \right|_{t=a} (t-a)^2 + \dots \\ &= A_0 + A_1 t + A_2 t^2 + \dots \end{aligned}$$

Thus the step, ramp and parabolic inputs can be seen as the first three order terms of a general input signal.

In discussing steady state response we are particularly interested in comparing it with some desired or ideal value. The difference will be defined as the *steady state error*. We will be careful to consider the cases for, with reference to Figure 3.1,

1.  $H(s) = 1$ , i.e. unity gain feedback
2.  $H(s) \neq 1$ , i.e. non-unity gain feedback

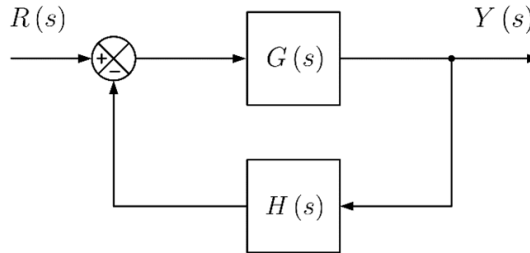


Figure 3.1: General feedback system.

As we have seen the closed loop gain of the system of Figure 3.1 is given by

$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

which can also be written as

$$\frac{Y(s)}{R(s)} = \frac{1}{H(s)} \cdot \frac{G(s)H(s)}{1 + G(s)H(s)}$$

For very large loop gains, i.e.  $|G(s)H(s)| \gg 1$ , the second factor reduces to unity:

$$\frac{G(s)H(s)}{1 + G(s)H(s)} \approx 1, \quad \text{for } |G(s)H(s)| \gg 1$$

so that

$$\frac{Y(s)}{R(s)} \approx \frac{1}{H(s)}, \quad \text{for } |G(s)H(s)| \gg 1$$

Of course, this is precisely the reason that feedback is used in the first place. Namely, for a large loop gain, the overall gain of the system can be controlled by a highly stable gain  $H(s)$ .

We consider the gain,  $\frac{1}{H(s)}$ , as the *ideal* closed loop gain. Furthermore, the term  $\frac{G(s)H(s)}{1 + G(s)H(s)}$ , will be seen as a *correction term*, which deviates from unity at finite values of loop gain.

### 3.1 Unity gain feedback

Consider the unity gain feedback system of Figure 3.2, where  $H(s) = 1$ .

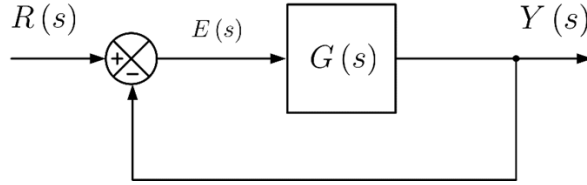


Figure 3.2: Unity gain feedback system.

The ideal output for a unity gain system is the input, i.e.  $Y(s) = R(s)$ , since  $\frac{Y(s)}{R(s)} = \frac{1}{H(s)} = 1$ . Any deviations of the output from the input, due to the correction term, can be seen as the *error*. So that we can define the *error signal* as

$$E(s) = R(s) - Y(s)$$

In terms of the unity gain feedback structure, this is seen as the signal at the input of the transfer function  $G(s)$ .

In the time domain, the error is expressed as:

$$e(t) = r(t) - y(t)$$

and the *steady state error*,  $e_{ss}$ , is defined as

$$\begin{aligned} e_{ss} &\triangleq \lim_{t \rightarrow \infty} e(t) \\ &= \lim_{t \rightarrow \infty} \{r(t) - y(t)\} \end{aligned}$$

To determine  $e_{ss}$ , the full response of a system,  $y(t)$ , may be determined and use of the above expression may be made. To illustrate this, let us consider a feedback system with  $G(s) = \frac{1}{s(s+2)}$  (and  $H(s) = 1$ ). The closed loop system gain is given by:

$$\begin{aligned} \frac{Y(s)}{R(s)} &= \frac{G(s)}{1 + G(s)} \\ &= \frac{\frac{1}{s(s+2)}}{1 + \frac{1}{s(s+2)}} \\ &= \frac{1}{(s+1)^2} \end{aligned}$$

With a unit step input,  $R(s) = \frac{1}{s}$ , the output,  $Y(s)$ , is given by

$$Y(s) = \frac{1}{(s+1)^2} \cdot \frac{1}{s}$$

A partial fraction expansion leads to:

$$Y(s) = \frac{1}{s} - \frac{1}{s+1} - \frac{1}{(s+1)^2}$$

So that the time domain output is given by

$$\begin{aligned} y(t) &= \mathcal{L}^{-1}\{Y(s)\} \\ &= 1 - e^{-t} - te^{-t} \end{aligned}$$

We can identify the two components of the output response:  $y_{ss}(t) = 1$ , and  $y_{tr}(t) = -e^{-t} - te^{-t}$ .

The steady state error is given by

$$\begin{aligned} e_{ss} &= \lim_{t \rightarrow \infty} \{r(t) - y(t)\} \\ &= \lim_{t \rightarrow \infty} \{1 - [1 - e^{-t} - te^{-t}]\} \\ &= 0 \end{aligned}$$

Note however that it is more expeditious to use the Laplace Transform Final Value Theorem to determine the steady state error,  $e_{ss}$ . The final value theorem states that for a signal  $e(t)$ , for which,  $\mathcal{L}\{e(t)\} = E(s)$ , the final value,  $e_{ss}$ , is given by

$$e_{ss} = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} sE(s)$$

Given the error as defined above, in the Laplace domain:

$$\begin{aligned} E(s) &= R(s) - Y(s) \\ &= R(s) - \frac{G(s)}{1+G(s)}R(s) \\ &= \frac{1}{1+G(s)}R(s) \end{aligned}$$

which gives

$$e_{ss} = \lim_{s \rightarrow 0} s \left[ \frac{1}{1+G(s)} \right] R(s)$$

Note that there is an important requirement for the use of the above formula and that is that the transfer function  $\frac{1}{1+G(s)}$  needs to be stable.

Continuing the above example (where  $G(s) = \frac{1}{s(s+2)}$ ) this results in

$$\begin{aligned} e_{ss} &= \lim_{s \rightarrow 0} s \left[ \frac{s(s+2)}{(s+1)^2} \right] \frac{1}{s} \\ &= 0 \end{aligned}$$

This agrees with the previous result obtained using the full time response and taking the limit as time goes to infinity.

Let us now consider the other two inputs:

**Unit Ramp input:**  $r(t) = t$ , with  $\mathcal{L}\{r(t)\} = R(s) = \frac{1}{s^2}$

$$\begin{aligned} e_{ss} &= \lim_{s \rightarrow 0} s \left[ \frac{1}{1 + G(s)} \right] \cdot R(s) \\ &= \lim_{s \rightarrow 0} s \left[ \frac{1}{1 + G(s)} \right] \cdot \frac{1}{s^2} \\ &= \lim_{s \rightarrow 0} s \left[ \frac{s(s+2)}{(s+1)^2} \right] \cdot \frac{1}{s^2} \\ &= 2 \end{aligned}$$

**Unit Parabolic input:**  $r(t) = \frac{1}{2}t^2$ , with  $\mathcal{L}\{r(t)\} = R(s) = \frac{1}{s^3}$

$$\begin{aligned} e_{ss} &= \lim_{s \rightarrow 0} s \left[ \frac{1}{1 + G(s)} \right] \cdot R(s) \\ &= \lim_{s \rightarrow 0} s \left[ \frac{1}{1 + G(s)} \right] \cdot \frac{1}{s^3} \\ &= \lim_{s \rightarrow 0} s \left[ \frac{s(s+2)}{(s+1)^2} \right] \cdot \frac{1}{s^3} \\ &= \infty \end{aligned}$$

To summarize, we see that the steady state error for the three inputs of unit step, unit ramp and unit parabola exhibit values of 0, 2 (a finite value) and infinity, respectively. In general, it can be stated that there are two main factors which impinge on the value of steady state error.

1. type of input (i.e. step, ramp, parabola etc.), and,
2. a property of the system, which is termed the *system type*. This is defined next.

### System Type:

The general form of the transfer function of a system is given as follows:

$$G(s) = \frac{n(s)}{d(s)}$$

where  $n(s)$  and  $d(s)$  denote the numerator and denominator polynomials, respectively. Let us now consider that a  $N$ -th order factor of  $s$  appears in  $d(s)$ . That is to say, there are  $N$  zeros at  $s = 0$  appearing in  $d(s)$ , so that  $G(s)$  can be expressed as

$$G(s) = \frac{n(s)}{s^N \cdot \bar{d}(s)}$$

where  $d(s) = s^N \cdot \bar{d}(s)$ . If this factorization is possible then we say that the system type number is  $N$ . In other words, the system type number is given by the number of poles at  $s = 0$  appearing in the transfer function  $G(s)$ . If there are none then the system type number is zero.

In determining the steady state error we used

$$e_{ss} = \lim_{s \rightarrow 0} s \left[ \frac{1}{1 + G(s)} \right] \cdot R(s)$$

Substituting for  $G(s)$  from above we have

$$e_{ss} = \lim_{s \rightarrow 0} s \left[ \frac{s^N \cdot \bar{d}(s)}{s^N \cdot \bar{d}(s) + n(s)} \right] \cdot R(s)$$

Given inputs that are powers of  $\frac{1}{s}$ , that is inputs of the form

$$R(s) = \frac{1}{s^i}$$

where constant,  $i = 1, 2, 3 \dots$ , then the steady state error is given by

$$e_{ss} = \lim_{s \rightarrow 0} \left[ \frac{\bar{d}(s)}{s^N \cdot \bar{d}(s) + n(s)} \right] \cdot s^{N-i+1}$$

Consequently, we have the three cases:

1. When,  $N - i + 1 > 0$  or  $i < N + 1$ , then  $e_{ss} = 0$
2. When,  $N - i + 1 = 0$  or  $i = N + 1$ , then  $e_{ss} = \lim_{s \rightarrow 0} \left[ \frac{\bar{d}(s)}{s^N \cdot \bar{d}(s) + n(s)} \right]$
3. When,  $N - i + 1 < 0$  or  $i > N + 1$ , then  $e_{ss} = \infty$

Relating this back to our previous example with transfer function given by

$$G(s) = \frac{1}{s(s+2)}$$

We see that the system type number is 1, i.e.  $N = 1$ , as there is one pole at  $s = 0$ . For a step input,  $i = 1$ , which corresponds to case (1) above leading to  $e_{ss} = 0$ . For a ramp input,  $i = 2$ , this is case (2) (since  $N = 1$ ) so that the steady state error is given by

$$\begin{aligned}
e_{ss} &= \lim_{s \rightarrow 0} \left[ \frac{\bar{d}(s)}{s^1 \cdot \bar{d}(s) + n(s)} \right] \\
&= \lim_{s \rightarrow 0} \left[ \frac{s + 2}{s^1 \cdot (s + 2) + 1} \right] \\
&= 2
\end{aligned}$$

For parabolic input,  $i = 3$ , we have case (3) (since  $N = 1$ ) resulting in a steady state error  $e_{ss} = \infty$ . These results agree with those previously obtained.

### Error Constants:

It is customary to define steady state error in terms of system error constants. These are the position error constant,  $K_p$ , the velocity error constant,  $K_v$ , and the acceleration error constant,  $K_a$ . They are defined for a system  $G(s)$  as follows:

**Position error constant:**

$$K_p \triangleq \lim_{s \rightarrow 0} G(s)$$

**Velocity error constant:**

$$K_v \triangleq \lim_{s \rightarrow 0} sG(s)$$

**Acceleration error constant:**

$$K_a \triangleq \lim_{s \rightarrow 0} s^2 G(s)$$

The steady state error may be defined in terms of these constants. As we have seen, in general

$$e_{ss} = \lim_{s \rightarrow 0} s \left[ \frac{1}{1 + G(s)} \right] \cdot R(s)$$

For the different inputs we have:

**Step input:**

$$\begin{aligned}
e_{ss} &= \lim_{s \rightarrow 0} s \left[ \frac{1}{1 + G(s)} \right] \cdot \frac{1}{s} \\
&= \frac{1}{1 + \lim_{s \rightarrow 0} G(s)} \\
&= \frac{1}{1 + K_p}
\end{aligned}$$

**Ramp input:**

$$\begin{aligned} e_{ss} &= \lim_{s \rightarrow 0} s \left[ \frac{1}{1 + G(s)} \right] \cdot \frac{1}{s^2} \\ &= \frac{1}{\lim_{s \rightarrow 0} sG(s)} \\ &= \frac{1}{K_v} \end{aligned}$$

**Parabolic input:**

$$\begin{aligned} e_{ss} &= \lim_{s \rightarrow 0} s \left[ \frac{1}{1 + G(s)} \right] \cdot \frac{1}{s^3} \\ &= \frac{1}{\lim_{s \rightarrow 0} s^2 G(s)} \\ &= \frac{1}{K_a} \end{aligned}$$

These results are summarized in Table 3.1 where the inputs have been scaled by a constant  $A$ .

Table 3.1: Steady State Error,  $e_{ss}$ , for systems of different type numbers,  $N$ , and for different inputs.

System Type Number, $N$	Step input: $r(t) = Au(t)$	Ramp input: $r(t) = At$	Parabolic input: $r(t) = \frac{A}{2}t^2$
0	$e_{ss} = \frac{A}{1 + K_p}$	$e_{ss} = \infty$	$e_{ss} = \infty$
1	$e_{ss} = 0$	$e_{ss} = \frac{A}{K_v}$	$e_{ss} = \infty$
2	$e_{ss} = 0$	$e_{ss} = 0$	$e_{ss} = \frac{A}{K_a}$
$\geq 3$	$e_{ss} = 0$	$e_{ss} = 0$	$e_{ss} = 0$

We can now summarize the procedure to determine the steady state error of unity gain feedback systems.

**Procedure to determine steady state error for unity gain feedback systems:**

Given  $G(s)$  (and  $H(s) = 1$ ) and the type of input:



1. Determine the system type. To do this determine the number of poles at zero (i.e. at  $s = 0$ ) of transfer function  $G(s)$ . Alternatively determine the number of zeros appearing at  $s = 0$  of the transfer function:  $1 - M(s)$ , where  $M(s) = \frac{G(s)}{1+G(s)}$ , is the closed loop gain.
2. With the system and input types, the steady state error,  $e_{ss}$ , can be read from Table 3.1 for most combinations or determined using the appropriate error constant.

### 3.1.1 Example 1:

Consider a unity gain feedback system with  $G(s) = \frac{40}{s+4}$  (and  $H(s) = 1$ ).

- a) Determine the steady state error for a unit step, unit ramp and unit parabolic inputs using the results of Table 3.1.
- b) Confirm your results of (a) by deriving the time domain response and taking the limit in time.

#### Solution:

- a) System type is  $N = 0$ , as there are no poles at  $s = 0$ . We can see the steady state errors for the three inputs from the first row of Table 3.1. For both the ramp and parabolic inputs, we have  $e_{ss} = \infty$ . For the step input, the steady state error is finite and given by  $e_{ss} = \frac{1}{1+K_p}$ , where the error constant,  $K_p$ , needs to be determined for this system.

It is given by

$$\begin{aligned} K_p &\triangleq \lim_{s \rightarrow 0} G(s) \\ &= \lim_{s \rightarrow 0} \frac{40}{s+4} \\ &= 10 \end{aligned}$$

Accordingly,

$$\begin{aligned} e_{ss} &= \frac{1}{1+K_p} \\ &= \frac{1}{1+10} \\ &= 0.0909 \end{aligned}$$

- b) We'll now confirm the above results by deriving the time domain responses for the three input types.

The closed loop system gain is given by

$$\begin{aligned}\frac{Y(s)}{R(s)} &= \frac{G(s)}{1+G(s)} \\ &= \frac{\frac{40}{s+4}}{1+\frac{40}{s+4}}\end{aligned}$$

So that

$$\frac{Y(s)}{R(s)} = \frac{40}{s+44}$$

**Step input:**

With a step input,  $r(t) = u(t)$ , so that,  $R(s) = \frac{1}{s}$ , the output,  $Y(s)$ , is given by

$$\begin{aligned}Y(s) &= \frac{40}{s+44} \cdot \frac{1}{s} \\ &= \frac{0.909}{s} - \frac{0.909}{s+44}\end{aligned}$$

So that

$$\begin{aligned}y(t) &= \mathcal{L}^{-1}\{Y(s)\} \\ &= 0.909 - 0.909e^{-44t}\end{aligned}$$

The steady state error is given by

$$\begin{aligned}e_{ss} &= \lim_{t \rightarrow \infty} \{r(t) - y(t)\} \\ &= \lim_{t \rightarrow \infty} \{1 - [0.909 + 0.909e^{-44t}]\} \\ &= 1 - 0.909 \\ &= 0.0909\end{aligned}$$

**Ramp input:**

For a ramp input,  $r(t) = t$ , so that,  $R(s) = \frac{1}{s^2}$ , the output,  $Y(s)$ , is given by

$$Y(s) = \frac{40}{s+44} \cdot \frac{1}{s^2}$$

A partial fraction expansion leads to

$$Y(s) = -\frac{5}{242} \cdot \frac{1}{s} + \frac{10}{11} \cdot \frac{1}{s^2} + \frac{5}{242} \cdot \frac{1}{s+44}$$

The steady state error is given by

$$\begin{aligned}
e_{ss} &= \lim_{t \rightarrow \infty} \{r(t) - y(t)\} \\
&= \lim_{t \rightarrow \infty} \left\{ t - \left[ -\frac{5}{242} + \frac{10}{11}t + \frac{5}{242}e^{-44t} \right] \right\} \\
&= \infty
\end{aligned}$$

### Parabolic input:

For a parabolic input,  $r(t) = \frac{1}{2}t^2$ , so that,  $R(s) = \frac{1}{s^3}$ , the output,  $Y(s)$ , is given by

$$Y(s) = \frac{40}{s+44} \cdot \frac{1}{s^3}$$

By a partial fraction expansion we have

$$Y(s) = \frac{5}{10648} \cdot \frac{1}{s} - \frac{5}{242} \cdot \frac{1}{s^2} + \frac{10}{11} \cdot \frac{1}{s^3} - \frac{5}{10648} \cdot \frac{1}{s+44}$$

The steady state error is given by

$$\begin{aligned}
e_{ss} &= \lim_{t \rightarrow \infty} \{r(t) - y(t)\} \\
&= \lim_{t \rightarrow \infty} \left\{ \frac{1}{2}t^2 - \left[ \frac{5}{10648} - \frac{5}{242}t + \frac{10}{11} \cdot \frac{1}{2}t^2 - \frac{5}{10648}e^{-44t} \right] \right\} \\
&= \infty
\end{aligned}$$

## 3.2 Non-Unity Feedback

We now consider the case of a non-unity feedback system, i.e.  $H(s) \neq 1$ . As discussed earlier, the closed loop gain is given by

$$\frac{Y(s)}{R(s)} = \frac{1}{H(s)} \cdot \frac{G(s)H(s)}{1+G(s)H(s)}$$

Under the condition of infinite loop gain the second factor becomes unity and the output is a scaled version of the input

$$Y(s) = \frac{1}{H(s)}R(s)$$

This is considered the ideal output. The factor  $\frac{G(s)H(s)}{1+G(s)H(s)}$  is considered a correction term which causes the actual output to deviate from the ideal when the loop gain, i.e.  $G(s)H(s)$ , is finite.

However, as we are interested in the steady state response, the dc gain of  $H(s)$  is used to define the scaling constant. To this end we define the term  $k_H$  (with the assumption that  $H(s)$  has no poles at zero):

$$k_H \triangleq \lim_{s \rightarrow 0} H(s) = H(0)$$

In steady state we will consider the ideal steady gain as,  $\frac{1}{k_H}$ , consequently the steady state error is defined by the steady state value of the difference between the ideal steady state output,  $\frac{1}{k_H}r(t)$ , and the actual output,  $y(t)$ . That is,

$$e_{ss} \triangleq \lim_{t \rightarrow \infty} \left\{ \frac{1}{k_H} r(t) - y(t) \right\}$$

Using the Laplace final value theorem this is given by

$$e_{ss} = \lim_{s \rightarrow 0} s \left\{ \frac{1}{k_H} R(s) - Y(s) \right\}$$

Substituting the closed loop transfer function,  $M(s)$

$$\begin{aligned} e_{ss} &= \lim_{s \rightarrow 0} s \left\{ \frac{1}{k_H} R(s) - M(s)R(s) \right\} \\ &= \frac{1}{k_H} \lim_{s \rightarrow 0} s \{1 - k_H M(s)\} R(s) \end{aligned}$$

where  $M(s)$  is given by

$$M(s) = \frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

We now define the system type number for the non-unity gain feedback system as the number of zeros of the transfer function  $1 - k_H M(s)$  appearing at  $s = 0$ . This, together with the type of input,  $R(s)$ , are the major determinants of the steady state error. Representing the transfer function,  $1 - k_H M(s)$ , in the following form:

$$1 - k_H M(s) = \frac{s^N \cdot \bar{n}(s)}{d(s)}$$

where  $N$  represents the system type and the numerator,  $n(s) = s^N \cdot \bar{n}(s)$

$$e_{ss} = \frac{1}{k_H} \lim_{s \rightarrow 0} s \left[ \frac{s^N \cdot \bar{n}(s)}{d(s)} \right] \cdot R(s)$$

For inputs of the form  $R(s) = \frac{A}{s^i}$ , (such as for a step ( $i = 1$ ), ramp ( $i = 2$ ), and parabolic inputs ( $i = 2$ )), the steady state error is given by

$$e_{ss} = \frac{A}{k_H} \lim_{s \rightarrow 0} \left[ \frac{\bar{n}(s)}{d(s)} \right] \cdot s^{N-i+1}$$

As before for the unity feedback system, we consider the following three cases:

1. When,  $N - i + 1 > 0$  or  $i < N + 1$ , then  $e_{ss} = 0$

2. When,  $N - i + 1 = 0$  or  $i = N + 1$ , then  $e_{ss} = \frac{A}{k_H} \lim_{s \rightarrow 0} \left[ \frac{\bar{n}(s)}{d(s)} \right]$

3. When,  $N - i + 1 < 0$  or  $i > N + 1$ , then  $e_{ss} = \infty$

Looking more closely at case (2), and represent  $M(s)$  by a rational polynomial:

$$M(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}$$

We are now in a position to determine the steady state errors for the various inputs. Substituting for  $M(s)$  in the following expression for  $e_{ss}$

$$e_{ss} = \frac{1}{k_H} \lim_{s \rightarrow 0} s \cdot [1 - k_H M(s)] \cdot R(s)$$

for each of the three inputs leads to the following results:

**Step input:**

For a step input,  $i = 1$ , case (2) corresponds to a type 0 system, since  $N = i - 1 = 0$ . Substituting  $R(s) = \frac{A}{s}$  leads to

$$\begin{aligned} e_{ss} &= \frac{1}{k_H} \cdot \lim_{s \rightarrow 0} s \cdot \left[ \frac{s^n + \dots + (a_1 - b_1 k_H) s + (a_0 - b_0 k_H)}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} \right] \cdot \frac{A}{s} \\ &= \frac{A}{k_H} \left[ \frac{(a_0 - b_0 k_H)}{a_0} \right] \end{aligned}$$

**Ramp input:**

For a ramp input,  $i = 2$ , case (2) corresponds to a type 1 system, since  $N = i - 1 = 1$ . Substituting  $R(s) = \frac{A}{s^2}$  leads to

$$\begin{aligned} e_{ss} &= \frac{1}{k_H} \cdot \lim_{s \rightarrow 0} s \cdot \left\{ \frac{s [s^{n-1} + \dots + (a_2 - b_2 k_H) s + (a_1 - b_1 k_H)]}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} \right\} \cdot \frac{A}{s^2} \\ &= \frac{A}{k_H} \left[ \frac{(a_1 - b_1 k_H)}{a_0} \right] \end{aligned}$$

**Parabolic input:**

For a parabolic input,  $i = 3$ , case (2) corresponds to a type 2 system, since  $N = i - 1 = 2$ . Substituting  $R(s) = \frac{A}{s^3}$  leads to

$$\begin{aligned} e_{ss} &= \frac{1}{k_H} \cdot \lim_{s \rightarrow 0} s \cdot \left\{ \frac{s^2 [s^{n-2} + \dots + (a_3 - b_3 k_H) s + (a_2 - b_2 k_H)]}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} \right\} \cdot \frac{A}{s^3} \\ &= \frac{A}{k_H} \left[ \frac{(a_2 - b_2 k_H)}{a_0} \right] \end{aligned}$$

The above results are summarized in Table 3.2. Note also that the unity feedback case where  $k_H = 1$  is also covered in these results and provide an alternative set of formulas from those shown in Table 3.2 which does not involve determination of error constants.

Table 3.2: Steady State Error,  $e_{ss}$ , for systems of different type numbers,  $N$ , and for different inputs for non-unity gain systems.  $k_H \triangleq \lim_{s \rightarrow 0} H(s) = H(0)$

System Type: $N$	Step input: $r(t) = Au(t)$	Ramp input: $r(t) = At$	Parabolic input: $r(t) = \frac{A}{2}t^2$
<b>0</b>	$e_{ss} = \frac{A}{k_H} \left[ \frac{(a_0 - b_0 k_H)}{a_0} \right]$	$e_{ss} = \infty$	$e_{ss} = \infty$
<b>1</b>	$e_{ss} = 0$	$e_{ss} = \frac{A}{k_H} \left[ \frac{(a_1 - b_1 k_H)}{a_0} \right]$	$e_{ss} = \infty$
<b>2</b>	$e_{ss} = 0$	$e_{ss} = 0$	$e_{ss} = \frac{A}{k_H} \left[ \frac{(a_2 - b_2 k_H)}{a_0} \right]$
$\geq 3$	$e_{ss} = 0$	$e_{ss} = 0$	$e_{ss} = 0$

We can now summarize the procedure to determine the steady state error of non-unity gain, as well as unity gain, feedback systems. A unity gain feedback systems error analysis using this approach is demonstrated in one of the examples below.

**Procedure to determine steady state error:**

Given  $G(s)$  and  $H(s)$  and the type of input:

1. Determine constant  $k_H$ , the DC gain of  $H(s)$ , i.e.  $k_H \triangleq \lim_{s \rightarrow 0} H(s) = H(0)$ .
2. Determine the system type. To do this determine the number of zeros appearing at  $s = 0$  of the transfer function:  $1 - k_H M(s)$ , where  $M(s) = \frac{G(s)}{1 + G(s)H(s)}$ , is the closed loop gain.
3. With the system type number and input type, the steady state error,  $e_{ss}$ , can be read from Table 3.2 for most combinations or determined using the appropriate coefficients of certain numerator and denominator terms of  $M(s)$ .

**3.2.1 Example 2:**

Given a feedback system with,  $G(s) = \frac{40}{s+4}$  and  $H(s) = 0.5$ .

- a) Find the steady state error,  $e_{ss}$ , for unit step, ramp and parabolic inputs.
- b) Confirm your results by deriving the full output time response and taking the limit as time goes to infinity. Recall that steady state error is defined as the difference between the ideal steady state output, i.e.  $\frac{1}{k_H}r(t)$ , and the actual output,  $y(t)$ , so that

$$e_{ss} = \lim_{t \rightarrow \infty} \left\{ \frac{1}{k_H} r(t) - y(t) \right\}$$

**Solution:**

a) Steps:

- 1) As  $H(s) = 0.5$ , which is frequency independent, the constant  $k_H = 0.5$ .
- 2) The closed loop gain  $M(s) = \frac{G(s)}{1+G(s)H(s)}$  is found to be

$$\begin{aligned} M(s) &= \frac{\frac{40}{s+4}}{1 + 0.5 \cdot \frac{40}{s+4}} \\ &= \frac{40}{s+24} \end{aligned}$$

Furthermore

$$\begin{aligned} 1 - k_H M(s) &= 1 - 0.5 \cdot \frac{40}{s+24} \\ &= \frac{s+4}{s+24} \end{aligned}$$

From which we see that there are no zeros at  $s = 0$ , so that the system type = 0.

- 3) With a system type of 0, from the first row of Table 3.2 we see that  $e_{ss} = \infty$  for both ramp and parabolic inputs and for a unit step (with  $A = 1$ ), the error is given by

$$e_{ss} = \frac{1}{k_H} \left[ \frac{(a_0 - b_0 k_H)}{a_0} \right]$$

Noting coefficients of the numerator and denominator of  $M(s)$  we have  $a_0 = 24$ ,  $b_0 = 40$  and with  $k_H = 0.5$ , we find  $e_{ss} = 0.33$ .

- b) We will confirm this last result first by deriving the full output time response:

$$\frac{Y(s)}{R(s)} = M(s) = \frac{40}{s+24}$$

We have the output

$$y(t) = \mathcal{L}^{-1} \{M(s) R(s)\}$$

For a unit step input,  $R(s) = \frac{1}{s}$ ,

$$y(t) = \mathcal{L}^{-1} \left\{ \frac{40}{s+24} \cdot \frac{1}{s} \right\}$$

Obtaining a partial fraction expansion

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \left\{ \frac{1.667}{s} - \frac{1.667}{s+24} \right\} \\ &= 1.667 - 1.667e^{-24t} \end{aligned}$$

This has the form

$$y(t) = y_{ss}(t) + y_{tr}(t)$$

where  $y_{ss}(t) = 1.667$  and  $y_{tr}(t) = -1.667e^{-24t}$ . The transient term  $y_{tr}(t)$  will decay to zero as time progresses. The steady state error is defined by

$$\begin{aligned} e_{ss} &= \lim_{t \rightarrow \infty} \left\{ \frac{1}{k_H} r(t) - y(t) \right\} \\ &= \lim_{t \rightarrow \infty} \left\{ \frac{1}{k_H} r(t) - [y_{ss}(t) + y_{tr}(t)] \right\} \end{aligned}$$

The transient response will die out to zero so ignoring the transient component in the output, we have:

$$\begin{aligned} e_{ss} &= \lim_{t \rightarrow \infty} \left\{ \frac{1}{k_H} r(t) - y_{ss}(t) \right\} \\ &= \frac{1}{0.5} - 1.667 \\ &= 0.33 \end{aligned}$$

This confirms our previous result.

A plot of the steady state ideal, i.e.  $\frac{1}{k_H} r(t)$ , and actual outputs are shown in Figure 3.3. The time evolution of the difference, representing the error is shown in Figure 3.4.



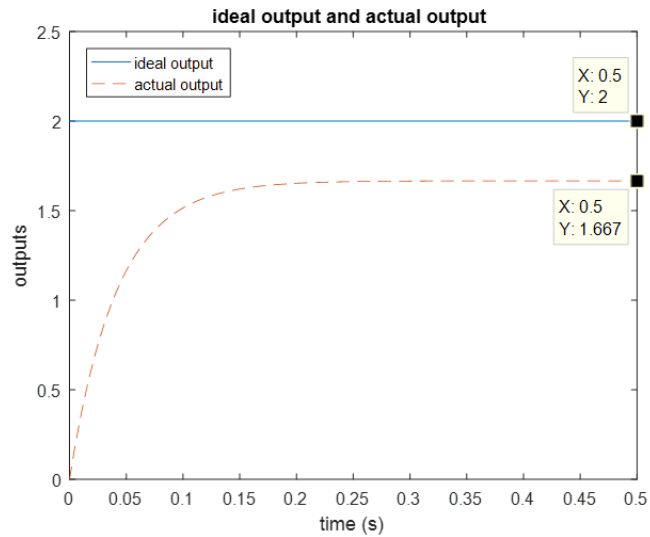


Figure 3.3: The ideal steady state output and actual output. In steady state the difference is seen to be  $2 - 1.667 = 0.33$ .

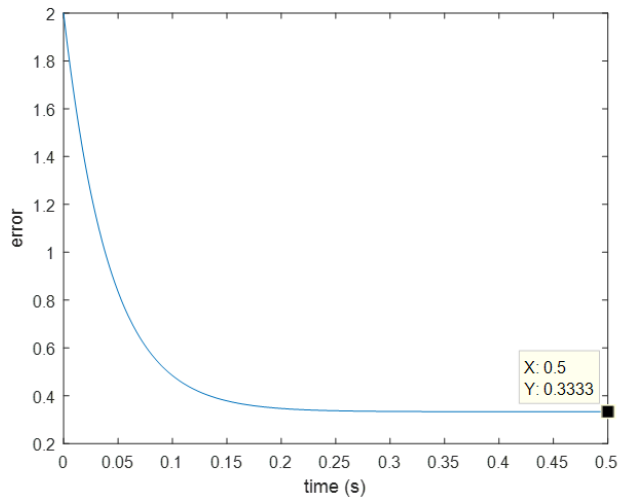


Figure 3.4: The time evolution of the error. The error signal settles to 0.33 in the steady state.

**Ramp input:**

For a unit ramp input,  $R(s) = \frac{1}{s^2}$ , the output is given by

$$y(t) = \mathcal{L}^{-1} \left\{ \frac{40}{s+24} \cdot \frac{1}{s^2} \right\}$$

Obtaining a partial fraction expansion

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \left\{ -\frac{0.069}{s} + \frac{1.667}{s^2} + \frac{0.069}{s+24} \right\} \\ &= -0.069 + 1.667t - 0.069e^{-24t} \end{aligned}$$

This has the form

$$y(t) = y_{ss}(t) + y_{tr}(t)$$

where  $y_{ss}(t) = -0.069 + 1.667t$  and  $y_{tr}(t) = -0.069e^{-24t}$ . The transient term  $y_{tr}(t)$  will decay to zero as time progresses, so ignoring the transient component in the output, we have:

$$e_{ss} = \lim_{t \rightarrow \infty} \left\{ \frac{1}{k_H} r(t) - y_{ss}(t) \right\}$$

For a unit ramp input,  $r(t) = t$ , this leads to

$$\begin{aligned} e_{ss} &= \lim_{t \rightarrow \infty} \left\{ \frac{1}{0.5} t - [-0.069 + 1.667t] \right\} \\ &= \lim_{t \rightarrow \infty} \{0.069 + 0.333t\} \\ &= \infty \end{aligned}$$

### Parabolic input:

For unit parabolic input,  $R(s) = \frac{1}{s^3}$ ,

$$y(t) = \mathcal{L}^{-1} \left\{ \frac{40}{s+24} \cdot \frac{1}{s^3} \right\}$$

Obtaining a partial fraction expansion

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \left\{ \frac{0.0029}{s} - \frac{0.069}{s^2} + \frac{1.667}{s^3} - \frac{0.0029}{s+24} \right\} \\ &= 0.0029 - 0.069t + 1.667 \frac{t^2}{2} - 0.0029e^{-24t} \end{aligned}$$

This has the form

$$y(t) = y_{ss}(t) + y_{tr}(t)$$

where  $y_{ss}(t) = 0.0029 - 0.069t + 1.667 \frac{t^2}{2}$  and  $y_{tr}(t) = -0.0029e^{-24t}$ . The transient term  $y_{tr}(t)$  will decay to zero as time progresses. The steady state error is given by:

$$e_{ss} = \lim_{t \rightarrow \infty} \left\{ \frac{1}{k_H} r(t) - y_{ss}(t) \right\}$$

For a unit parabolic input,  $r(t) = \frac{t^2}{2}$ , this leads to

$$\begin{aligned} e_{ss} &= \lim_{t \rightarrow \infty} \left\{ \frac{1}{0.5} \cdot \frac{t^2}{2} - [0.0029 - 0.069t + 1.667 \frac{t^2}{2}] \right\} \\ &= \lim_{t \rightarrow \infty} \left\{ -0.0029 + 0.069t + 0.333 \frac{t^2}{2} \right\} \\ &= \infty \end{aligned}$$

### 3.2.2 Example 3:

Given a feedback system with,  $G(s) = \frac{40}{s+4}$  and  $H(s) = 1$ .

- Find the steady state error,  $e_{ss}$ , for unit step, ramp and parabolic inputs.
- Confirm your result for the input that produces a finite (non-zero) steady state error by deriving the full output time response and taking the limit as time goes to infinity.

**Solution:**

a) Steps:

- As  $H(s) = 1$ , the constant  $k_H = 1$ .
- The closed loop gain  $M(s) = \frac{G(s)}{1+G(s)H(s)} = \frac{G(s)}{1+G(s)}$  so that

$$\begin{aligned} M(s) &= \frac{\frac{40}{s+4}}{1 + \frac{40}{s+4}} \\ &= \frac{40}{s+44} \end{aligned}$$

and

$$\begin{aligned} 1 - k_H M(s) &= 1 - 1 \cdot \frac{40}{s+44} \\ &= \frac{s+4}{s+44} \end{aligned}$$

From which we can see that the system type = 0, as there are no zeros at  $s = 0$ .

- 3) With a system type of 0, from the first row of Table 3.2 we see that  $e_{ss} = \infty$  for both ramp and parabolic inputs and for a unit step (with  $A = 1$ ), the error is given by

$$e_{ss} = \frac{1}{k_H} \left[ \frac{(a_0 - b_0 k_H)}{a_0} \right]$$

Noting coefficients of the numerator and denominator of  $M(s)$  we have  $a_0 = 44$ ,  $b_0 = 40$  and with  $k_H = 1$ , we find  $e_{ss} = 0.091$ .

- b) We will confirm this last result by obtaining the full time domain response.

$$M(s) = \frac{Y(s)}{R(s)} = \frac{40}{s + 44}$$

For a unit step input,  $R(s) = \frac{1}{s}$ , and the output is

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \{M(s) \cdot R(s)\} \\ &= \mathcal{L}^{-1} \left\{ \frac{40}{s + 44} \cdot \frac{1}{s} \right\} \end{aligned}$$

Obtaining a partial fraction expansion

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \left\{ \frac{0.909}{s} - \frac{0.909}{s + 44} \right\} \\ &= 0.91 - 0.91e^{-44t} \end{aligned}$$

This has the form

$$y(t) = y_{ss}(t) + y_{tr}(t)$$

where  $y_{ss}(t) = 0.91$  and  $y_{tr}(t) = -0.91e^{-44t}$ . The steady state error is given by:

$$e_{ss} = \lim_{t \rightarrow \infty} \left\{ \frac{1}{k_H} r(t) - y_{ss}(t) \right\}$$

For a unit step,  $r(t) = 1, t > 0$ , this leads to

$$\begin{aligned} e_{ss} &= \lim_{t \rightarrow \infty} \{1 - 0.909\} \\ &= 0.091 \end{aligned}$$

This confirms our previous result.

### 3.2.3 Example 4:

Given a feedback system with,  $G(s) = \frac{1}{s}$  and  $H(s) = \frac{2}{s+4}$ .

- Find the steady state error,  $e_{ss}$ , for unit step, ramp and parabolic inputs.
- Confirm your results for step and ramp inputs by deriving the full output time response and taking the limit as time goes to infinity.

**Solution:**

a) Steps:

1) As  $H(s) = \frac{2}{s+4}$ , the constant  $k_H \triangleq \lim_{s \rightarrow 0} H(s) = H(0) = \frac{2}{4} = 0.5$ .

2) The closed loop gain  $M(s) = \frac{G(s)}{1+G(s)H(s)}$  so that

$$\begin{aligned} M(s) &= \frac{\frac{1}{s}}{1 + \frac{1}{s} \cdot \frac{2}{s+4}} \\ &= \frac{s+4}{s^2+4s+2} \end{aligned}$$

and

$$\begin{aligned} 1 - k_H M(s) &= 1 - 0.5 \cdot \frac{s+4}{s^2+4s+2} \\ &= \frac{s(s+3.5)}{s^2+4s+2} \end{aligned}$$

From which we can see that the system type = 1, as there is one zero at  $s = 0$ .

- 3) With a system type of 1, from the second row of Table 3.2 we see that  $e_{ss} = 0$  and  $e_{ss} = \infty$  for step and parabolic inputs, respectively. For a unit ramp (with  $A = 1$ ), the error is given by

$$e_{ss} = \frac{1}{k_H} \left[ \frac{(a_1 - b_1 k_H)}{a_0} \right]$$

Noting the numerator and denominator coefficients of  $M(s)$  we see that  $a_0 = 2$ ,  $a_1 = 4$ ,  $b_1 = 1$  and with  $k_H = 0.5$ , we find that  $e_{ss} = 3.5$ .

- b) We will confirm this last result from the full time domain response for a ramp input ( $R(s) = \frac{1}{s^2}$ ).

$$M(s) = \frac{Y(s)}{R(s)} = \frac{s+4}{s^2+4s+2}$$

We have the output

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \{M(s) \cdot R(s)\} \\ &= \mathcal{L}^{-1} \left\{ \frac{s+4}{s^2+4s+2} \cdot \frac{1}{s^2} \right\} \end{aligned}$$

Obtaining a partial fraction expansion

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \left\{ -\frac{3.5}{s} + \frac{2}{s^2} - \frac{1.78}{s+3.41} + \frac{3.52}{s+0.586} \right\} \\ &= -3.5 + 2t - 1.78e^{-3.41t} + 3.52e^{-0.586t} \end{aligned}$$

This has the form

$$y(t) = y_{ss}(t) + y_{tr}(t)$$

Where the first two terms represent the steady state response, i.e.  $y_{ss}(t) = -3.5 + 2t$  and the last two terms represent the transient response, i.e.  $y_{tr}(t) = -1.78e^{-3.41t} + 3.52e^{-0.586t}$ .

For a unit ramp,  $r(t) = t$ , the steady state error is given by:

$$\begin{aligned} e_{ss} &= \lim_{t \rightarrow \infty} \left\{ \frac{1}{k_H} r(t) - y_{ss}(t) \right\} \\ &= \frac{1}{0.5} t - (-3.5 + 2t) \\ &= 3.5 \end{aligned}$$

This confirms our previous result.

Figure 3.5 shows the ideal and actual outputs as a function of time. The difference is the error. The error as a function of time is shown in Figure 3.6, where it is seen that it converges to its steady state value of 3.5 after 10 seconds.

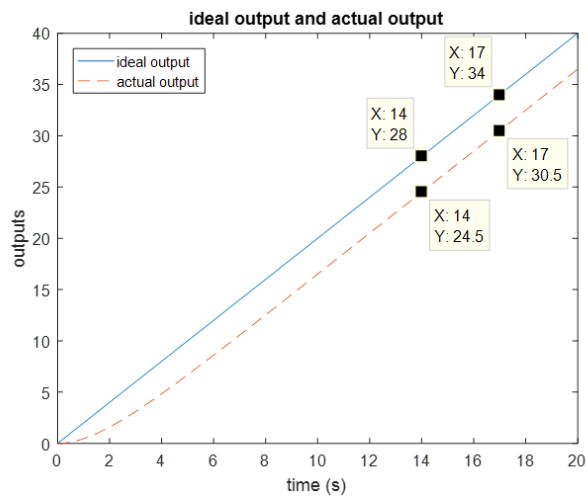


Figure 3.5: The ideal and actual outputs. The difference of 3.5 between the outputs represents the steady state error.

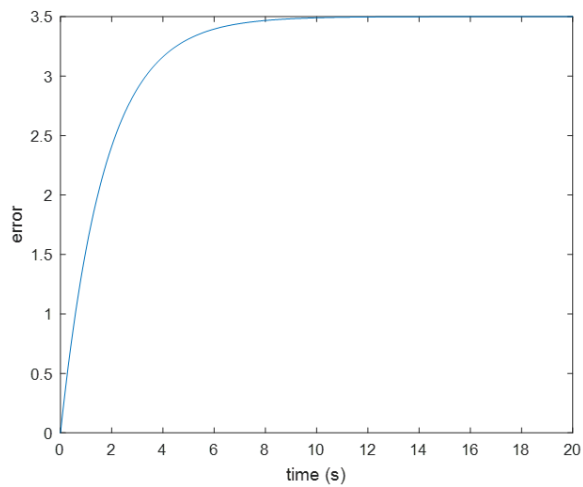


Figure 3.6: The error is seen to reach its steady state value of 3.5 after 10 seconds.

### Step input:

We will now confirm the steady state error result for a unit step input

( $R(s) = \frac{1}{s}$ ) by deriving the full time domain response:

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \{M(s) \cdot R(s)\} \\ &= \mathcal{L}^{-1} \left\{ \frac{s+4}{s^2+4s+2} \cdot \frac{1}{s} \right\} \end{aligned}$$

Obtaining a partial fraction expansion

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \left\{ \frac{2}{s} + \frac{0.061}{s+3.41} - \frac{2.06}{s+0.586} \right\} \\ &= 2 + 0.061e^{-3.41t} + 2.06e^{-0.586t} \end{aligned}$$

This has the form

$$y(t) = y_{ss}(t) + y_{tr}(t)$$

Where the first term represents the steady state response, i.e.  $y_{ss}(t) = 2$ , and the remaining two terms represent the transient response, i.e.  $y_{tr}(t) = 0.061e^{-3.41t} + 2.06e^{-0.586t}$ .

For a unit step input,  $r(t) = 1, t > 0$ , the steady state error is given by:

$$\begin{aligned} e_{ss} &= \lim_{t \rightarrow \infty} \left\{ \frac{1}{k_H} r(t) - y_{ss}(t) \right\} \\ &= \frac{1}{0.5} \cdot 1 - 2 \\ &= 0 \end{aligned}$$

This confirms our previous result.

### 3.2.4 Example 5:

Given a feedback system with,  $G(s) = \frac{1}{s}$  and  $H(s) = \frac{4}{s+4}$ .

- Find the steady state error,  $e_{ss}$ , for unit step, ramp and parabolic inputs.
- Confirm your results for the step and ramp inputs by deriving the full output time response and taking the limit as time goes to infinity.

**Solution:**

- Steps:

- As  $H(s) = \frac{4}{s+4}$ , the constant  $k_H \triangleq \lim_{s \rightarrow 0} H(s) = H(0) = \frac{4}{4} = 1$ .



2) The closed loop gain  $M(s) = \frac{G(s)}{1+G(s)H(s)}$  so that

$$\begin{aligned} M(s) &= \frac{\frac{1}{s}}{1 + \frac{1}{s} \cdot \frac{4}{s+4}} \\ &= \frac{s+4}{(s+2)^2} \end{aligned}$$

and

$$\begin{aligned} 1 - k_H M(s) &= 1 - 1 \cdot \frac{s+4}{(s+2)^2} \\ &= \frac{s(s+3)}{s^2+4s+4} \\ &= \frac{s(s+3)}{(s+2)^2} \end{aligned}$$

Since there is one zero at  $s = 0$ , the system type = 1.

3) With a system type of 1, from the second row of Table 3.2 we see that  $e_{ss} = 0$  and  $e_{ss} = \infty$  for step and parabolic inputs, respectively. For a unit ramp, (with  $A = 1$ ), the error is given by

$$e_{ss} = \frac{1}{k_H} \left[ \frac{(a_1 - b_1 k_H)}{a_0} \right]$$

Noting coefficients of the numerator and denominator of  $M(s)$  we have,  $a_0 = 4$ ,  $a_1 = 4$ ,  $b_1 = 1$  and with  $k_H = 1$ , we find,  $e_{ss} = 0.75$ .

b) We will confirm this last result by deriving the full time domain response for a ramp input ( $R(s) = \frac{1}{s^2}$ ).

$$M(s) = \frac{Y(s)}{R(s)} = \frac{s+4}{(s+2)^2}$$

We have the output

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \{M(s) \cdot R(s)\} \\ &= \mathcal{L}^{-1} \left\{ \frac{s+4}{(s+2)^2} \cdot \frac{1}{s^2} \right\} \end{aligned}$$

Obtaining a partial fraction expansion

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \left\{ -\frac{0.75}{s} + \frac{1}{s^2} + \frac{0.75}{s+2} + \frac{0.5}{(s+2)^2} \right\} \\ &= -0.75 + t + 0.75e^{-2t} + 0.5te^{-2t} \end{aligned}$$

This has the form

$$y(t) = y_{ss}(t) + y_{tr}(t)$$

Where the first two terms represent the steady state response, i.e.  $y_{ss}(t) = -0.75 + t$ , and the remaining two terms represent the transient response, i.e.  $y_{tr}(t) = 0.75e^{-2t} + 0.5te^{-2t}$ .

For a unit step ramp,  $r(t) = t$ , the steady state error is given by:

$$\begin{aligned} e_{ss} &= \lim_{t \rightarrow \infty} \left\{ \frac{1}{k_H} r(t) - y_{ss}(t) \right\} \\ &= \lim_{t \rightarrow \infty} \{t - (-0.75 + t)\} \\ &= 0.75 \end{aligned}$$

### Step input:

We will now confirm the steady state error result for a unit step input ( $R(s) = \frac{1}{s}$ ) by deriving the full time domain response:

$$M(s) = \frac{Y(s)}{R(s)} = \frac{s+4}{(s+2)^2}$$

We have the output

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \{M(s) \cdot R(s)\} \\ &= \mathcal{L}^{-1} \left\{ \frac{s+4}{(s+2)^2} \cdot \frac{1}{s} \right\} \end{aligned}$$

Obtaining a partial fraction expansion

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \left\{ \frac{1}{s} - \frac{1}{s+2} - \frac{1}{(s+2)^2} \right\} \\ &= 1 - e^{-2t} - te^{-2t} \end{aligned}$$

This has the form

$$y(t) = y_{ss}(t) + y_{tr}(t)$$

where  $y_{ss}(t) = 1$  and  $y_{tr}(t) = -e^{-2t} - te^{-2t}$ .

This leads to

$$\begin{aligned} e_{ss} &= \lim_{t \rightarrow \infty} \left\{ \frac{1}{k_H} r(t) - y_{ss}(t) \right\} \\ &= \lim_{t \rightarrow \infty} \{1 - 1\} \\ &= 0 \end{aligned}$$

This confirms the result previously obtained.

### 3.2.5 Example 6:

Given a feedback system with,  $G(s) = \frac{1}{s^2(s+10)}$  and  $H(s) = \frac{8(s+1)}{s+4}$ .

- Find the steady state error,  $e_{ss}$ , for unit step, ramp and parabolic inputs.
- Confirm your results for a unit ramp input by deriving the full output time response and taking the limit as time goes to infinity.

**Solution:**

a) Steps:

- As  $H(s) = \frac{8(s+1)}{s+4}$ , the constant  $k_H = \lim_{s \rightarrow 0} H(s) = H(0) = \frac{8}{4} = 2$ .
- The closed loop gain,  $M(s) = \frac{G(s)}{1+G(s)H(s)}$ , so that

$$\begin{aligned} M(s) &= \frac{\frac{1}{s^2(s+10)}}{1 + \frac{1}{s^2(s+10)} \cdot \frac{8(s+1)}{s+4}} \\ &= \frac{s+4}{s^4 + 14s^3 + 40s^2 + 8s + 8} \end{aligned}$$

and

$$\begin{aligned} 1 - k_H M(s) &= 1 - 2 \cdot \frac{s+4}{s^4 + 14s^3 + 40s^2 + 8s + 8} \\ &= \frac{s(s^3 + 14s^2 + 40s + 6)}{s^4 + 14s^3 + 40s^2 + 8s + 8} \end{aligned}$$

Since there is one zero at  $s = 0$ , the system type = 1.

- With a system type of 1, from the second row of Table 3.2 we see that  $e_{ss} = 0$  and  $e_{ss} = \infty$  for step and parabolic inputs, respectively. For a unit ramp (with  $A = 1$ ), the error is given by

$$e_{ss} = \frac{1}{k_H} \left[ \frac{(a_1 - b_1 k_H)}{a_0} \right]$$

Noting coefficients of the numerator and denominator of  $M(s)$  we have,  $a_0 = 8$ ,  $a_1 = 8$ ,  $b_1 = 1$  and with  $k_H = 2$ , we find  $e_{ss} = 0.375$

- We will confirm this last result by deriving the full time domain response for a ramp input ( $R(s) = \frac{1}{s^2}$ ). Since

$$M(s) = \frac{Y(s)}{R(s)} = \frac{s+4}{s^4 + 14s^3 + 40s^2 + 8s + 8}$$

We have the output

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \{M(s) \cdot R(s)\} \\ &= \mathcal{L}^{-1} \left\{ \frac{s+4}{s^4 + 14s^3 + 40s^2 + 8s + 8} \cdot \frac{1}{s^2} \right\} \end{aligned}$$

Obtaining a partial fraction expansion

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \left\{ -\frac{0.375}{s} + \frac{0.5}{s^2} + \frac{9.3 \cdot 10^{-5}}{s+10} + \frac{2 \cdot 10^{-4}}{s+3.75} \right. \\ &\quad \left. + \frac{0.187 + j0.521}{s+0.067 - j0.454} + \frac{0.187 - j0.521}{s+0.067 + j0.454} \right\} \\ &= -0.375 + 0.5t + 9.3 \cdot 10^{-5} e^{-10t} + 2 \cdot 10^{-4} e^{-3.75t} \\ &\quad + 0.375 e^{-0.067t} \cos(0.454t) - 1.04 e^{-0.067t} \sin(0.454t) \end{aligned}$$

A note at the end of this chapter is provided that illustrates how a complex conjugate pole pair and their associated complex conjugate residues may be converted to a cos/sin representation, as shown above.

The form of the output response is

$$y(t) = y_{ss}(t) + y_{tr}(t)$$

where the first two terms represent the steady state response, i.e.  $y_{ss}(t) = -0.375 + 0.5t$ , and the remaining terms represent the transient response.

This leads to

$$\begin{aligned} e_{ss} &= \lim_{t \rightarrow \infty} \left\{ \frac{1}{k_H} r(t) - y_{ss}(t) \right\} \\ &= \lim_{t \rightarrow \infty} \left\{ \frac{1}{2} t - (-0.375 + 0.5t) \right\} \\ &= 0.375 \end{aligned}$$

Figure 3.7 show the time evolution of the error, which attains a steady state value of 0.375.

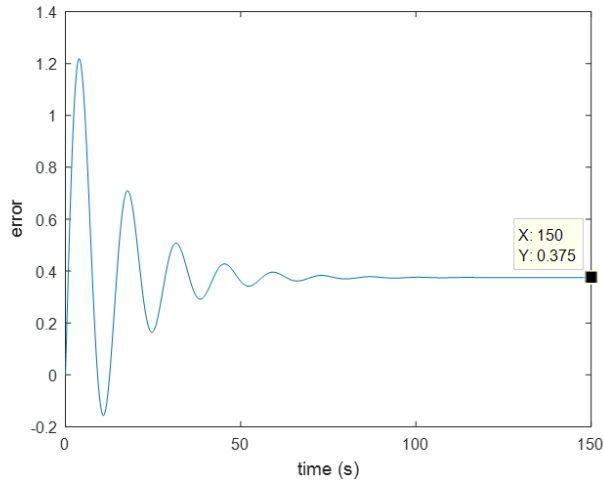


Figure 3.7: The error signal as a function of time.

### 3.2.6 Example 7:

Given a feedback system with,  $G(s) = \frac{1}{s^2(s+10)}$  and  $H(s) = \frac{4(s+1)}{s+4}$ .

- Find the steady state error,  $e_{ss}$ , for unit step, ramp and parabolic inputs.
- Confirm your results for a unit ramp input by deriving the full output time response and taking the limit as time goes to infinity.

**Solution:**

a) Steps:

- As  $H(s) = \frac{4(s+1)}{s+4}$ , the constant  $k_H \triangleq \lim_{s \rightarrow 0} H(s) = H(0) = \frac{4}{4} = 1$ .
- The closed loop gain  $M(s) = \frac{G(s)}{1+G(s)H(s)}$  so that

$$M(s) = \frac{\frac{1}{s^2(s+10)}}{1 + \frac{1}{s^2(s+10)} \cdot \frac{4(s+1)}{s+4}}$$

$$= \frac{s+4}{s^4 + 14s^3 + 40s^2 + 4s + 4}$$

and

$$1 - k_H M(s) = 1 - 1 \cdot \frac{s+4}{s^4 + 14s^3 + 40s^2 + 4s + 4}$$

$$= \frac{s(s^3 + 14s^2 + 40s + 3)}{s^4 + 14s^3 + 40s^2 + 4s + 4}$$

Since there is one zero at  $s = 0$ , the system type = 1.

- 3) With a system type of 1, from the second row of Table 3.2 we see that  $e_{ss} = 0$  and  $e_{ss} = \infty$  for both step and parabolic inputs, respectively. For a unit ramp (with  $A = 1$ ), the error is given by

$$e_{ss} = \frac{1}{k_H} \left[ \frac{(a_1 - b_1 k_H)}{a_0} \right]$$

Noting coefficients of the numerator and denominator of  $M(s)$  we have,  $a_0 = 4$ ,  $a_1 = 4$ ,  $b_1 = 1$  and with  $k_H = 1$ , we find  $e_{ss} = 0.75$ .

- b) We will confirm this last result by deriving the full time domain response for a unit ramp input ( $R(s) = \frac{1}{s^2}$ ).

$$M(s) = \frac{Y(s)}{R(s)} = \frac{s + 4}{s^4 + 14s^3 + 40s^2 + 4s + 4}$$

We have the output

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \{M(s) \cdot R(s)\} \\ &= \mathcal{L}^{-1} \left\{ \frac{s + 4}{s^4 + 14s^3 + 40s^2 + 4s + 4} \cdot \frac{1}{s^2} \right\} \end{aligned}$$

Obtaining a partial fraction expansion

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \left\{ -\frac{0.75}{s} + \frac{1}{s^2} + \frac{9.62 \cdot 10^{-5}}{s + 10} + \frac{9.1 \cdot 10^{-5}}{s + 3.875} \right. \\ &\quad \left. + \frac{0.375 + j1.53}{s + 0.033 - j0.32} + \frac{0.375 - j1.53}{s + 0.033 + j0.32} \right\} \\ &= -0.75 + t + 9.62 \cdot 10^{-5} e^{-10t} + 9.1 \cdot 10^{-5} e^{-3.875t} \\ &\quad + 0.75 e^{-0.033t} \cos(0.32t) - 3.06 e^{-0.033t} \sin(0.32t) \end{aligned}$$

This has the form

$$y(t) = y_{ss}(t) + y_{tr}(t)$$

where the first two terms represent the steady state response. i.e.  $y_{ss}(t) = -0.75 + t$ , and the remaining terms represent the transient response which will decay to zero as time progresses.

This leads to

$$\begin{aligned} e_{ss} &= \lim_{t \rightarrow \infty} \left\{ \frac{1}{k_H} r(t) - y(t) \right\} \\ &= \lim_{t \rightarrow \infty} \{t - (-0.75 + t)\} \\ &= 0.75 \end{aligned}$$

Figure 3.8 show the time evolution of the error, which attains a steady state value of 0.75.

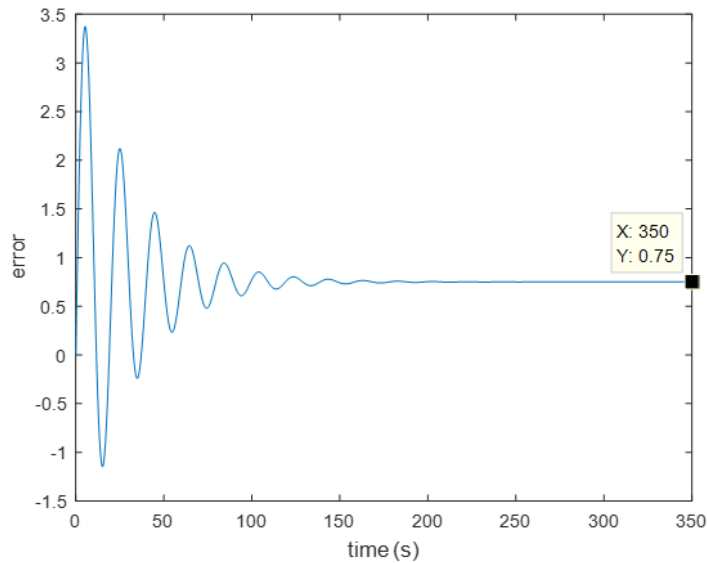


Figure 3.8: The error signal as a function of time.

### 3.2.7 Example 8:

This final example will illustrate both the non-unity gain and unity gain feedback methods in assessing steady state error.

Given a feedback system with,  $G(s) = \frac{4(s+1)}{s^2(s+10)(s+4)}$  and  $H(s) = 1$ .

- a) Find the steady state error,  $e_{ss}$ , for unit step, ramp and parabolic inputs.
- b) Confirm your results for a unit ramp input by deriving the full output time response and taking the limit as time goes to infinity.

**Solutions:**

**Non unity gain method:**

a) Steps:

1) As  $H(s) = 1$ , the constant  $k_H \triangleq \lim_{s \rightarrow 0} H(s) = H(0) = 1$ .

2) The closed loop gain  $M(s) = \frac{G(s)}{1+G(s)H(s)} = \frac{G(s)}{1+G(s)}$  so that

$$\begin{aligned}
M(s) &= \frac{\frac{4(s+1)}{s^2(s+10)(s+4)}}{1 + \frac{4(s+1)}{s^2(s+10)(s+4)}} \\
&= \frac{4s+4}{s^4 + 14s^3 + 40s^2 + 4s + 4}
\end{aligned}$$

and

$$\begin{aligned}
1 - k_H M(s) &= 1 - 1 \cdot \frac{4s+4}{s^4 + 14s^3 + 40s^2 + 4s + 4} \\
&= \frac{s^2(s^2 + 14s + 40)}{s^4 + 14s^3 + 40s^2 + 4s + 4}
\end{aligned}$$

Since there are two zeros at  $s = 0$ , the system type = 2.

- 3) With a system type of 2, from the second row of Table 3.2 we see that  $e_{ss} = 0$  and  $e_{ss} = 0$  for both step and ramp inputs, respectively. For a unit parabola (with  $A = 1$ ), the error is given by

$$e_{ss} = \frac{1}{k_H} \left[ \frac{(a_2 - b_2 k_H)}{a_0} \right]$$

Noting coefficients of the numerator and denominator of  $M(s)$  we have,  $a_0 = 4$ ,  $a_2 = 40$ ,  $b_2 = 0$  and with  $k_H = 1$ , we find  $e_{ss} = 10$ .

- b) We will confirm this result by deriving the full time domain response for a unit parabolic input ( $R(s) = \frac{1}{s^3}$ ).

$$M(s) = \frac{Y(s)}{R(s)} = \frac{4s+4}{s^4 + 14s^3 + 40s^2 + 4s + 4}$$

We have the output

$$\begin{aligned}
y(t) &= \mathcal{L}^{-1} \{M(s) \cdot R(s)\} \\
&= \mathcal{L}^{-1} \left\{ \frac{4s+4}{s^4 + 14s^3 + 40s^2 + 4s + 4} \cdot \frac{1}{s^3} \right\}
\end{aligned}$$



Obtaining a partial fraction expansion

$$\begin{aligned}
 y(t) &= \mathcal{L}^{-1} \left\{ -\frac{10}{s} + \frac{1}{s^3} - \frac{5.72 \cdot 10^{-5}}{s+10} + \frac{2.15 \cdot 10^{-3}}{s+3.875} \right. \\
 &\quad \left. + \frac{5-j0.53}{s+0.033-j0.32} + \frac{5+j0.53}{s+0.033+j0.32} \right\} \\
 &= -10 + \frac{t^2}{2} + 5.7 \cdot 10^{-5} e^{-10t} + 2.15 \cdot 10^{-3} e^{-3.875t} \\
 &\quad + 10e^{-0.033t} \cos(0.32t) + 1.06e^{-0.033t} \sin(0.32t)
 \end{aligned}$$

This has the form

$$y(t) = y_{ss}(t) + y_{tr}(t)$$

Where the first two terms represent the steady state components (i.e.  $y_{ss}(t) = -10 + \frac{t^2}{2}$ ) and the remaining terms represent the transient response which will decay to zero as time progresses.

This leads to

$$\begin{aligned}
 e_{ss} &= \lim_{t \rightarrow \infty} \left\{ \frac{1}{k_H} r(t) - y(t) \right\} \\
 &= \lim_{t \rightarrow \infty} \left\{ \frac{t^2}{2} - \left( -10 + \frac{t^2}{2} \right) \right\} \\
 &= 10
 \end{aligned}$$

Figure 3.9 shows the time evolution of output, both the ideal, i.e.  $\frac{1}{k_H} r(t)$ , and the actual outputs. However, due the axis scaling involved the outputs appear as identical. By looking closely at a small portion of the output responses the difference between the ideal and actual outputs can be seen. This is shown in Figure 3.10 where the difference can be seen to have a value of 10, which represents the steady state error. Figure 3.11 shows the time evolution of the error.

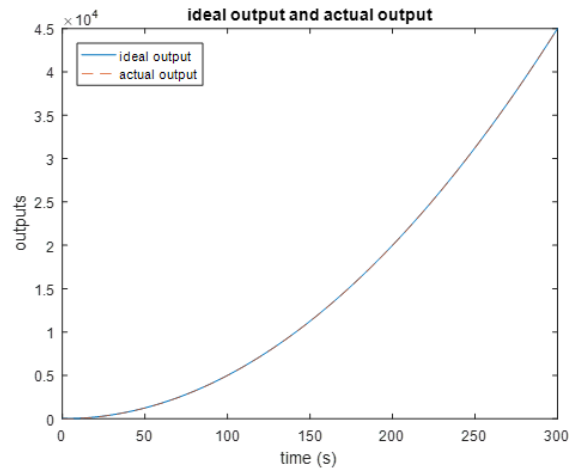


Figure 3.9: The ideal output and actual output are indistinguishable, however there is a difference between these signals which attains a value of 10 in steady state.

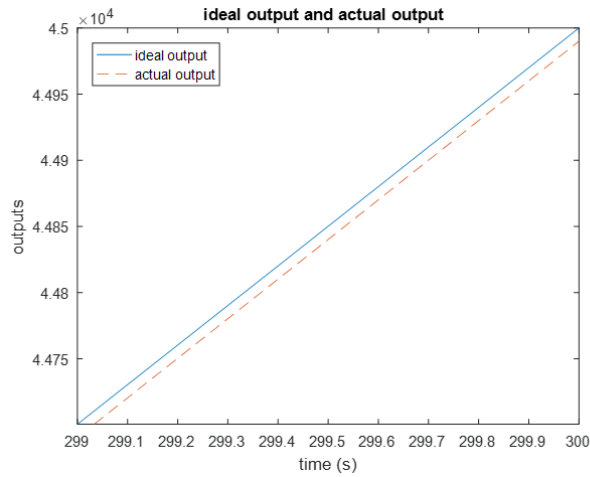


Figure 3.10: The difference between the ideal and actual outputs can be seen here when looking at a small portion of the time response. The difference seen above has a value of 10, the steady state error.

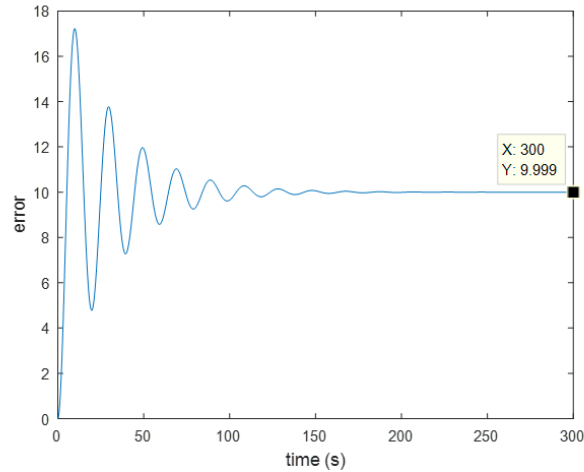


Figure 3.11: The error signal as a function of time settles to a steady state value of 10.

**Unity gain method:**

Given that  $G(s) = \frac{4(s+1)}{s^2(s+10)(s+4)}$  and  $H(s) = 1$ , which represents unity gain feedback, the previously described procedure to assess steady state error may be followed.

Following the procedure given for unity gain feedback system, we first determine the system type by noting the number of poles appearing at  $s = 0$  of transfer function  $G(s)$ . The type number = 2. Consulting Table 3.1 across the third row of this table which corresponds to a system of type number 2, it can be seen that  $e_{ss} = 0$  for both the step and ramp inputs and that for a parabolic input,  $e_{ss} = \frac{1}{K_a}$ , (for  $A = 1$ ) where  $K_a$  is the acceleration error constant given by  $K_a = \lim_{s \rightarrow 0} s^2 \cdot G(s)$

For the present system

$$\begin{aligned}
 K_a &= \lim_{s \rightarrow 0} s^2 \cdot \frac{4(s+1)}{s^2(s+10)(s+4)} \\
 &= \frac{1}{10}
 \end{aligned}$$

Therefore the steady state error is

$$\begin{aligned}
 e_{ss} &= \frac{1}{K_a} \\
 &= 10
 \end{aligned}$$

This agrees with our previous result.

### 3.3 Note: partial fraction containing a complex pole pair

#### Inverse Laplace transformation simplification of a partial fraction expansion containing a complex conjugate pole pair

Given a complex conjugate pole pair,  $p_1$  and  $p_1^*$ , where  $p_1 = \alpha + j\beta$  and  $p_1^* = \alpha - j\beta$ , with corresponding residues,  $k_1$  and  $k_1^*$ , where  $k_1 = a + jb$  and  $k_1^* = a - jb$ , the following simplification can be made when determining the inverse Laplace transform.

$$\begin{aligned}\mathcal{L}^{-1} \left[ \frac{k_1}{s - p_1} + \frac{k_1^*}{s - p_1^*} \right] &= \mathcal{L}^{-1} \left[ \frac{a + jb}{s - \alpha - j\beta} + \frac{a - jb}{s - \alpha + j\beta} \right] \\ &= \mathcal{L}^{-1} \left[ \frac{(a + jb)(s - \alpha + j\beta) + (a - jb)(s - \alpha - j\beta)}{(s - \alpha - j\beta)(s - \alpha + j\beta)} \right] \\ &= \mathcal{L}^{-1} \left[ \frac{2a(s - \alpha) - 2b\beta}{(s - \alpha)^2 + \beta^2} \right] \\ &= 2a\mathcal{L}^{-1} \left[ \frac{(s - \alpha)}{(s - \alpha)^2 + \beta^2} \right] - 2b\mathcal{L}^{-1} \left[ \frac{\beta}{(s - \alpha)^2 + \beta^2} \right] \\ &= 2ae^{\alpha t} \cos(\beta t) - 2be^{\alpha t} \sin(\beta t)\end{aligned}$$

# Chapter 4

## Bode Plots

### Introduction

In the following we will present magnitude and phase asymptotic approximations to a number of basic transfer functions. The value of these asymptotic approximations is that simplified mathematical expressions may be used to precisely describe the asymptote. In the case of the magnitude characteristic it is generally displayed with magnitude in dB versus frequency presented on a log scale. When plotted using these scales, it can be well approximated using straight line asymptotic segments. It is these segments that exact formulas may be presented. Note however that annotations on the magnitude plot use the absolute value rather than the relative value which the dB value represents.

Magnitude in dB is obtained by:

$$|H(s)|_{dB} = 20 \log [H(s)]|_{s=j\omega} \quad (4.1)$$

The phase plot is represented by phase angle in degrees versus log frequency. The phase plot of a transfer function is obtained by setting  $s = j\omega$  where  $j = \sqrt{-1}$ , and where  $\omega$  represents the angular radian frequency. In sections of this book, we will prefer to use  $f = \frac{\omega}{2\pi}$  when dealing with actual frequencies.

$$\angle H(s) = \angle H(s)|_{s=j\omega} \quad (4.2)$$

### 4.1 Simple Gain

The simplest transfer function is that of constant gain which we will denote as  $A$ , so that the transfer function is given as:

$$H(s) = A \quad (4.3)$$

In this case the asymptotic and exact magnitude and phase characteristics are the same and are shown in figure 4.1. As just mentioned above, note that the magnitude line is annotated with its absolute value, rather than its dB value.

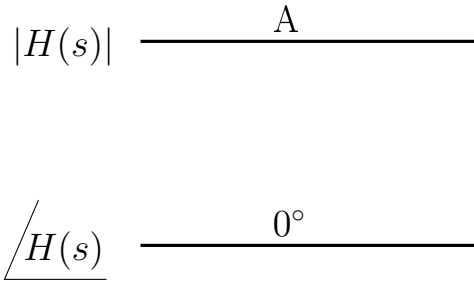


Figure 4.1: Simple Gain

## 4.2 Pole at Zero

The next transfer function considered is that of a single pole at zero frequency, as given by:

$$H(s) = \frac{A}{s} \quad (4.4)$$

This transfer function is that of an integrator. The case of a finite pole frequency different than zero is tackled below. The asymptotic magnitude and phase plots are shown in figure 4.2. With reference to this figure, we can interpret the parameter  $A$  as the gain of the transfer function at the angular frequency of  $\omega = 1$ .

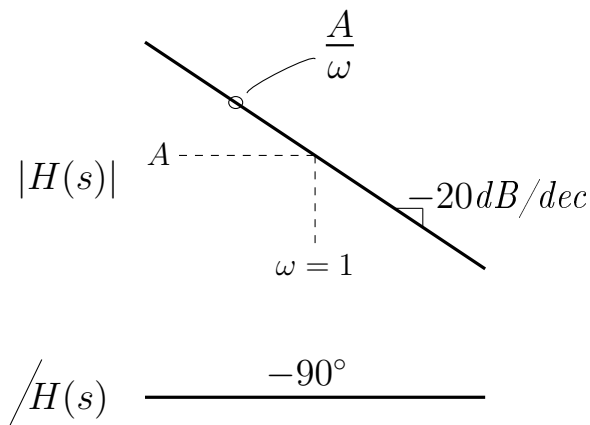


Figure 4.2: Pole at Zero

Alternatively, this same transfer function may be represented as  $H(s) = \frac{\omega_o}{s}$  where  $\omega_o = A$ . With reference to figure 4.3, we see that  $\omega_o$  may be interpreted as the value of frequency for which the transfer function has a gain of unity.

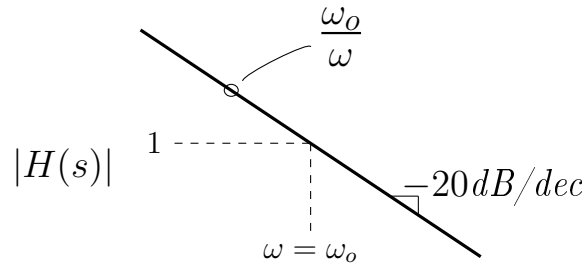


Figure 4.3: Pole at Zero

### 4.3 Zero at Zero

The next transfer function considered is that of a single zero at zero frequency, as given by:

$$H(s) = As \quad (4.5)$$

This transfer function is that of a differentiator. The case of a finite zero frequency different than zero is tackled below. The asymptotic magnitude and phase plots are shown in figure 4.4. With reference to this figure, we can interpret the parameter  $A$  as the gain of the transfer function at the angular frequency of  $\omega = 1$ .

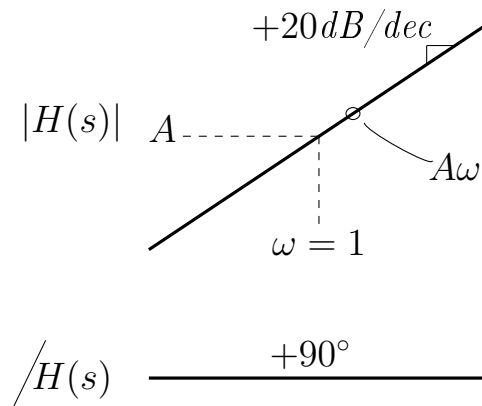


Figure 4.4: Zero at Zero

Alternatively, this same transfer function may be represented as  $H(s) = \frac{s}{\omega_o}$  where  $\omega_o = \frac{1}{A}$ . With reference to figure 4.5, we see that  $\omega_o$  may be interpreted as the value of frequency for which the transfer function has a gain magnitude of unity.

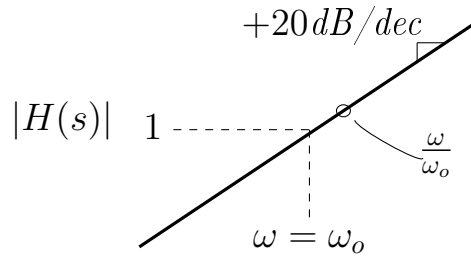


Figure 4.5: Zero at Zero

#### 4.4 Pole at $\omega_o$

The next transfer function considered is that of a single pole at a frequency of  $\omega_o$ , as given by:

$$H(s) = \frac{A}{1 + \frac{s}{\omega_o}} \quad (4.6)$$

The asymptotic magnitude and phase plots are shown in figure 4.6.

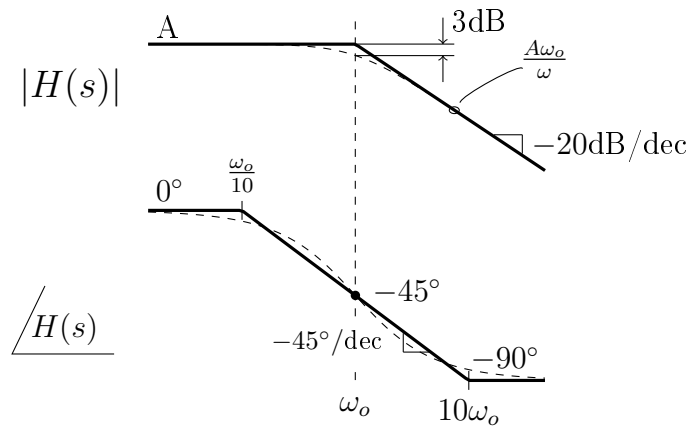


Figure 4.6: Pole at  $\omega_o$



## 4.5 Zero at $\omega_o$

The next transfer function considered is that of a single zero at a frequency of  $\omega_o$ , as given by:

$$H(s) = A \left( 1 + \frac{s}{\omega_o} \right) \quad (4.7)$$

The asymptotic magnitude and phase plots are shown in figure 4.7.

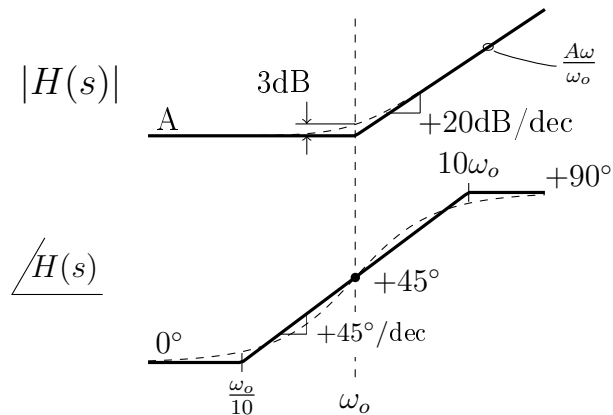


Figure 4.7: Zero at  $\omega_o$

## 4.6 Right Half Plane Zero at $\omega_o$

The next transfer function considered is that of a single right half plane zero at a frequency of  $\omega_o$ , as given by:

$$H(s) = A \left( 1 - \frac{s}{\omega_o} \right) \quad (4.8)$$

The asymptotic magnitude and phase plots are shown in figure 4.8.

## 4.7 Complex Pole Pair with Resonant Frequency at $\omega_o$

The next transfer function considered is that of a complex pole pair with resonant frequency at  $\omega_o$ , as given by:

$$H(s) = \frac{A}{1 + \frac{s}{Q\omega_o} + \left( \frac{s}{\omega_o} \right)^2} \quad (4.9)$$

The asymptotic magnitude and phase plots are shown in figure 4.9.

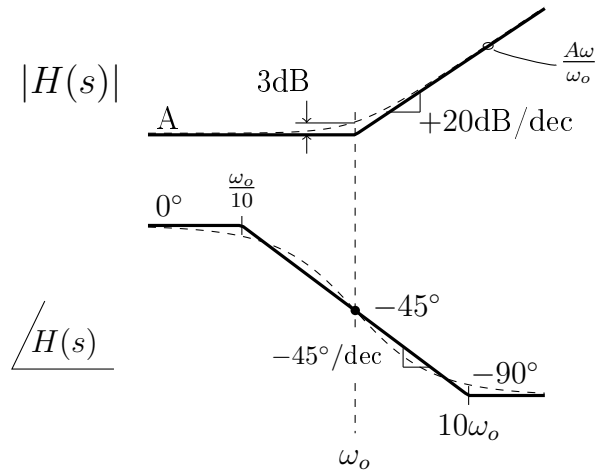


Figure 4.8: Right Half Plane Zero at  $\omega_o$

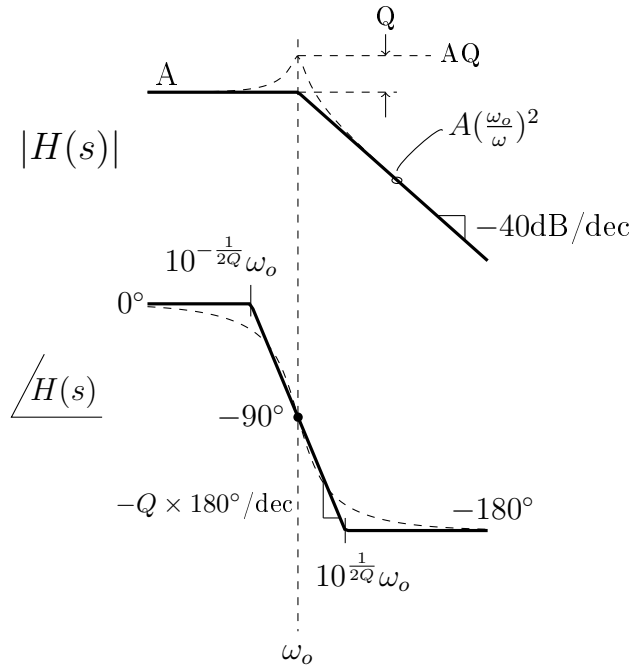


Figure 4.9: Second Order Complex Pole at  $\omega_o$

## 4.8 Complex Zero Pair with Resonant Frequency at $\omega_o$

The next transfer function considered is that of a complex zero pair with resonant frequency of  $\omega_o$ , as given by:

$$H(s) = A \left( 1 + \frac{s}{Q\omega_o} + \left( \frac{s}{\omega_o} \right)^2 \right) \quad (4.10)$$

The asymptotic magnitude and phase plots are shown in figure 4.10.

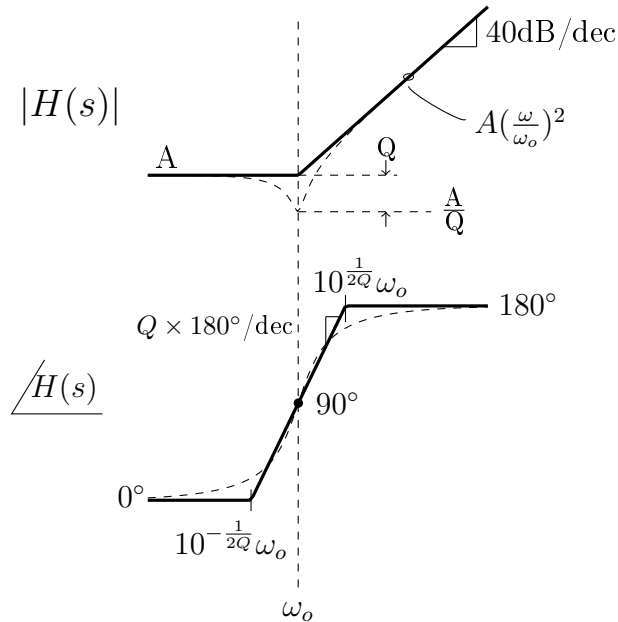


Figure 4.10: Second Order Complex Zero at  $\omega_o$

## 4.9 Composite Transfer Functions

Having just presented the asymptotic Bode plots for some basic transfer function we will next demonstrate how these may be used in constructing asymptotic Bode plots for more complicated transfer functions. This then will allow forming simplified mathematical expressions for magnitude and phase which will be used in compensator design.

As an example we present the following transfer function:

$$T(s) = \frac{T_o}{\left( 1 + \frac{s}{\omega_1} \right) \left( 1 + \frac{s}{\omega_2} \right) \left( 1 + \frac{s}{\omega_3} \right)} \quad (4.11)$$

where

$$T_o = 250, \omega_1 = 2\pi(10), \omega_2 = 2\pi(100), \omega_3 = 2\pi(300)$$

Somewhat arbitrarily we can identify three constituent basic transfer functions which when multiplied together form the composite transfer function.

$$T(s) = \frac{T_o}{\underbrace{\left(1 + \frac{s}{\omega_1}\right)}_{T_a(s)}} \frac{1}{\underbrace{\left(1 + \frac{s}{\omega_2}\right)}_{T_b(s)}} \frac{1}{\underbrace{\left(1 + \frac{s}{\omega_3}\right)}_{T_c(s)}} \quad (4.12)$$

We will first construct the magnitude plot. Figure 4.11a shows the asymptotic response for each of the components. Note that the magnitude annotations are given in absolute terms so that the composite magnitude is simply the product of the constituent magnitudes in the relevant frequency interval. In contrast, the slopes expressed in dB are simply added together. The result of the composite magnitude is shown in figure 4.11b.

Similarly the constituent phase response is shown in figure 4.12a; and composite phase plot is shown in figure 4.12b.

The final form of the asymptotic Bode plot is given in its customary form with magnitude response placed above the phase response, shown in figure 4.13.

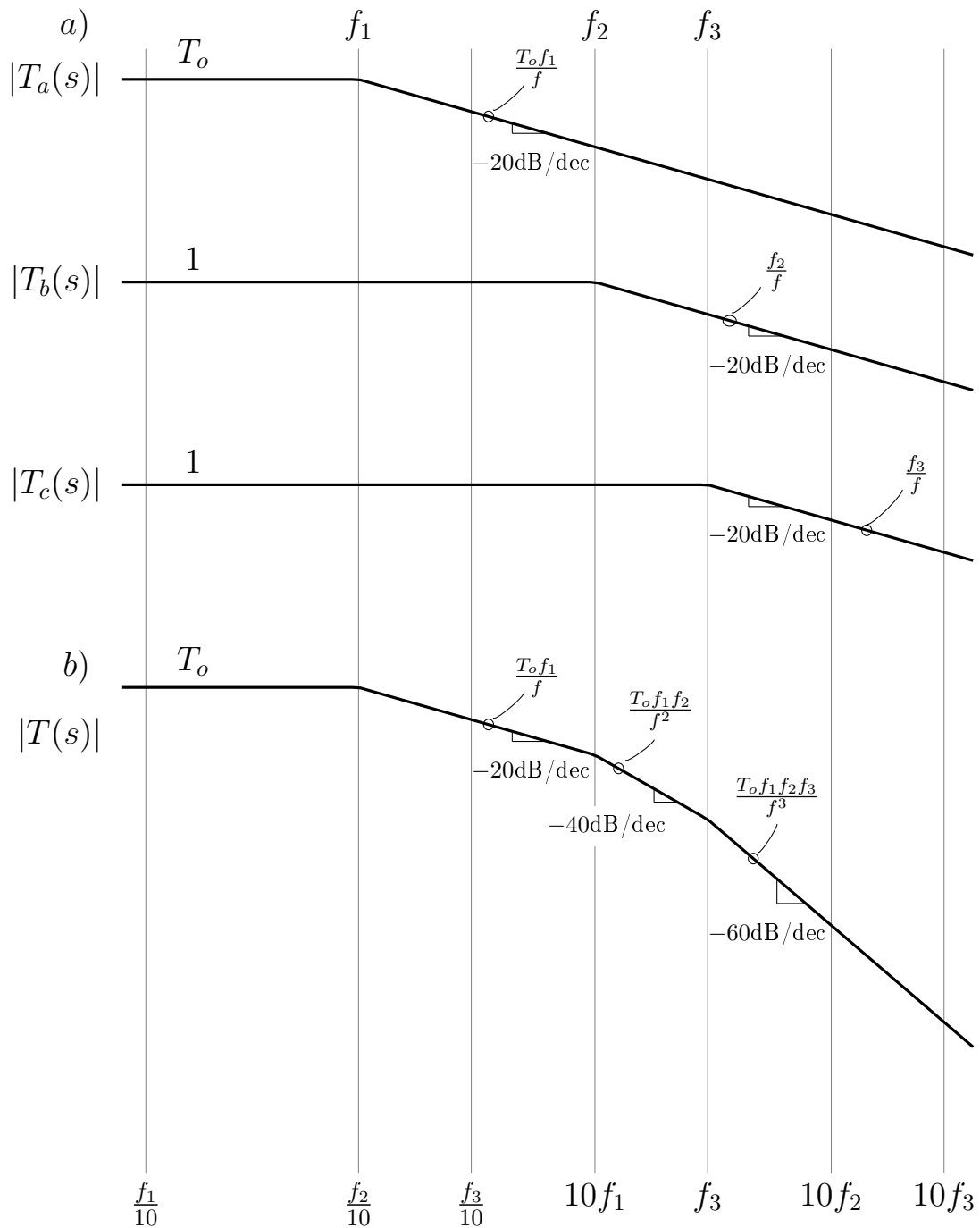


Figure 4.11: a) Asymptotic magnitude plots for the constituent transfer functions, b) Asymptotic magnitude plot for the composite transfer function

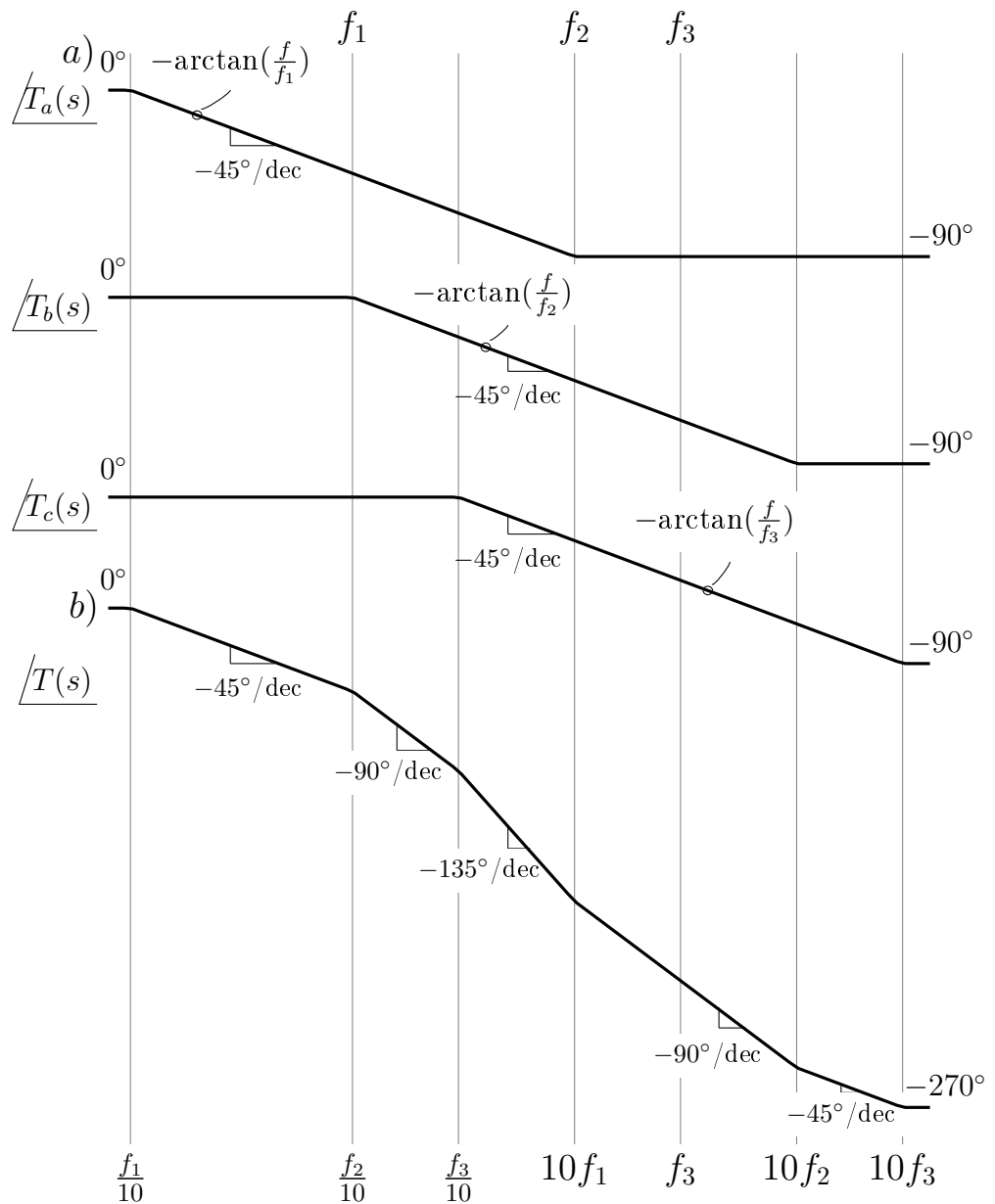


Figure 4.12: a) Asymptotic phase plots for the constituent transfer functions, b) Asymptotic plot plot for the composite transfer function

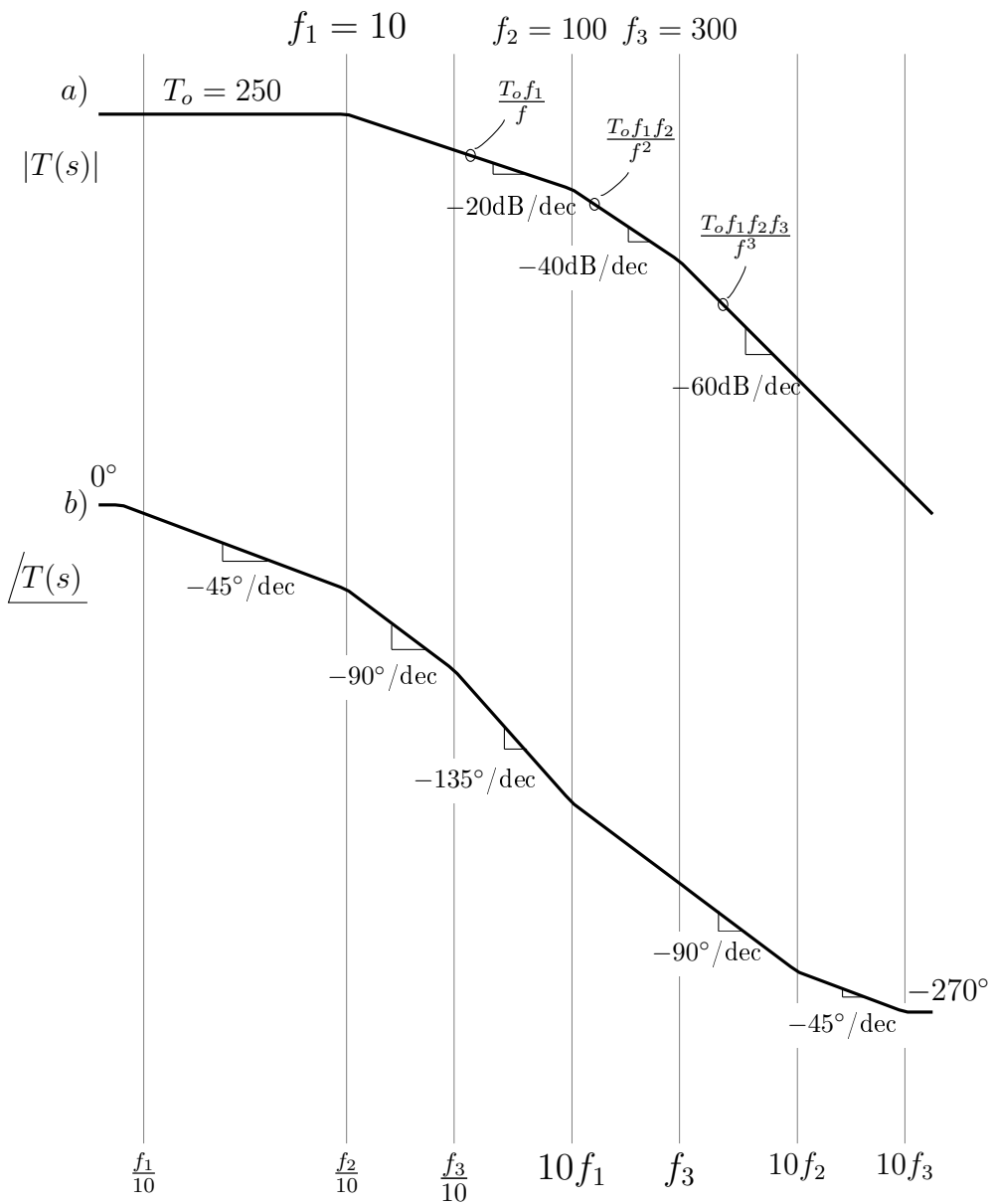

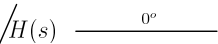
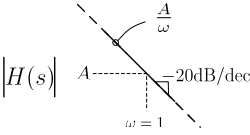
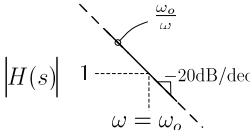
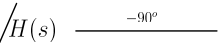
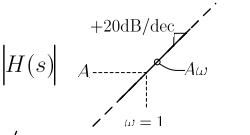
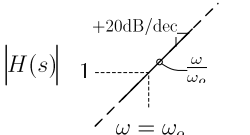
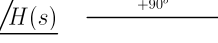
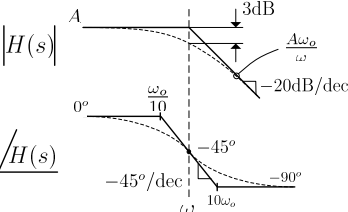
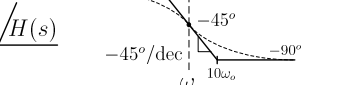
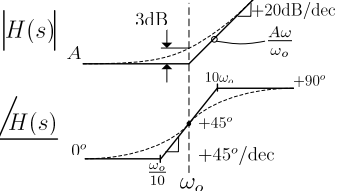
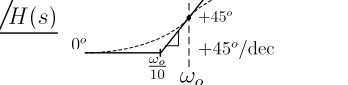
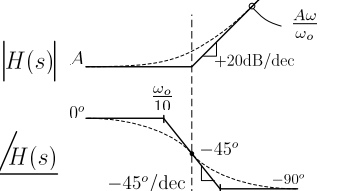
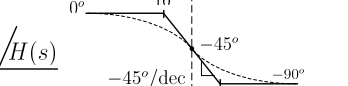


Figure 4.13: Final constructed asymptotic Bode plot showing, a) asymptotic magnitude response, b) asymptotic phase response

## 4.10 Summary of Bode Plots

$H(s) = A$	$ H(s) $ 	
Simple Gain	$\angle H(s)$ 	
$H(s) = \frac{A}{s}$	$ H(s) $  Or: $H(s) = \frac{\omega_o}{s}$ $\omega_o = A$ 	
Pole at Zero	$\angle H(s)$ 	
$H(s) = As$	$ H(s) $  Or: $H(s) = \frac{s}{\omega_o}$ $\omega_o = \frac{1}{A}$ 	
Zero at Zero	$\angle H(s)$ 	
$H(s) = \frac{A}{1 + \frac{s}{\omega_o}}$	$ H(s) $  $\angle H(s)$ 	$(\text{If } \omega \gg \omega_o,  H(s)  = \frac{A}{(\omega/\omega_o)} = \frac{A\omega_o}{\omega})$ Maximum Error @ $\omega_o = 3\text{dB}$ Maximum Error @ $\frac{\omega_o}{10}$ & $10\omega_o = 5.7^\circ$ Exact Phase: $-\tan^{-1}\left(\frac{\omega}{\omega_o}\right), \forall \omega$ Approx. Phase: $-45^\circ \log_{10}\left(\frac{10\omega}{\omega_o}\right),$ $\frac{\omega_o}{10} \leq \omega \leq 10\omega_o$
$H(s) = A\left(1 + \frac{s}{\omega_o}\right)$	$ H(s) $  $\angle H(s)$ 	Maximum Error @ $\omega_o = 3\text{dB}$ Maximum Error @ $\frac{\omega_o}{10}$ & $10\omega_o = 5.7^\circ$ Exact Phase: $\tan^{-1}\left(\frac{\omega}{\omega_o}\right), \forall \omega$ Approx. Phase: $45^\circ \log_{10}\left(\frac{10\omega}{\omega_o}\right),$ $\frac{\omega_o}{10} \leq \omega \leq 10\omega_o$
$H(s) = A\left(1 - \frac{s}{\omega_o}\right)$	$ H(s) $  $\angle H(s)$ 	Maximum Error @ $\omega_o = 3\text{dB}$ Maximum Error @ $\frac{\omega_o}{10}$ & $10\omega_o = 5.7^\circ$ Exact Phase: $-\tan^{-1}\left(\frac{\omega}{\omega_o}\right), \forall \omega$ Approx. Phase: $-45^\circ \log_{10}\left(\frac{10\omega}{\omega_o}\right),$ $\frac{\omega_o}{10} \leq \omega \leq 10\omega_o$
Right Half Plane Zero at $\omega_o$		



$H(s) = \frac{A}{1 + \frac{s}{Q\omega_o} + \left(\frac{s}{\omega_o}\right)^2}$ <p>Second Order Complex Pole</p>		$\omega_o = \text{Corner Frequency}$ $Q > \frac{1}{2} \implies \text{Complex Roots}$ $Q = \text{Quality Factor: Exact Gain @ } \omega_o$ $\text{Approximate Maximum Value}$ $\text{Exact Phase: } -\tan^{-1} \left[ \frac{\frac{1}{Q} \frac{\omega}{\omega_o}}{1 - \left(\frac{\omega}{\omega_o}\right)^2} \right], \forall \omega$
$H(s) = A \left( 1 + \frac{s}{Q\omega_o} + \left(\frac{s}{\omega_o}\right)^2 \right)$ <p>Second Order Complex Zero</p>		$\omega_o = \text{Corner Frequency}$ $Q > \frac{1}{2} \implies \text{Complex Roots}$ $\text{Exact Phase: } \tan^{-1} \left[ \frac{\frac{1}{Q} \frac{\omega}{\omega_o}}{1 - \left(\frac{\omega}{\omega_o}\right)^2} \right], \forall \omega$



# Chapter 5

## Compensator Design

### 5.1 Design Procedure

#### 5.1.1 Introduction

In this chapter we will demonstrate a procedure for designing frequency compensators for the standard feedback configuration shown in Figure 5.1.  $G(s)$ , as before, represents the plant transfer function;  $H(s)$  represents the feedback gain and is used to set the closed loop gain and  $G_c(s)$  represents the compensator.

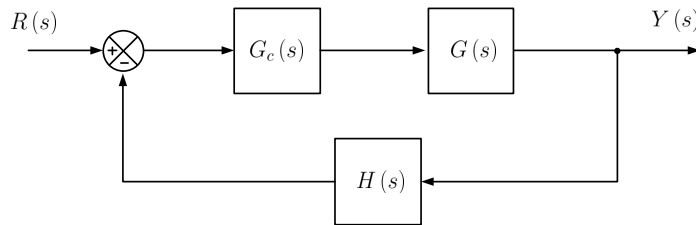


Figure 5.1: Feedback System Block Diagram

To demonstrate the design procedure, in the sequel we will use a plant and feedback gain with the following transfer functions:

$$G(s) = \frac{G_o}{\left(1 + \frac{s}{\omega_1}\right) \left(1 + \frac{s}{\omega_2}\right) \left(1 + \frac{s}{\omega_3}\right)} \quad (5.1)$$

$$H(s) = k \quad (5.2)$$

where  $G_o = 500$ ,  $\omega_1 = 2\pi(10)$ ,  $\omega_2 = 2\pi(100)$ ,  $\omega_3 = 2\pi(300)$ , and  $k = 0.5$ .

The compensators considered in the sequel are the following:

1) Proportional (P) compensator:

$$G_c(s) = k_p \quad (5.3)$$

2) Dominant pole (I, integrator) compensator:

$$G_c(s) = \frac{\omega_I}{s} \quad (5.4)$$

3) Dominant pole with zero (PI, proportional plus integrator) compensator:

$$G_c(s) = \frac{\omega_I}{s} \left( 1 + \frac{s}{\omega_z} \right) \quad (5.5)$$

4) Lead compensator:

$$G_c(s) = G_{co} \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{\omega_p}}, \quad \omega_z < \omega_p \quad (5.6)$$

5) Lead with integrator and zero compensator

$$G_c(s) = \frac{\omega_I \left( 1 + \frac{s}{\omega_{z1}} \right) \left( 1 + \frac{s}{\omega_{z2}} \right)}{s \left( 1 + \frac{s}{\omega_p} \right)} \quad (5.7)$$

The first three compensators may be considered to be members of the three term controller, PID (proportional, integral, derivative), family of compensators. We will see that as the complexity of the compensator increases the performance also increases. The performance measures used are the rise time, settling time and percentage overshoot of the step response. For the simpler compensators, i.e. proportional and dominant pole compensators, only one design parameter is needed to be found. For the most involved compensator, the lead with integrator and zero compensator, there are a total of four design parameters to be determined.

### 5.1.2 Uncompensated System

We start our evaluation with the uncompensated loop gain  $T(s) = kG_c(s)G(s)$ , where  $G_c(s) = 1$ . The loop gain is given as

$$T(s) = \frac{T_o}{\left( 1 + \frac{s}{\omega_1} \right) \left( 1 + \frac{s}{\omega_2} \right) \left( 1 + \frac{s}{\omega_3} \right)} \quad (5.8)$$

where

$$T_o = G_o k = 500 \cdot 0.5 = 250$$

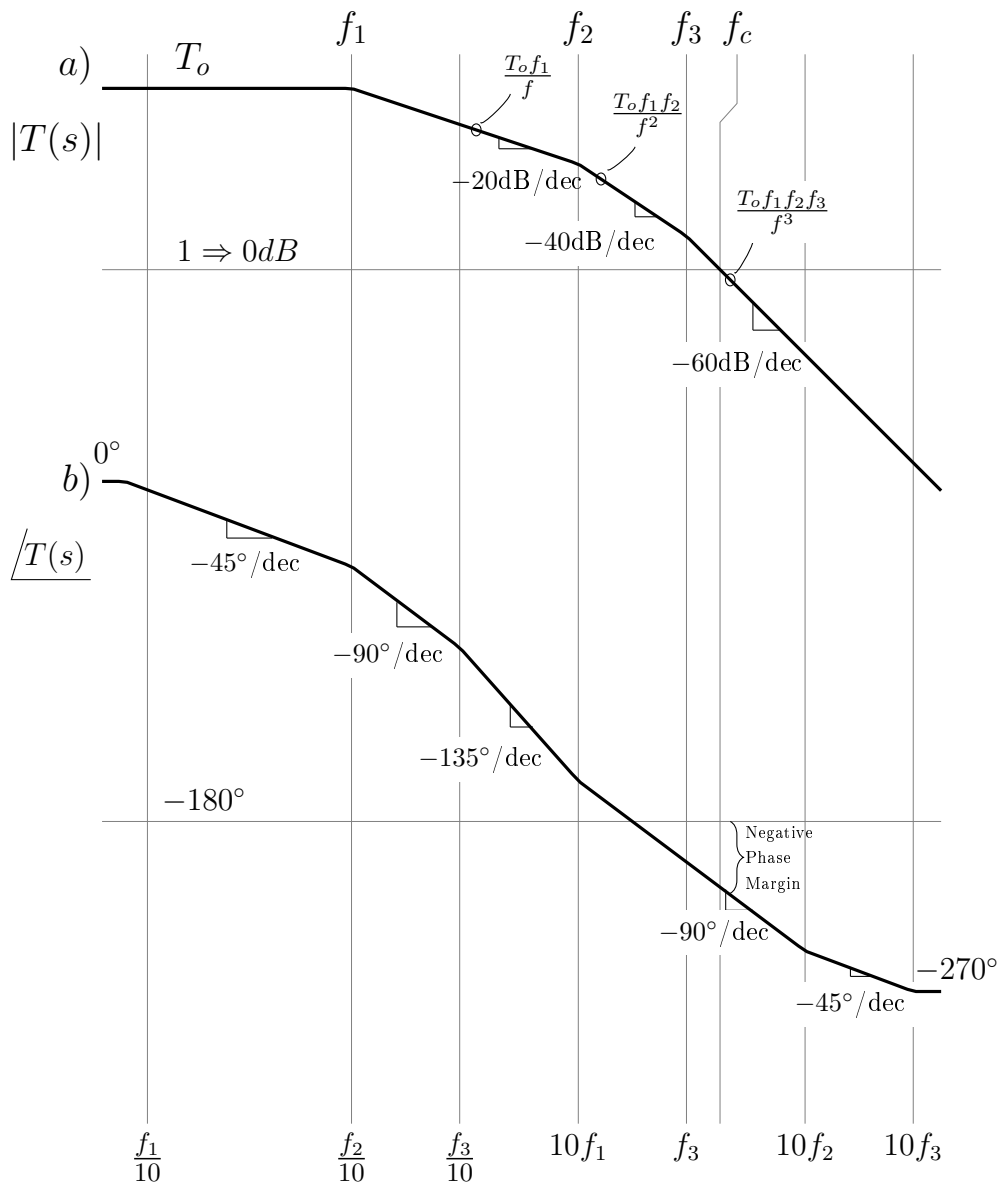


Figure 5.2: Bode Plot: Uncompensated System

$$\omega_1 = 2\pi(10), \omega_2 = 2\pi(100), \omega_3 = 2\pi(300)$$

We construct the Bode plot of the loop gain, shown in Figure 5.2. Using this constructed plot we can easily determine simplified (approximate) expressions

for the  $f_c$ , the unity gain crossover frequency, and  $PM$ , the phase margin:

$$\frac{T_o f_1 f_2 f_3}{f_c^3} = 1 \implies f_c = \sqrt[3]{T_o f_1 f_2 f_3} \quad (5.9)$$

$$PM = 180 - \arctan\left(\frac{f_c}{f_1}\right) - \arctan\left(\frac{f_c}{f_2}\right) - \arctan\left(\frac{f_c}{f_3}\right) \quad (5.10)$$

Equations (5.9) and (5.10) result in  $f_c = 422$  Hz and  $PM = -40^\circ$ .

In a similar fashion we can also determine  $f_{GM}$ , the frequency at which the phase reaches  $-180^\circ$ , and subsequently the gain margin:

$$-180 = -\arctan\left(\frac{f_{GM}}{f_1}\right) - \arctan\left(\frac{f_{GM}}{f_2}\right) - \arctan\left(\frac{f_{GM}}{f_3}\right) \quad (5.11)$$

$$GM = -20 \log\left(\frac{T_o f_1 f_2}{f_{GM}^2}\right) \quad (5.12)$$

Evaluating 5.11 and 5.12 results in  $f_{GM} = 184$  Hz and  $GM = -17.3$  dB, respectively.

Figure 5.3 is a Matlab Bode plot of the uncompensated loop gain produced using the 'margin' command. Matlab uses the unapproximated transfer function models and so is able to accurately determine the margins and associated frequencies:  $f_c = 385$  Hz,  $PM = -36.1^\circ$ ,  $f_{GM} = 184$  Hz and  $GM = -14.8$  dB. A side by side comparison of these results with those from the asymptotic line analysis is shown in Table 5.1.

Table 5.1: Uncompensated System margins

	PM ( $^\circ$ )	$f_C$ (Hz)	GM (dB)	$f_{GM}$ (Hz)
<b>Asymptotes</b>	-40	422	-17.3	184
<b>Matlab</b>	-36.1	385	-14.8	184

The phase margin test indicates that the uncompensated closed loop system is unstable. A Matlab time domain simulation of the step response of the uncompensated system is shown in Figure 5.4. The output quickly becomes unbounded for the unit step input indicative of an unstable system. So compensation is needed to make the system stable and further to improve the performance.

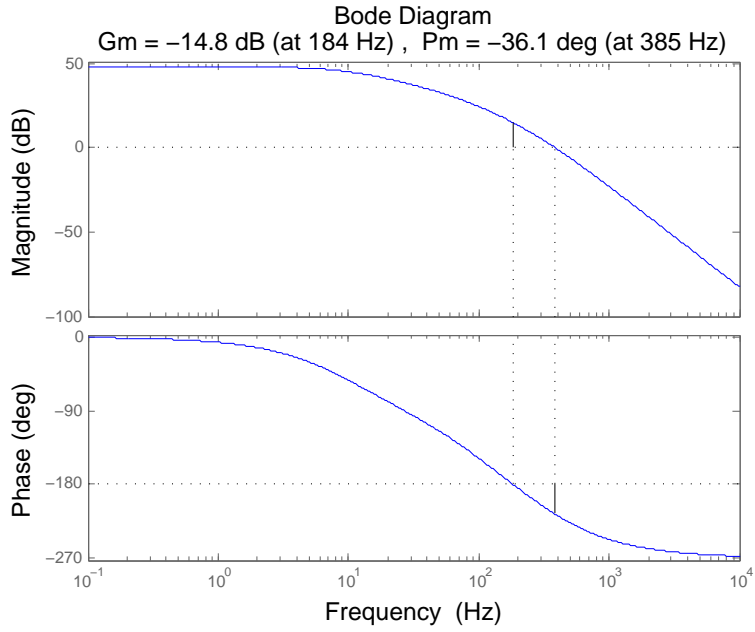


Figure 5.3: Matlab Analysis of Uncompensated System

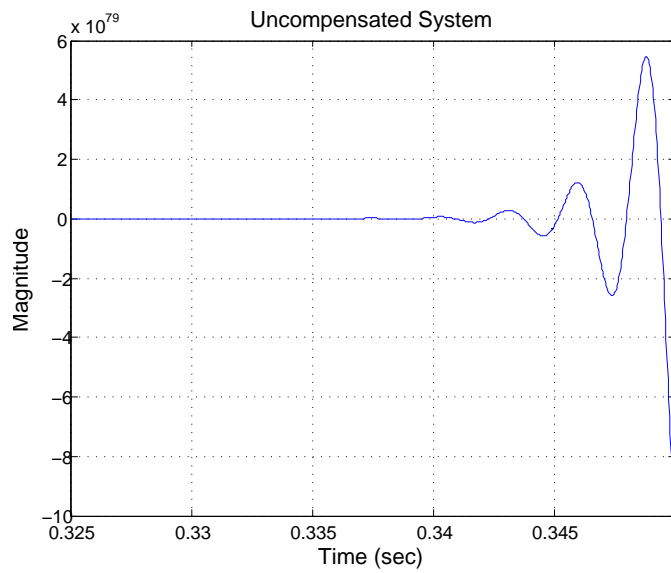


Figure 5.4: Matlab Analysis of Uncompensated System

### 5.1.3 Proportional Compensated System

In our first compensator design we will assess the efficacy of using a proportional compensator:

$$G_c(s) = k_p \quad (5.13)$$

$k_p$  simply represents a constant gain. Note that the effect of varying the value of  $k_p$  is to raise and lower the magnitude Bode plot while keeping the phase plot unaffected. So the value of  $k_p$  can be set to obtain a unity gain crossover frequency ( $f_c$ ) which results in an acceptable phase margin.

Generally the design procedure would require that asymptotic Bode plots for the now compensated loop gain be constructed, however, in the case of a proportional compensator, since the shape of the plots (magnitude and phase) are unchanged we need simply to replace any occurrences of the  $To$  with  $k_p To$  in Figure 5.2 and proceed accordingly.

As a general rule of thumb, to obtain an acceptable phase margin (generally  $45^\circ \leq PM \leq 60^\circ$ ) one usually sets the unity gain crossover frequency ( $f_c$ ) to occur in the segment of the asymptotic magnitude plot that has a slope of  $-20\text{dB/dec}$ . From the constructed magnitude plot we find

$$\frac{k_p T_o f_1}{f_c} = 1 \implies f_c = k_p T_o f_1 \quad (5.14)$$

We can substitute the above equation for  $f_c$  into the phase margin equation shown next to determine the required value of  $k_p$ .

$$\begin{aligned} PM &= 180 - \arctan\left(\frac{f_c}{f_1}\right) - \arctan\left(\frac{f_c}{f_2}\right) - \arctan\left(\frac{f_c}{f_3}\right) \\ &= 180 - \arctan(k_p T_o) - \arctan\left(\frac{k_p T_o f_1}{f_2}\right) - \arctan\left(\frac{k_p T_o f_1}{f_3}\right) \end{aligned} \quad (5.15)$$

With a desired value of phase margin of  $PM = 45^\circ$  equation (5.15) evaluates to  $k_p = 0.0311$  and  $f_c = 77.65$ .

Using the obtained value of  $k_p$  an evaluation of the compensated loop gain with the unapproximated transfer functions was performed by Matlab and is shown in Figure 5.5 where we see that the obtained phase margin is  $55^\circ$ . This value, due to the approximate nature of our design equations, turns out to more than required, but nonetheless acceptable. Had it not been so, one could simply iterate.

A step response simulation of the proportional compensated closed-loop system is shown in Figure 5.6. A summary of all the performance results (as furnished by the Matlab *stepinfo* function) are given in the table 5.2. There we see in particular that the overshoot, rise-time, settling-time and steady-state error values are 20%, 2.9 ms, 15.4 ms and  $-11\%$ , respectively.



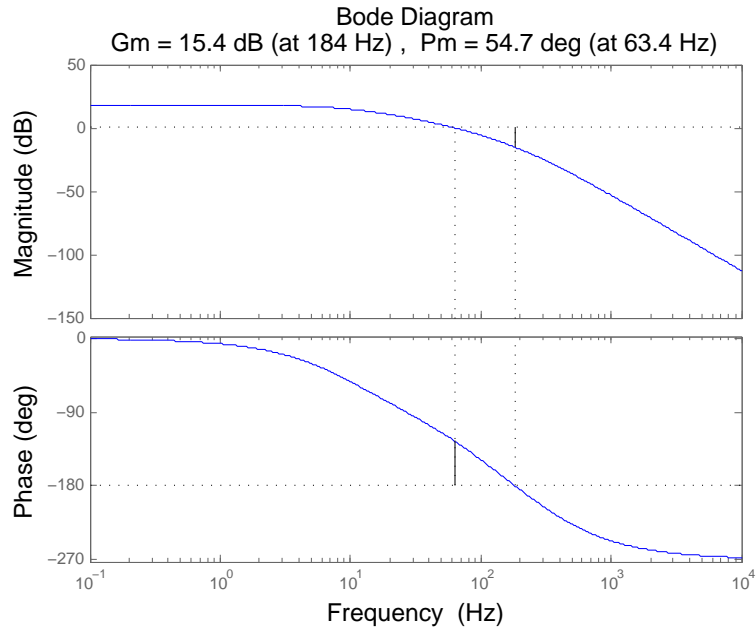


Figure 5.5: Matlab Proportional Compensated Loop Gain Bode Plot

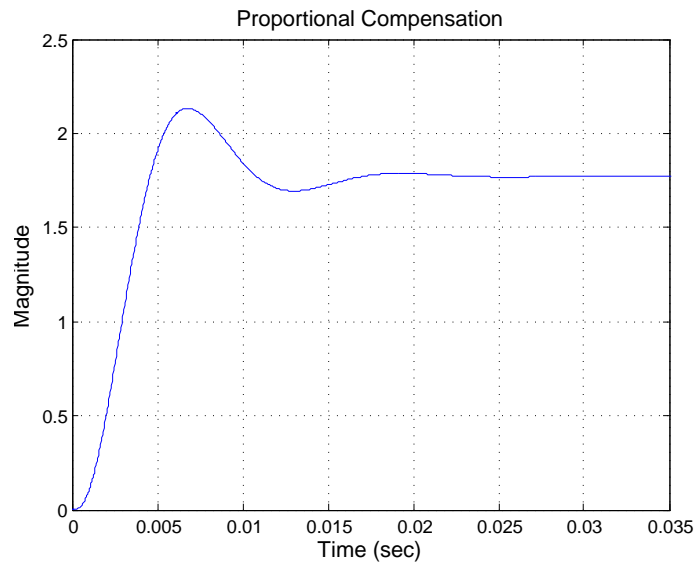


Figure 5.6: Step Response of Proportional Compensated Closed-Loop System

Table 5.2: Proportionally Compensated System Performance Features

<b>Proportional Compensation</b>	
Characteristics	Value
Overshoot	20 %
Rise time	2.9 ms
Settling time	15.4 ms
Steady-state error	-11 %
Bandwidth	63 Hz
Phase margin	55°
Gain margin	15 dB

#### 5.1.4 Dominant Pole Compensated System

The next form of frequency compensation to be discussed is the dominant pole compensation. As the name suggests a pole is inserted in the loop gain which dominates the dynamics of the loop. That is to say that the frequency response of the loop gain up to the unity gain crossover frequency is mainly determined by the dominant pole. To achieve this the pole needs to be placed at a frequency much lower (usually a decade or so) than the lowest pole or zero of the uncompensated loop gain. This requirement unfortunately reduces the loop bandwidth and subsequently the speed of response. However, compensator design is simplified and good stability margins may be easily obtained. If the pole is placed at zero frequency then this represents an integrator which may result in obtaining a zero steady state error characteristic.

In the design that follows, an integrator is employed so that the compensator transfer function is given by:

$$G_c(s) = \frac{\omega_I}{s}$$

where  $\omega_I = 2\pi \cdot f_I$  is an appropriately chosen design constant. It represents the frequency (in Hz) at which the compensator gain is at unity. Figure 5.7 shows the Bode plot asymptotes for the magnitude and phase of this compensator.

Design of the compensator now consists of selecting an appropriate compensator parameter,  $f_I$ .

Figure 5.8 shows the graphical construction of the phase asymptotes for the loop gain with the compensator. Note that because the plant's transfer function is third order (with no zeros), it features a phase shift of  $-270^\circ$  at high frequencies. Also, since the next higher frequency  $f_2 \geq 10 \cdot f_1$  the phase shift at  $f_1$  is  $-45^\circ$ . Furthermore, the compensator contributes its own  $-90^\circ$  phase shift and does so for all frequencies. Consequently, the total phase shift of the

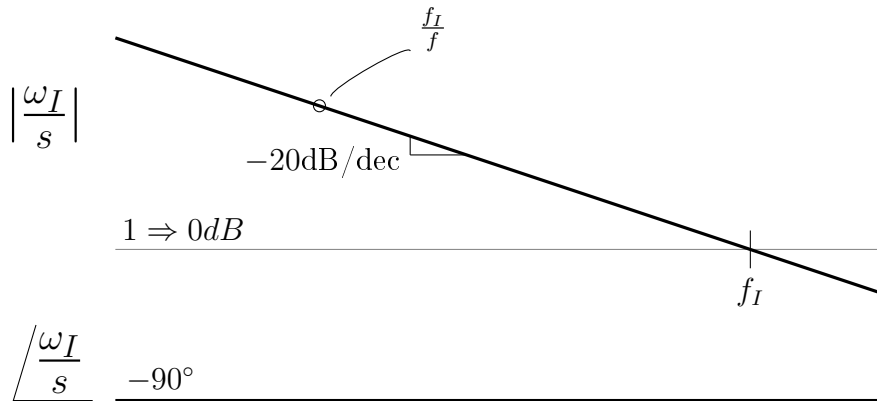


Figure 5.7: Bode Plot: Dominant Pole Compensator

compensated open loop transfer function at  $f_1$  is  $-135^\circ$ . Consequently, in this design the frequency  $f_1$  is chosen to be the unity gain crossover frequency of the overall system, so that we get a  $+45^\circ$  phase margin.

Figure 5.9 shows how the plant and compensator transfer functions combine to produce the magnitude response of the compensated loop. To achieve a phase margin that is  $+45^\circ$ , we require the magnitude at  $f_1$  to equal 1 (0dB), therefore  $f_c = f_1$ .

$$\frac{f_I T_o}{f_1} = 1$$

$$f_I = \frac{f_1}{T_o} = \frac{10}{250} = 0.04$$

The dominant pole compensator in this case is:

$$G_c(s) = \frac{\omega_I}{s} = \frac{2\pi \cdot 0.04}{s}$$

Figure 5.10 shows resulting gain and phase asymptotic construction of the Bode plot. We next run the full unapproximated compensated loop transfer function through the Matlab 'margin' command to verify the design results. Figure 5.11 shows the results obtained. In particular, a phase margin of  $PM = 45.9^\circ$  with a unity gain crossover frequency  $f_c = 7.84$  Hz was obtained, which compares favorably with the design values of  $PM = 45^\circ$  and  $f_c = f_1 = 10$  Hz.

Figure 5.12 shows the unit step response of the dominant pole compensated closed loop system. It is clearly seen that zero steady state response has been attained but not without going through significant overshoot first. The features of the closed loop system are summarized in the Table 5.3.

Table 5.3: Dominant Pole Compensated System Features

Dominant Pole Compensation	
Characteristics	Value
Peak amplitude	22.2% overshoot
Rise time	24.7 ms
Settling time	134 ms
Steady-state error	0 %
Bandwidth	7.84 Hz
Phase margin	45.9°
Gain margin	18.2 dB

### 5.1.5 Dominant Pole Compensated System with zero

The dominant pole compensator of the previous section, while stable and featuring zero steady state error performance, exhibits several undesirable characteristics which include a large overshoot and long settling time. In our effort to achieve stability, we have sacrificed the bandwidth of the system. Unfortunately, the lower the bandwidth, the slower the response. Thus, we need to increase the bandwidth to speed up the response. This can be done by adding a zero to the compensator that will cancel the lowest frequency pole of the plant. By doing this, the canceled pole no longer affects the phase lag seen at the gain crossover frequency. For our system a the 10 Hz pole is canceled so that the remaining poles at 100 and 300 Hz are now the poles which limit the bandwidth. Accordingly we will be able to extend the bandwidth by a decade.

The transfer function for the new compensator is given by

$$G_c(s) = \frac{\omega_I}{s} \left( 1 + \frac{s}{\omega_z} \right)$$

where

$$\omega_z = \omega_1$$

The compensated loop gain  $T(s) = kG_c(s)G(s)$  with this compensator now becomes:

$$T(s) = \frac{T_o\omega_I}{s \left( 1 + \frac{s}{\omega_2} \right) \left( 1 + \frac{s}{\omega_3} \right)}$$

The construction of the compensated magnitude plot is shown in Figure 5.13 where the uncompensated plant and compensator magnitude asymptotes are combined. The final magnitude and phase plots using the asymptotic construction procedure is shown in Figure 5.14. Using the low frequency magnitude asymptote we see that at the unity gain crossover frequency  $f_c$

$$\frac{T_o f_I}{f_c} = 1$$

$$\implies f_c = T_o f_I \quad (5.16)$$

The general expression for  $\phi_f$ , the phase response at an arbitrary frequency  $f$  is given by:

$$\phi_f = -90 - \arctan\left(\frac{f}{f_2}\right) - \arctan\left(\frac{f}{f_3}\right) \quad (5.17)$$

Consequently the phase margin is given by:

$$\begin{aligned} PM &= 180 + \phi_{f_c} \\ &= 90 - \arctan\left(\frac{f_c}{f_2}\right) - \arctan\left(\frac{f_c}{f_3}\right) \\ &= 90 - \arctan\left(\frac{T_o f_I}{f_2}\right) - \arctan\left(\frac{T_o f_I}{f_3}\right) \end{aligned} \quad (5.18)$$

With a design value of  $PM = 45^\circ$  the required value of  $f_I$  is solved using Equation (5.18) to yield  $f_I = 0.258$  so that  $\omega_I = 2\pi f_I = 1.623$ . This results, using (5.16), in a gain crossover frequency,  $f_c = 64.58$  Hz.

To verify the design procedure the unapproximated loop gain transfer function is run through the Matlab 'margin' command which produces the plot shown in Figure 5.15. The obtained phase margin and bandwidth are seen to be  $PM = 51^\circ$  and  $f_c = 56$  Hz. The bandwidth has been greatly improved.

The unit step response of the closed loop system is shown in Figure 5.16. A summary of the performance using this compensator is shown in Table 5.4.

Table 5.4: Dominant Pole with Zero Compensated System Features

<b>Dominant Pole with Zero Compensation</b>	
Characteristics	Value
Overshoot	22.2%
Rise time	3.05 ms
Settling time	16.7 ms
Steady-state error	0 %
Bandwidth	62.3 Hz
Phase margin	46.3 degree
Gain margin	14.5 dB

### 5.1.6 Dominant Pole Compensated System with zero, improved phase margin

The addition of the zero to dominant pole compensation has allowed us to increase the speed of response by extending bandwidth. This is manifested in the time domain response by a substantial decrease in the rise and settling times. However, the overshoot, while slightly improved, may be seen as overly large. Nevertheless, it too may be reduced by appreciating the fact that overshoot and phase margin are inversely related. So we will increase the phase margin to reduce the overshoot.

Using the same compensator as in the previous section, we will now redesign it to achieve a phase margin of  $60^\circ$ . The zero frequency is left unchanged, so that  $f_z = f_1$ , but a new value of  $f_I$  will be determined to achieve the desired  $60^\circ$  phase margin. Solving equation (5.18), now with  $PM = 60$  we see the required value of  $f_I = 0.1636$  so that  $\omega_I = 2\pi f_I = 1.0276$ .

To verify the design the Bode plot of the loop gain was produced using the 'margin' command in Matlab. This plot is shown in figure 5.17. We see that a phase margin of  $62^\circ$  was achieved with a bandwidth of 37.9 Hz. Of course the extension in phase margin is necessarily accompanied by a reduction in bandwidth.

The step response of the closed loop system is shown in Figure 5.18. The resulting performance characteristics are tabulated in the Table 5.5. There we see that the overshoot has been reduced to 4.46%. Recall that a phase margin of  $45^\circ$  had previously resulted in a 17% overshoot. This reduction in overshoot has been attained at the expense of an increase in rise time. However, the settling time has been slightly reduced.

Table 5.5: Dominant Pole with Zero (Improved PM) Compensated System Features

<b>Dominant Pole with Zero Compensation (Improved Margin)</b>	
Characteristics	Value
Overshoot	4.46 %
Rise time	5.81 ms
Settling time	16.5 ms
Steady-state error	0
Bandwidth	35.1 Hz
Phase margin	64 degree
Gain margin	20.6 dB

### 5.1.7 Lead Compensated System

Next we consider the case of using a lead compensator, the transfer function of which is given by:

$$G_c(s) = G_{c_o} \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{\omega_p}}, \quad \omega_z < \omega_p \quad (5.19)$$

The design of this compensator requires the appropriate determination of the three variables,  $G_{c_o}$ ,  $f_z$  and  $f_p$ , the low frequency gain, zero frequency and pole frequency, respectively. Note in particular that the zero frequency is required to be at a lower value than the pole frequency. This constraint exists so that the phase response, first starting at zero at low frequencies, becomes positive, i.e. leads, as the frequency increases before returning to zero at high frequencies. In essence, the lead compensator provides a phase boost that is adjustable based on the pole and zero frequencies. The maximum phase boost  $\phi_{max}$  possible is  $\phi_{max} = 90^\circ$  and occurs at a frequency  $f_{\phi_{max}}$  which is the geometric mean of the zero and pole frequencies of the compensator. The geometric mean of two numbers represents the midpoint between these numbers when represented on a logarithmic scale.

$$f_{\phi_{max}} = \sqrt{f_z f_p} \quad (5.20)$$

The compensator also provides a gain boost at higher frequencies so that with proper design it can both extend loop bandwidth while increasing the phase margin. Proper design of the compensator requires that the frequency of maximum boost afforded by this compensator is set to the unity gain crossover frequency,  $f_c$ . The asymptotic Bode plot of the lead compensator is shown in Figure 5.19.

When the compensator is placed in the loop, the loop gain of the system now becomes:

$$T(s) = \frac{T_o G_{c_o} \left(1 + \frac{s}{\omega_z}\right)}{\left(1 + \frac{s}{\omega_p}\right) \left(1 + \frac{s}{\omega_1}\right) \left(1 + \frac{s}{\omega_2}\right) \left(1 + \frac{s}{\omega_3}\right)}$$

The construction of the asymptotic magnitude response from the constituent parts is shown in Figure 5.20. Both the resulting magnitude and phase asymptotic responses of the loop gain are shown in Figure 5.21. As for previous compensators, the annotations on these plots are used in the design procedure. The exact expression for the phase  $\phi$  is given by:

$$\begin{aligned}\phi_f = \arctan\left(\frac{f}{f_z}\right) - \arctan\left(\frac{f}{f_p}\right) - \arctan\left(\frac{f}{f_1}\right) \\ - \arctan\left(\frac{f}{f_2}\right) - \arctan\left(\frac{f}{f_3}\right)\end{aligned}\quad (5.21)$$

Consequently the phase margin is given by:

$$\begin{aligned}PM = 180 + \phi_{f_c} \\ = 180 + \arctan\left(\frac{f_c}{f_z}\right) - \arctan\left(\frac{f_c}{f_p}\right) - \arctan\left(\frac{f_c}{f_1}\right) \\ - \arctan\left(\frac{f_c}{f_2}\right) - \arctan\left(\frac{f_c}{f_3}\right)\end{aligned}\quad (5.22)$$

As mentioned previously, we will set the unity gain crossover frequency  $f_c$  to the maximum phase boost frequency  $f_{\phi_{max}}$  so that using equation (5.20) we have:

$$\begin{aligned}f_c = f_{\phi_{max}} \\ = \sqrt{f_z f_p}\end{aligned}\quad (5.23)$$

In order to minimize the effect on the phase margin of the phase lag due to the compensator pole we will set this pole frequency an order of magnitude above the crossover frequency:

$$f_p = 10f_c \quad (5.24)$$

Equation (5.24) together with equation (5.23) results in a relationship between the zero and pole frequencies of the compensator:

$$f_p = 100f_z \quad (5.25)$$

We can use the zero frequency of the compensator to cancel the pole frequency  $f_2$  of the plant, so that  $f_z = f_2$ . This together with equations (5.25) and (5.22) may be used to determine the unity gain crossover frequency,  $f_c$ , for a given desired phase margin. For a margin of  $60^\circ$  we find  $f_c = 187$  Hz.

From the magnitude asymptote we see that

$$\frac{T_o G_{c_o} f_1}{f_c} = 1 \quad (5.26)$$

so that for a given  $f_c$  we can solve for the required compensator low frequency gain  $G_{c_o}$ :

$$G_{c_o} = \frac{f_c}{T_o f_1} \quad (5.27)$$

For our design we obtain a value of  $G_{c_o} = 0.0749$ .



To verify our design we produce the plot using the Matlab 'margin' command on the unapproximated transfer functions of the compensated loop. This is shown in figure 5.22. There we find that the obtained phase margin is  $PM = 63.9^\circ$  with  $f_c = 164$  Hz, which validates our design procedure.

The unit step response of the closed loop system with this compensation is shown in figure 5.23. The features of the step response are presented in the Table 5.7. We see, due to the extended bandwidth, that the speed of response, represented by the rise and settling times is quite good. Percentage overshoot is also relatively low due to the  $60^\circ$  phase margin employed. However, as there is no longer an integrator in the forward path, the steady state error is now non-zero. This will be remedied in the next and final compensator design.

Table 5.6: Lead Compensation

<b>Lead Compensation</b>	
Feature	Value
Overshoot	4.23 %
Rise time	2.25 ms
Settling time	6.29 ms
Steady-state error	10%
Bandwidth	83.8 Hz
Phase margin	$71.2^\circ$
Gain margin	19.3 dB

### 5.1.8 Lead Compensated System with integrator and zero

In this final compensator design, we will augment the lead compensator of the previous section with an integrator and zero. The integrator is added to provide the closed loop system with a zero steady state error performance. This is contrasted with the approach previously discussed where the integrator was primarily used to as the dominant pole. In the present case by canceling lower frequency poles, we're able to extend the unity gain bandwidth of the loop gain to obtain quick closed loop response. The zero is added to leave the high frequency magnitude and phase response unchanged from the lead compensator case.

The transfer function of the compensator considered here is given by:

$$G_c(s) = \frac{\omega_I \left(1 + \frac{s}{\omega_{z_1}}\right) \left(1 + \frac{s}{\omega_{z_2}}\right)}{s \left(1 + \frac{s}{\omega_p}\right)} \quad (5.28)$$

The parameters  $\omega_{z_2}$  and  $\omega_p$  correspond to  $\omega_z$  and  $\omega_p$  of the lead compensator design, which leaves  $\omega_I$  and  $\omega_{z_1}$  to be determined.  $\omega_{z_1}$  can be simply set to  $\omega_1$ .

The low frequency gain of the lead compensator of the previous section was denoted  $G_{c_o}$ . This was the value of the loop magnitude at  $f_1$  (in particular, and below this frequency, in general). To maintain this value of gain at  $f_1$  we will adjust  $\omega_I$  to achieve this. The low frequency magnitude asymptote is given by  $\frac{f_I}{f}$  so that at  $f_1$  we have

$$\frac{f_I}{f_1} = G_{c_o} \implies f_I = G_{c_o} f_1 \quad (5.29)$$

This completes the design of this compensator. To verify our design we produce the plot using the Matlab 'margin' command on the unapproximated transfer functions of the compensated loop. This is shown in figure 5.24. There we find that the obtained phase margin is  $PM = 60^\circ$  with  $f_c = 164$  Hz, which very closely agrees with the results obtained for the lead compensator.

Next, we examine the step response of the closed loop system which is shown in 5.25. It is clear that now we have zero steady state error. Further performance features are given in the table below.

Table 5.7: Lead Compensation with integrator and zero

<b>Lead Compensation with integrator and zero</b>	
Feature	Value
Overshoot	4.23 %
Rise time	2.25 ms
Settling time	6.29 ms
Steady-state error	10%
Bandwidth	83.8 Hz
Phase margin	71.2°
Gain margin	19.3 dB

### 5.1.9 Summary

The following table shows the summary of all of the results.

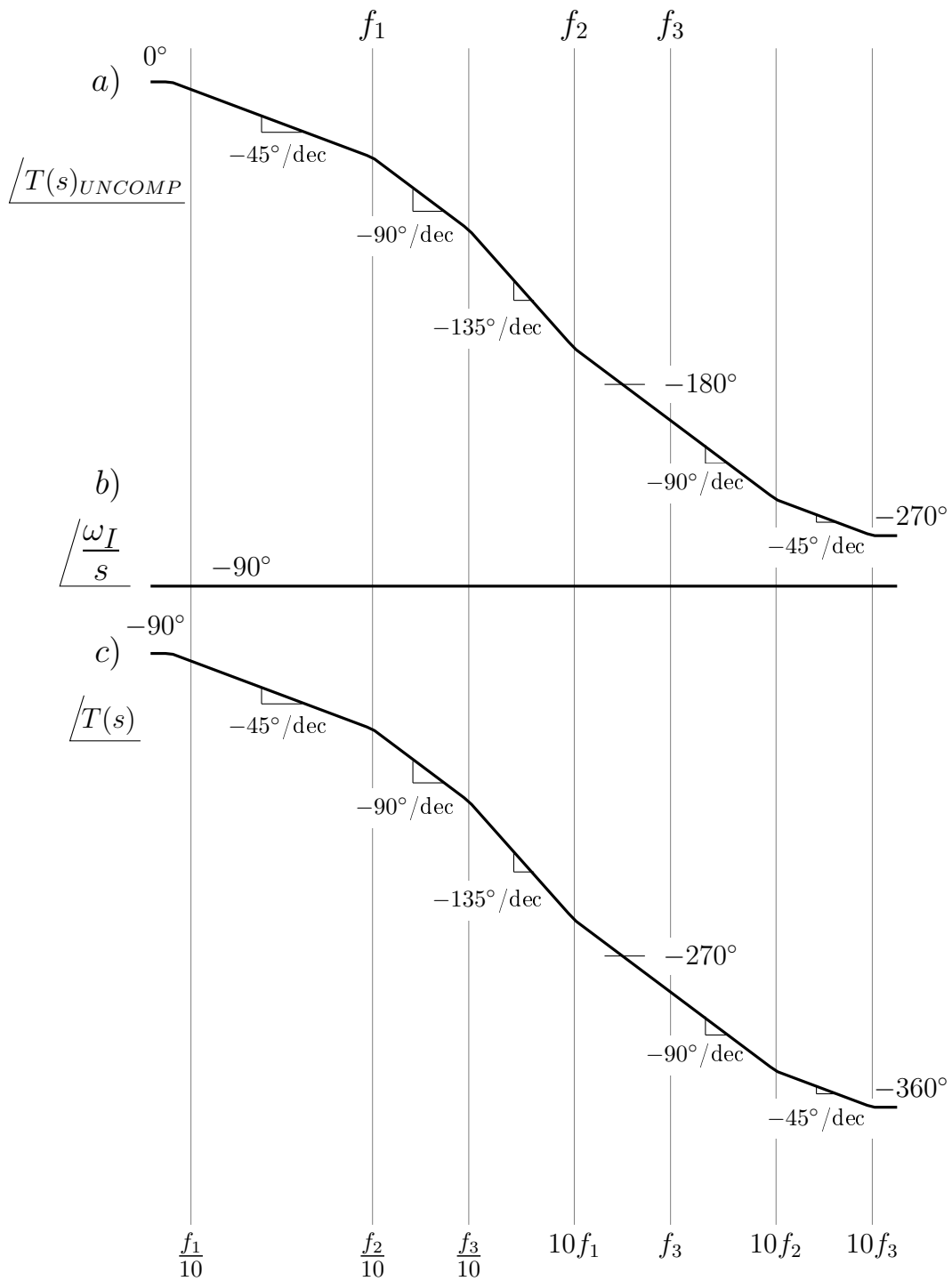


Figure 5.8: Dominant Pole: Phase Construction

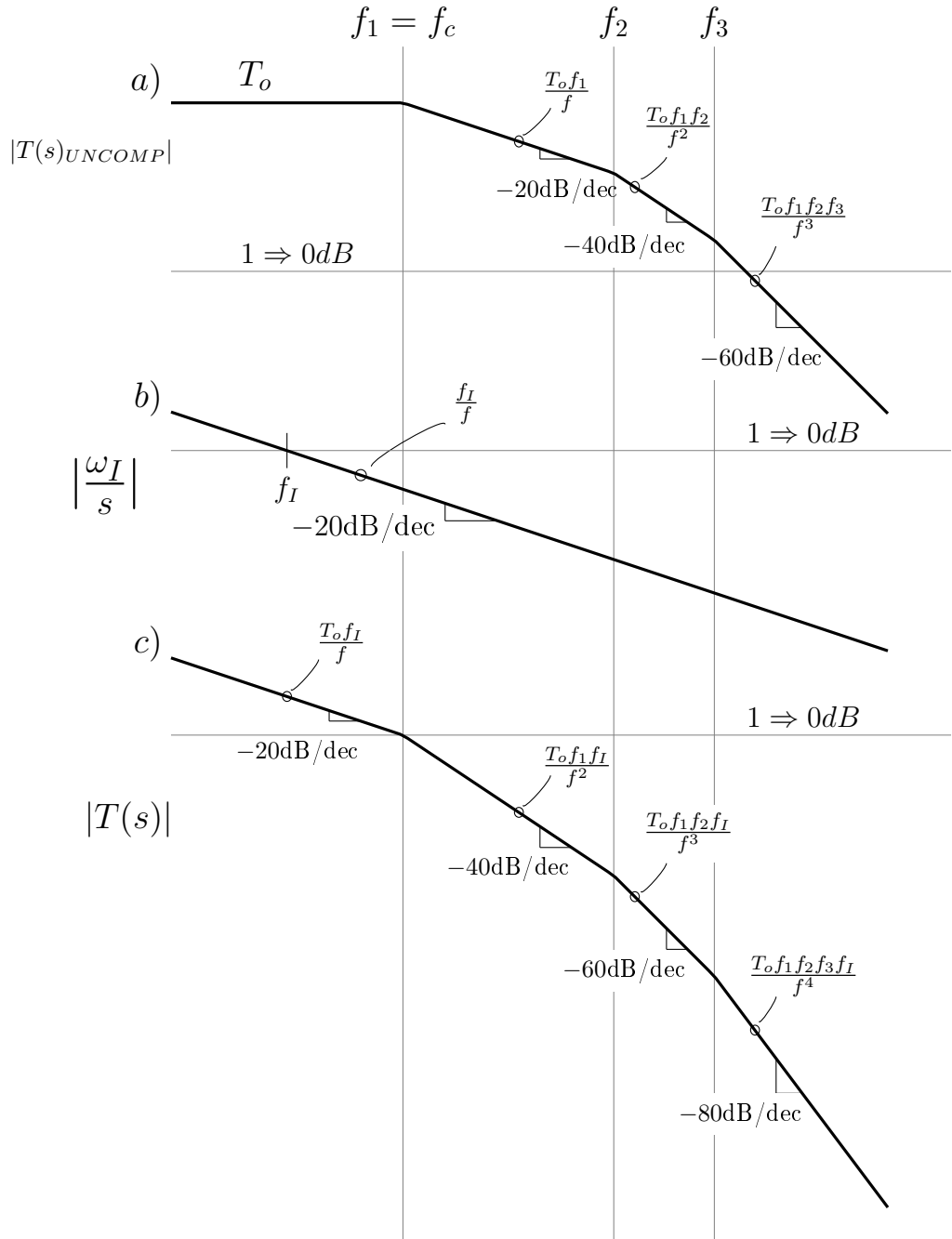


Figure 5.9: Dominant Pole: Magnitude Construction

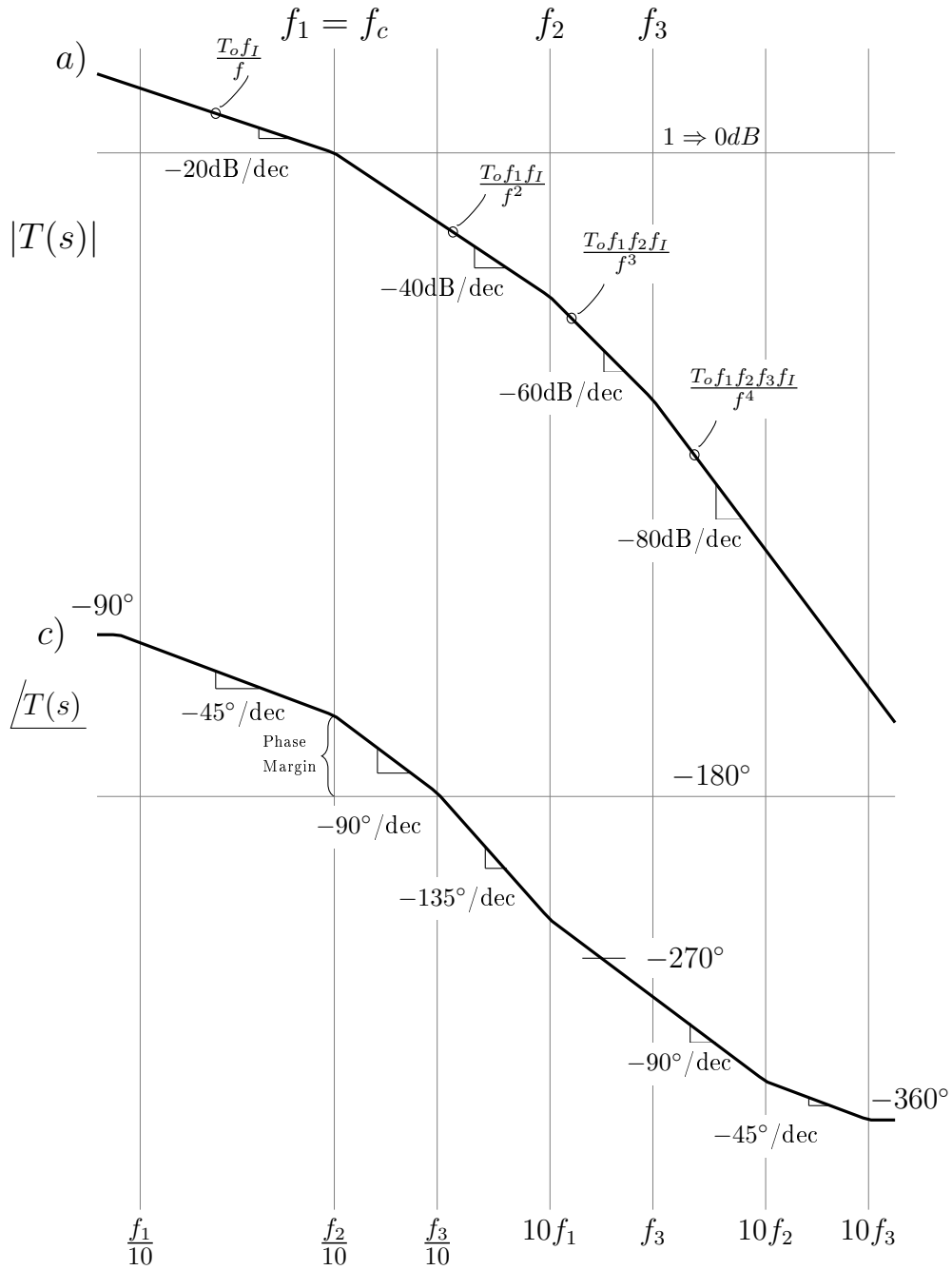


Figure 5.10: Dominant Pole Compensated System

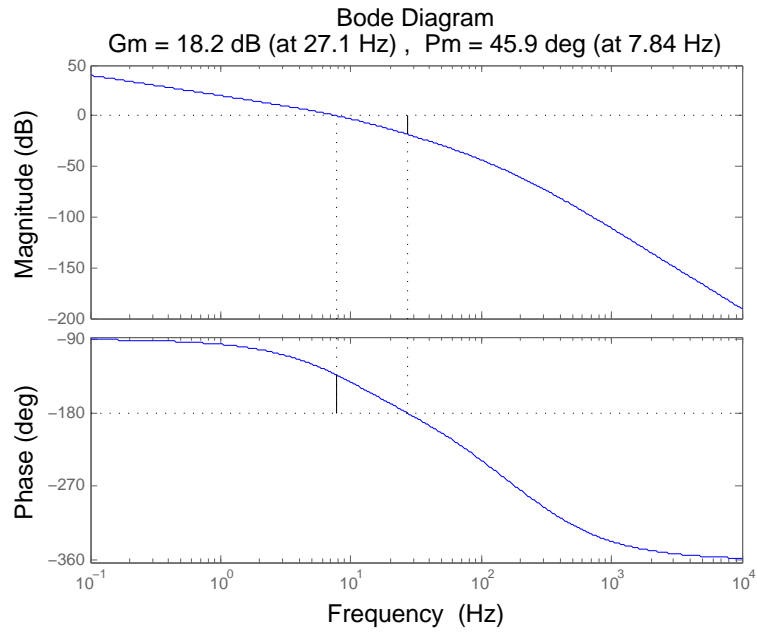


Figure 5.11: Matlab Analysis of Dominant Pole Compensated System

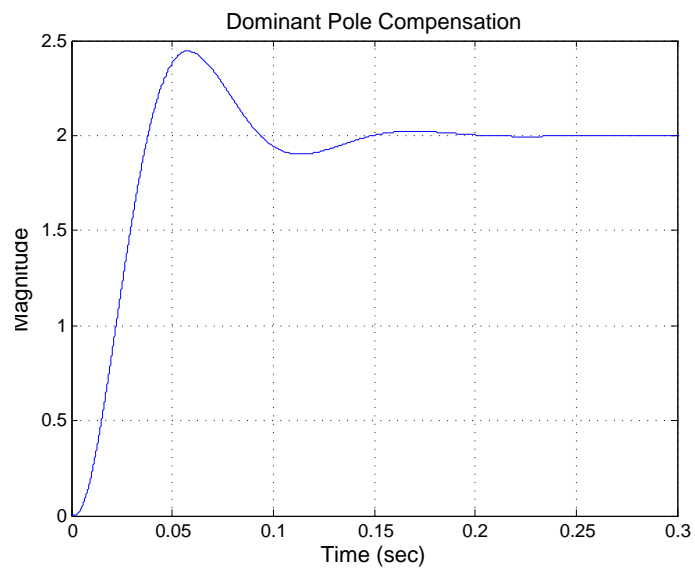


Figure 5.12: Step Response of the Dominant Pole Compensated System

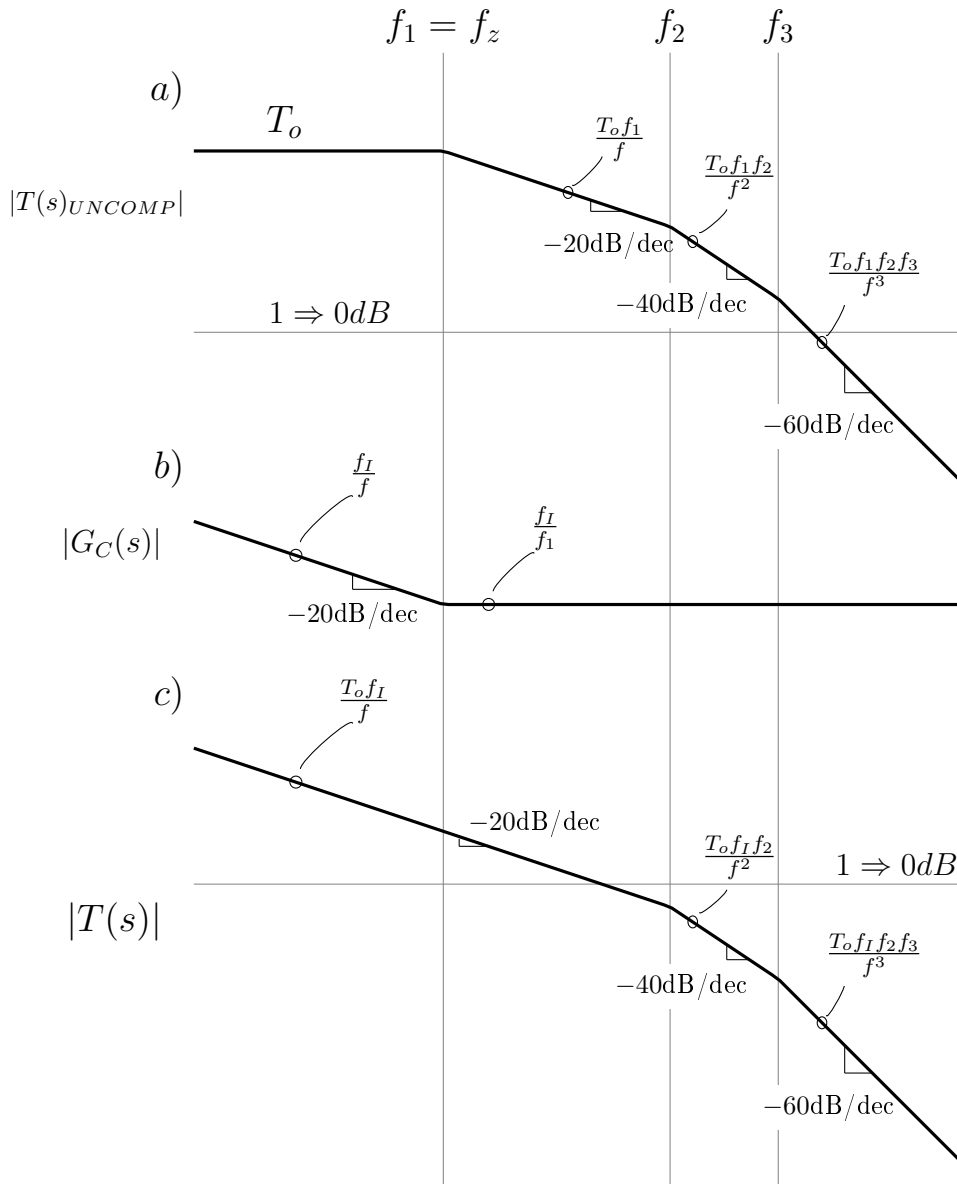


Figure 5.13: Dominant Pole with Zero Magnitude Construction

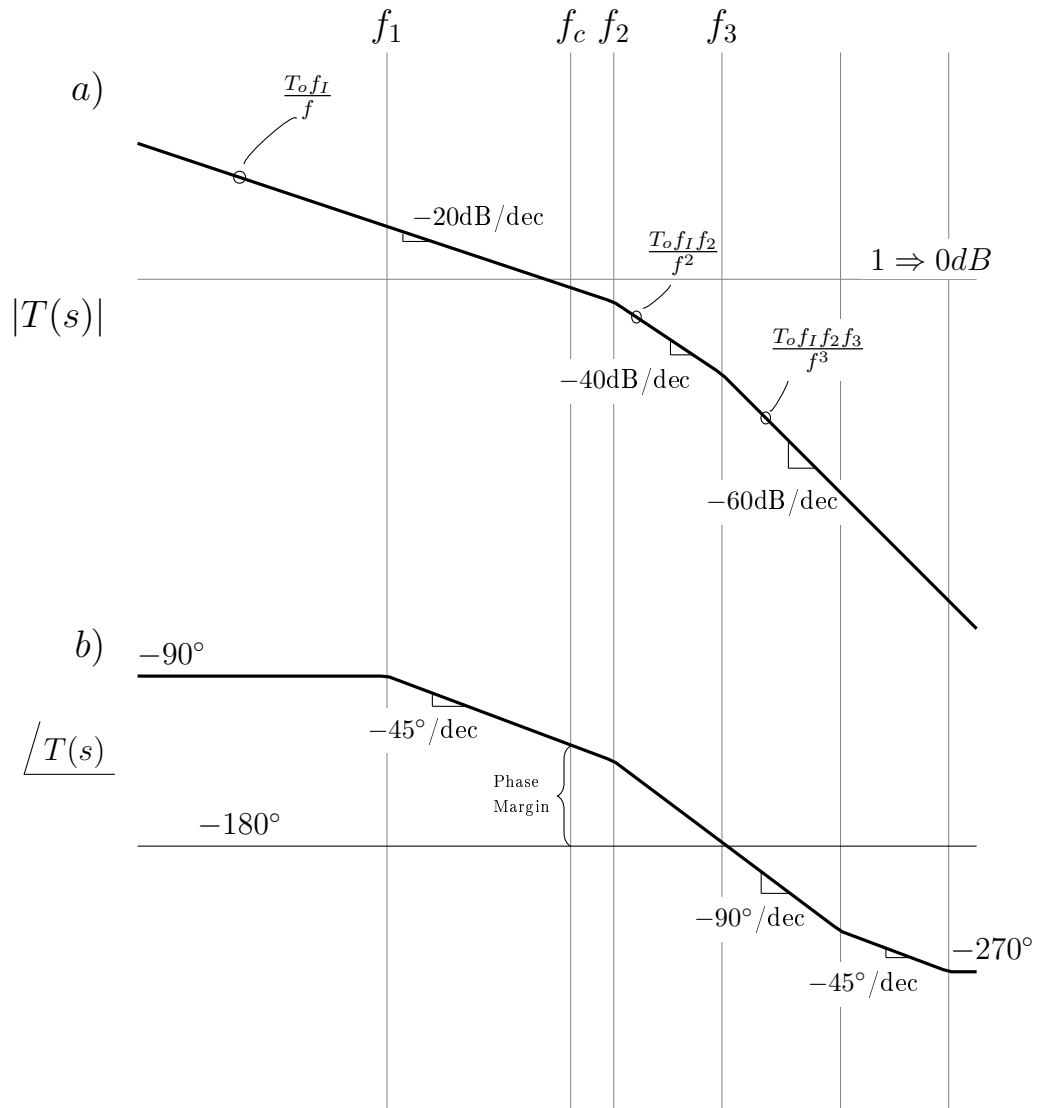


Figure 5.14: Dominant Pole with Zero Compensated System



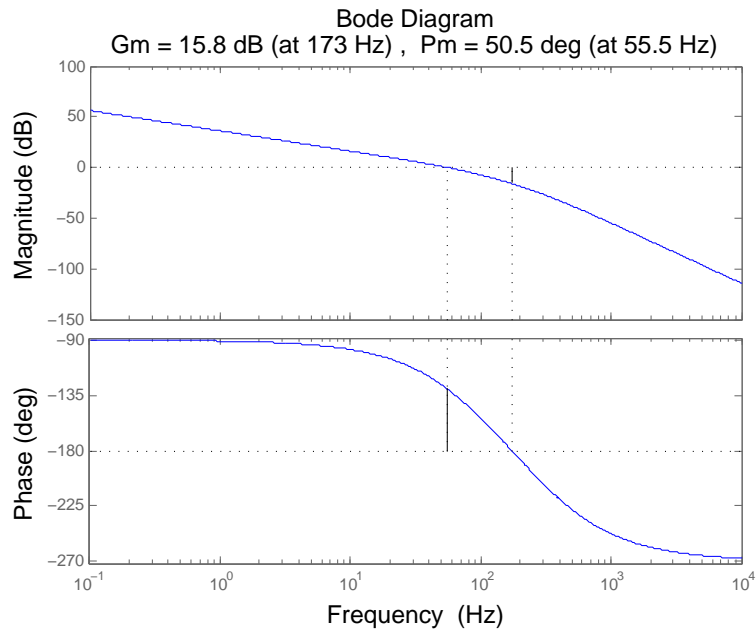


Figure 5.15: Loop Gain and Phase Response of the Dominant Pole Compensated System with Zero

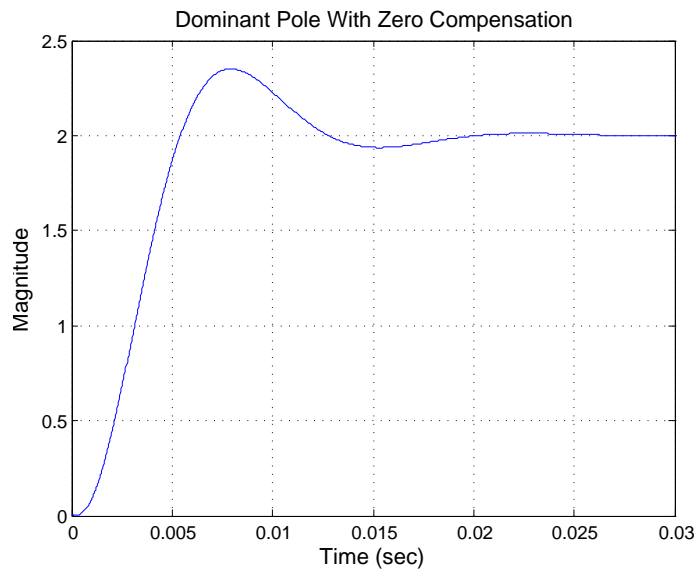


Figure 5.16: Step Response of the Dominant Pole with Zero Compensated System

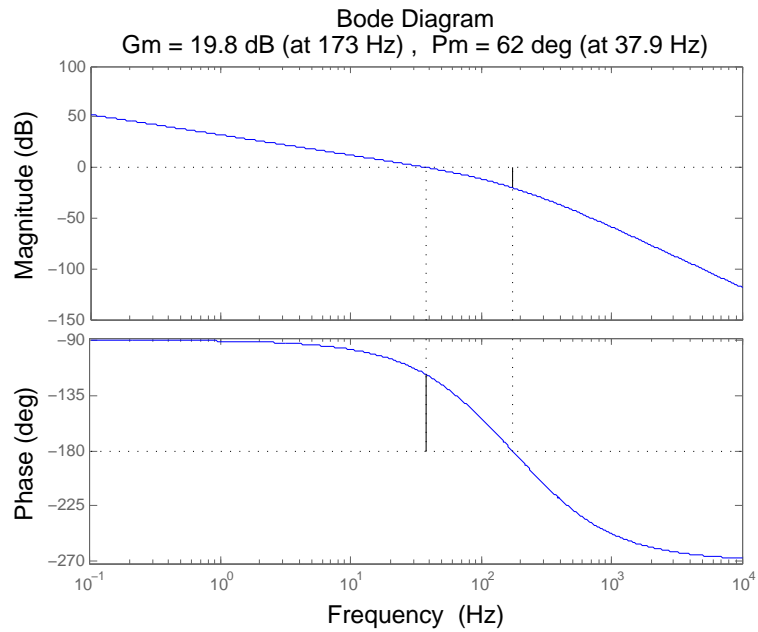


Figure 5.17: Matlab Analysis of Dominant Pole Compensated System with Zero (Improved Margin)

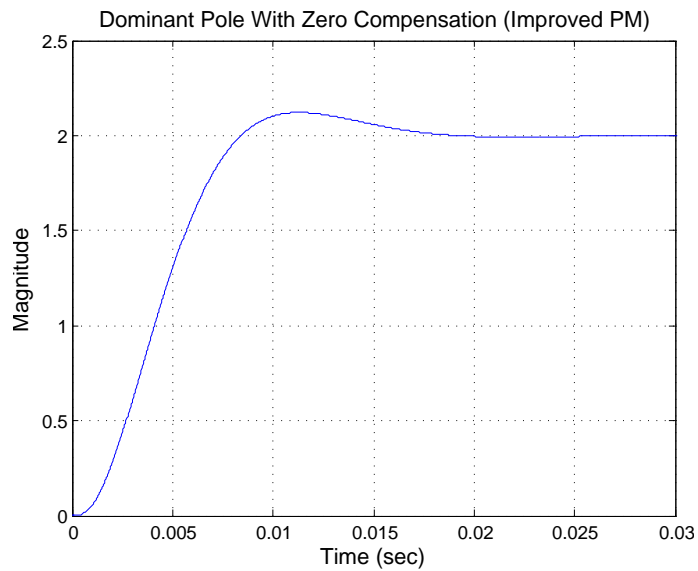


Figure 5.18: Step Response of the Dominant Pole with Zero Compensated System (Improved Margin)

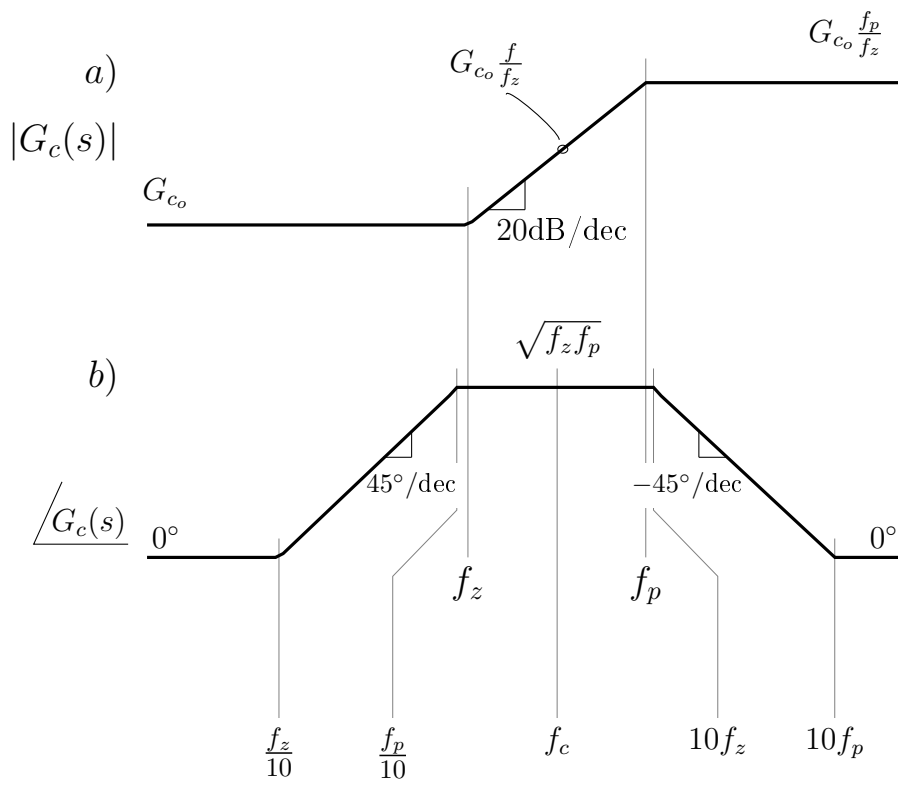


Figure 5.19: Bode Diagram: Lead Compensator

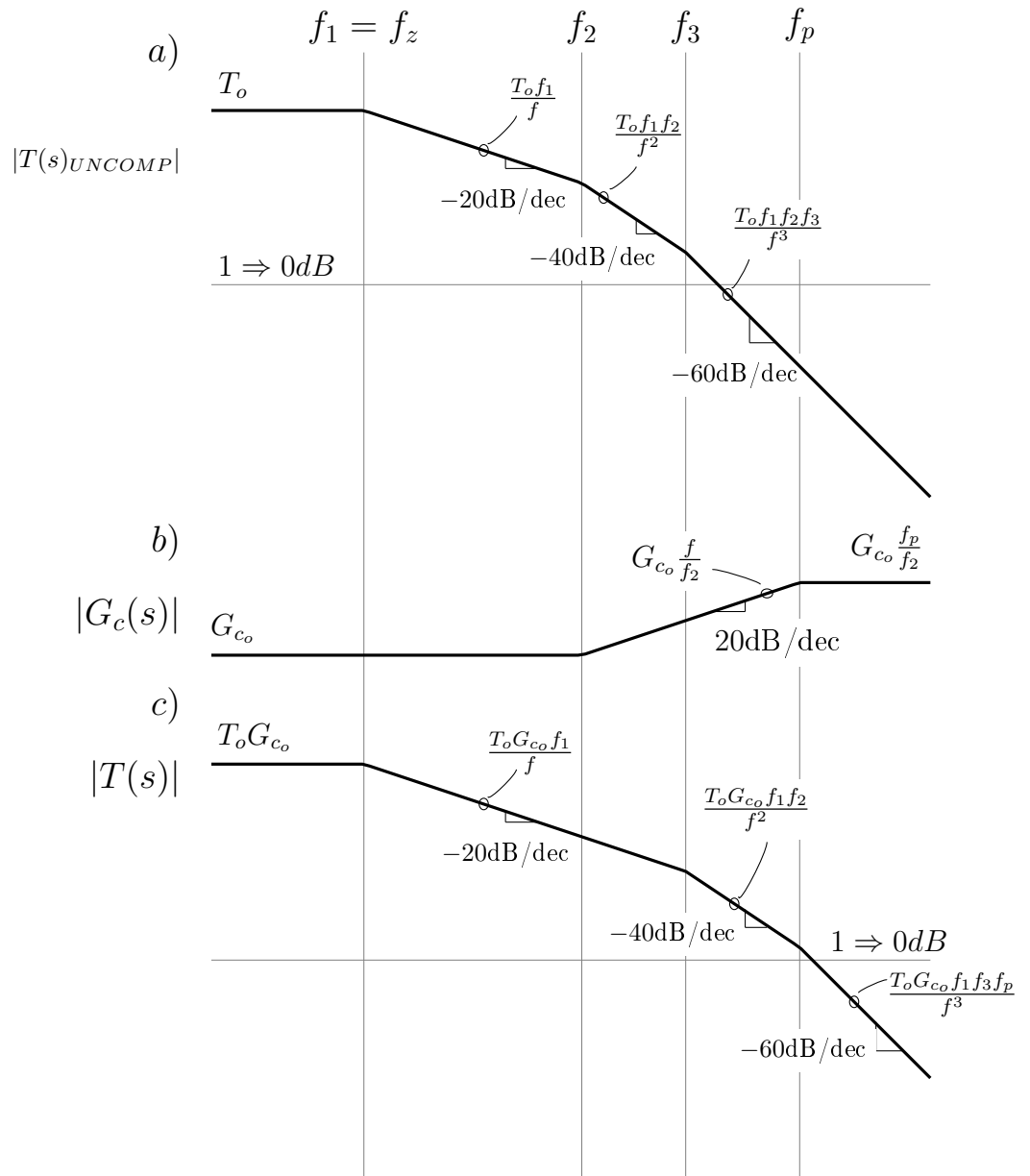


Figure 5.20: Lead Compensation Magnitude Construction

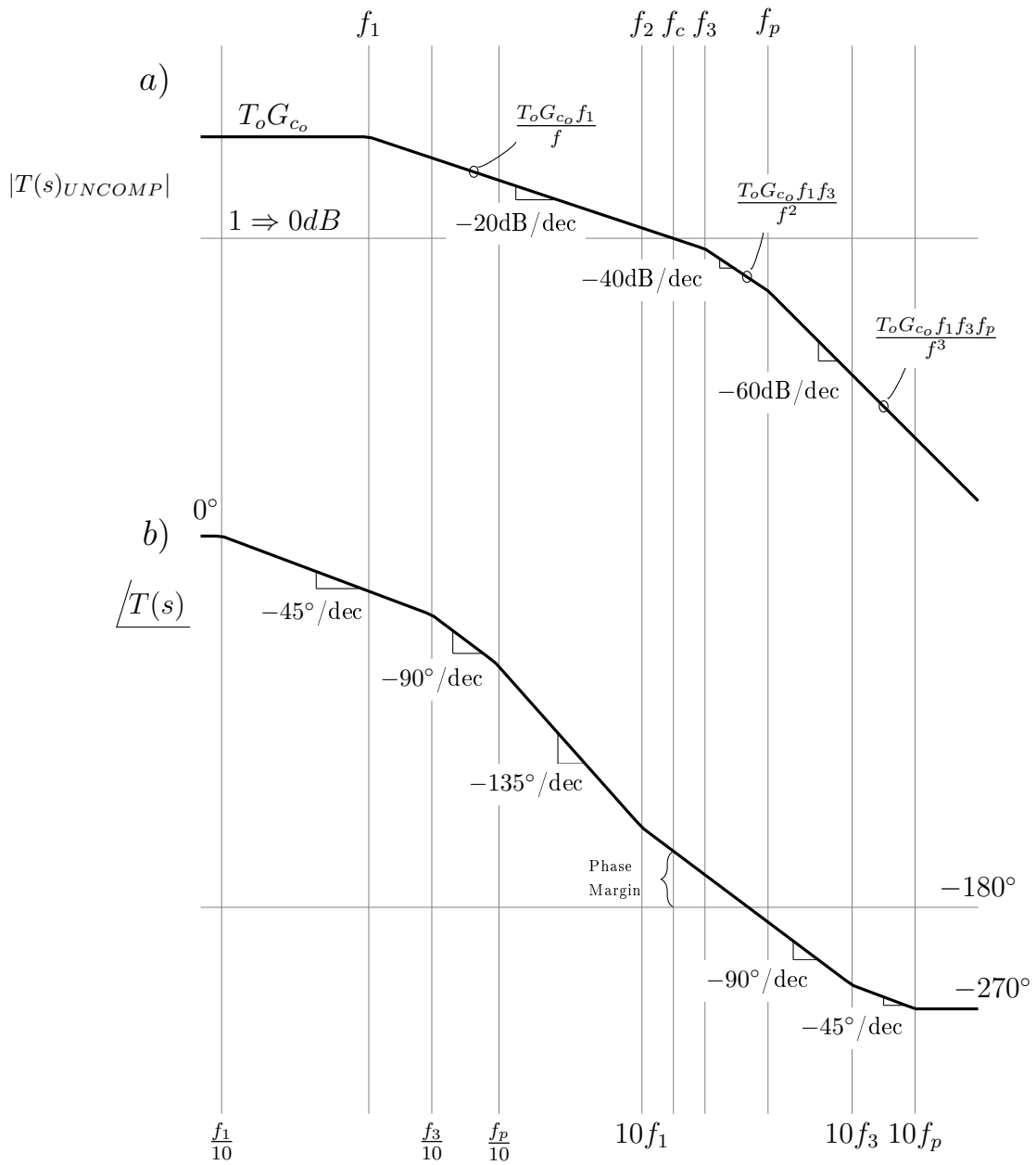


Figure 5.21: Lead Compensated System

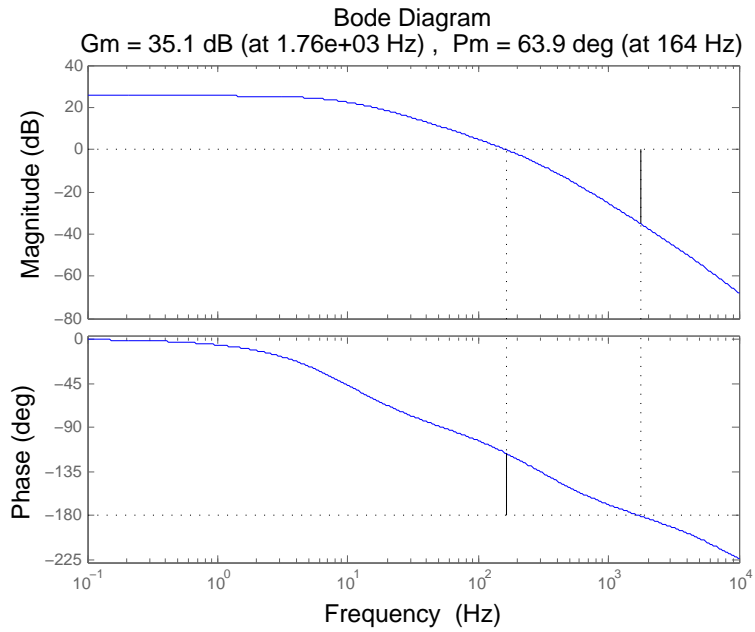


Figure 5.22: Matlab Analysis of Lead Compensated System

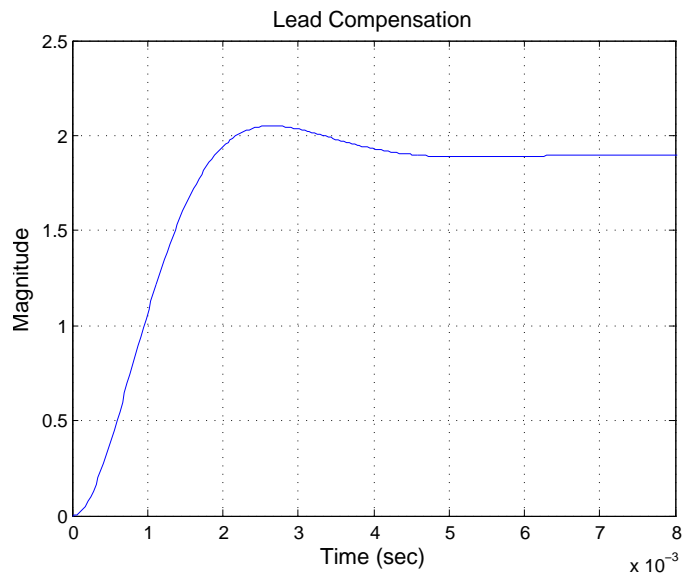


Figure 5.23: Step Response of the Lead Compensated System

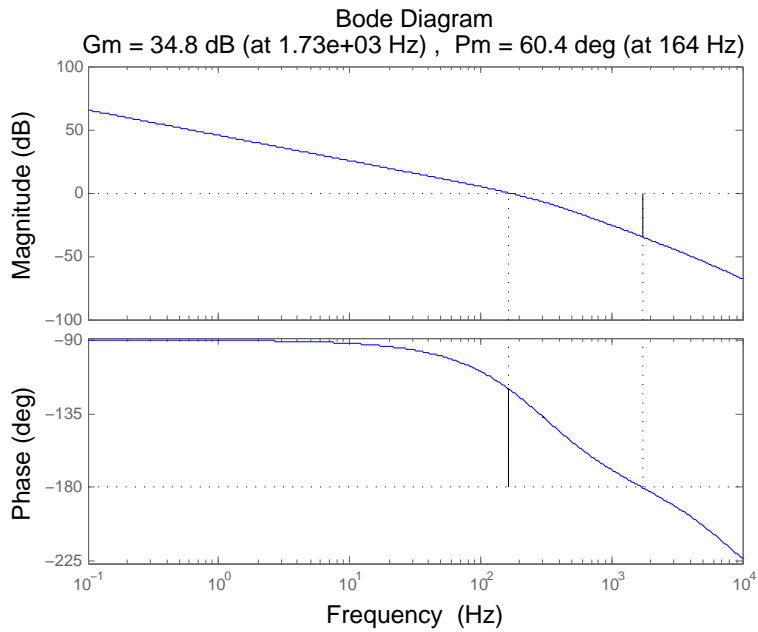


Figure 5.24: Matlab Analysis of Lead Compensated System with integrator and zero

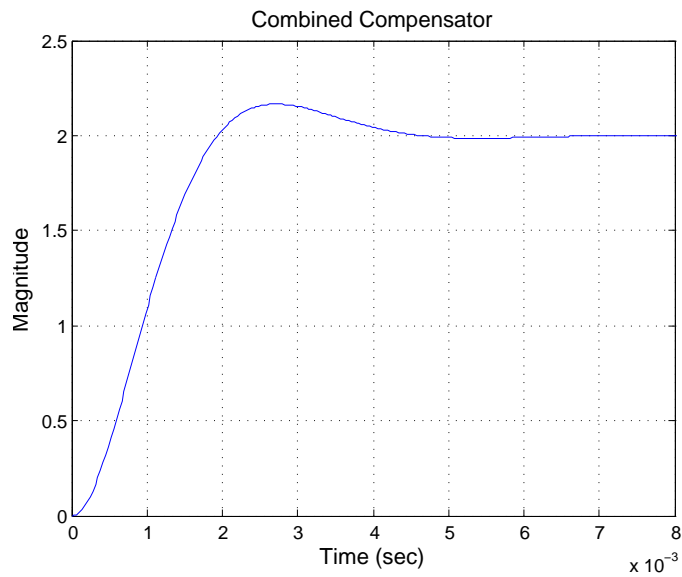


Figure 5.25: Step Response of the Lead Compensated System with integrator and zero

Table 5.8: Summary of Compensators

$$G(s) = \frac{G_o}{\left(1 + \frac{s}{\omega_1}\right)\left(1 + \frac{s}{\omega_2}\right)\left(1 + \frac{s}{\omega_3}\right)}$$

$$H(s) = k$$

where  $G_o = 500$ ,  $\omega_1 = 2\pi(10)$ ,  $\omega_2 = 2\pi(100)$ ,  $\omega_3 = 2\pi(300)$ , and  $k = 0.5$ .

$G_c(s)$	$G_c(s)$ parameters	$f_c$ (Hz)	$\phi_{PM}$ (deg)	GM (dB)	OS (%)	$t_r$ (ms)	$t_s$ (ms)	ERR (%)
1 (uncompensated)	none	385	-36	-15	na	na	na	$\infty$
$k_p$	$k_p = 0.03$	63	55	15	20	2.9	15.4	-11
$\frac{\omega_I}{s}$	$\omega_I = 0.25$	7.8	46	18	22	25	137	0
$\frac{\omega_I}{s} \left(1 + \frac{s}{\omega_z}\right)$	$\omega_I = 1.62$ $\omega_z = 2\pi(10)$	56	51	16	17	34	18	0
$\frac{\omega_I}{s} \left(1 + \frac{s}{\omega_z}\right)$	$\omega_I = 1.03$ $\omega_z = 2\pi(10)$	38	62	20	6	5.3	16	0
$G_{c_o} \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{\omega_p}}$	$G_{c_o} = 0.075$ $\omega_z = 2\pi(100)$ $\omega_p = 2\pi(10,000)$	164	64	35	8.1	1.3	3.9	-5
$\frac{\omega_I \left(1 + \frac{s}{\omega_{z1}}\right) \left(1 + \frac{s}{\omega_{z2}}\right)}{s \left(1 + \frac{s}{\omega_p}\right)}$	$\omega_I = 4.71$ $\omega_{z1} = 2\pi(10)$ $\omega_{z2} = 2\pi(100)$ $\omega_p = 2\pi(10,000)$	164	60	35	8.3	1.3	4	0



## 5.1.10 MATLAB Code

```
1 function compensators
2
3 clear all;
4 close all;
5
6 t = linspace(0, 0.35, 10000);
7 f = logspace(-1, 4, 1000);
8 w = 2*pi*f;
9 s = tf('s');
10
11 Go = 500;
12 f1 = 10;
13 f2 = 100;
14 f3 = 300;
15
16 w1 = 2*pi*f1;
17 w2 = 2*pi*f2;
18 w3 = 2*pi*f3;
19 k = 0.5;
20 To = Go*k;
21 yf = 2;
22
23 %=====
24 %
25 % UNCOMPENSATED
26 %=====
27
28 G = Go/((1+s/w1)*(1+s/w2)*(1+s/w3));
29
30 titl = 'Uncompensated System';
31 Gc = 1;
32
33 xlmt = [0.325 0.35];
34 disp(titl)
35 analysis(Gc, G, k, w, titl, t, xlmt, yf)
36
37 %      RiseTime: 0.0031
38 %      SettlingTime: 0.3500
39 %      SettlingMin: -9.9007e+79
40 %      SettlingMax: -8.9587e+79
41 %      Overshoot: 0
42 %      Undershoot: 55.0478
43 %      Peak: 9.9007e+79
44 %      PeakTime: 0.3500
45
46 %=====
47 %=====
48 % Proportional Compensation
49 %=====
50
51 fn = @(kp) 135 - atand(To*f1*kp/f1) - atand(To*f1*kp/f2) ...
52         - atand(To*f1*kp/f3);
53
54 kp = fzero(fn, 0)      % kp = 0.0311
```

```

55 fc = To*f1*kp          % fc = 77.6459
56
57 titl = 'Proportional Compensation';
58 Gc = kp;
59
60 xlmt = [0 0.035];
61 disp(titl)
62 analysis(Gc, G, k, w, titl, t, xlmt, yf)
63
64 %           RiseTime: 0.0029
65 %       SettlingTime: 0.0154
66 %       SettlingMin: 1.6011
67 %       SettlingMax: 2.1341
68 %       Overshoot: 20.4502
69 %       Undershoot: 0
70 %           Peak: 2.1341
71 %       PeakTime: 0.0068
72
73 %=====
74 %=====
75 % Dominant Pole Compensation
76 %=====
77
78 titl = 'Dominant Pole Compensation';
79 wI = w1/To           % 0.2513
80 Gc = wI/s;
81
82 xlmt = [0 0.3];
83 disp(titl)
84 analysis(Gc, G, k, w, titl, t, xlmt, yf)
85
86 %           RiseTime: 0.0245
87 %       SettlingTime: 0.1365
88 %       SettlingMin: 1.8000
89 %       SettlingMax: 2.4440
90 %       Overshoot: 22.2103
91 %       Undershoot: 0
92 %           Peak: 2.4440
93 %       PeakTime: 0.0580
94
95 %=====
96 %=====
97 % Dominant Pole With Zero Compensation
98 %=====
99
100 titl = 'Dominant Pole With Zero Compensation';
101
102 fn = @(fI) 45 - atand(To*fI/f2) - atand(To*fI/f3);
103 fI = fzero(fn, 0)
104
105 fc = To*fI          % 64.58
106 wI = 2*pi*fI       % 1.623
107 Gc = wI/s*(1+s/w1);
108
109
110 xlmt = [0 0.030];
111 disp(titl)

```

```

112 analysis(Gc, G, k, w, titl, t, xlmt, yf)
113
114 %           RiseTime: 0.0034
115 %       SettlingTime: 0.0175
116 %       SettlingMin: 1.8012
117 %       SettlingMax: 2.3496
118 %           Overshoot: 17.4798
119 %           Undershoot: 0
120 %           Peak: 2.3496
121 %           PeakTime: 0.0079
122
123 %=====
124 %=====
125 % Dominant Pole With Zero Compensation (Improved Phase Margin)
126 %=====
127
128 titl = 'Dominant Pole With Zero Compensation (Improved PM)';
129
130 fn = @(fI) 30 - atand(To*fI/f2) - atand(To*fI/f3);
131 fI = fzero(fn, 0)
132
133 fc = To*fI           % 40.88
134 wI = 2*pi*fI        % 1.0276
135 Gc = wI/s*(1+s/wI);
136
137 xlmt = [0 0.03];
138 disp(titl)
139 analysis(Gc, G, k, w, titl, t, xlmt, yf)
140
141 %           RiseTime: 0.0053
142 %       SettlingTime: 0.0159
143 %       SettlingMin: 1.8014
144 %       SettlingMax: 2.1219
145 %           Overshoot: 6.0951
146 %           Undershoot: 0
147 %           Peak: 2.1219
148 %           PeakTime: 0.0113
149
150 %=====
151 %=====
152 % Lead Compensation
153 %=====
154
155 fz = 100           % = f2
156 fp = 100*fz
157 fn = @(fc) 120 + atand(fc/fz) - atand(fc/f1) ...
158       - atand(fc/f2) - atand(fc/f3) - atand(fc/fp);
159 fc = fzero(fn, 0)   % 187
160 Gco = fz*fc/(To*f1*f2) % 0.0749
161 wz = 2*pi*fz;
162 wp = 2*pi*fp;
163
164 titl = 'Lead Compensation';
165 Gc = Gco*(1+s/wz)/(1+s/wp);
166
167 xlmt = [0 0.008];
168 disp(titl)

```

```

169 analysis(Gc, G, k, w, titl, t, xlmt, yf)
170
171 %           RiseTime: 0.0013
172 %       SettlingTime: 0.0039
173 %       SettlingMin: 1.7192
174 %       SettlingMax: 2.0517
175 %           Overshoot: 8.0633
176 %           Undershoot: 0
177 %           Peak: 2.0517
178 %           PeakTime: 0.0027
179
180 %=====
181 %=====
182 % Combined Compensation
183 %=====
184
185 titl = 'Combined Compensator';
186 % fc, unity gain xover frequency: same value as for lead compensator
187 wI = 2*pi*fc/To % 4.7
188 wz1 = w1;
189 wz2 = w2;
190 % wp: use the same value as for lead compensator
191 Gc = wI*(1+s/wz1)*(1+s/wz2)/(s*(1+s/wp));
192
193 xlmt = [0 0.008];
194 disp(titl)
195 analysis(Gc, G, k, w, titl, t, xlmt, yf)
196
197 %           RiseTime: 0.0013
198 %       SettlingTime: 0.0040
199 %       SettlingMin: 1.8121
200 %       SettlingMax: 2.1662
201 %           Overshoot: 8.3110
202 %           Undershoot: 0
203 %           Peak: 2.1662
204 %           PeakTime: 0.0027
205 end
206
207
208 function s = analysis(Gc, G, k, w, titl, t, xlmt, yf)
209
210 % loop gain
211 L = Gc*G*k;
212
213 figure
214 [mag, phase] = bode(L,w);
215 margin(mag, phase, w);
216
217 h = gcr;
218 h.AxesGrid.Xunits = 'Hz';
219 h.AxesGrid.TitleStyle.FontSize = 12;
220 h.AxesGrid.XLabelStyle.FontSize = 12;
221 h.AxesGrid.YLabelStyle.FontSize = 12;
222
223 %=====
224
225 % closed loop gain

```

```
226 Gs = 1/k * L/(1+L);
227
228 % Plot of Step response results
229 figure
230 yout = step(Gs, t);
231 plot(t, yout);
232 grid on;
233 title(titl, 'FontSize', 12);
234 xlabel('Time (sec)', 'FontSize', 12);
235 ylabel('Magnitude', 'FontSize', 12);
236 xlim(xlmt);
237
238 % Time Domain Analysis Parameters
239 s = stepinfo(yout, t);
240 ss_error = (yout(end) - yf) / yf * 100
241 end
```



## Part II

# Classical Control: Application





## Chapter 6

# Modelling - Introduction

### 6.1 Introduction

In the following chapters we will present a simple practical system to which we will apply precise control. This system is a dc-to-dc power converter for which we desire to have accurate control of the output voltage as it will function as a voltage regulator subject to input voltage and output load variations.

thus we start with a plant which is given as a circuit configuration for which we need to derive a model, which in the case of the classical control design approach we will be applying, will need to be a transfer function model. Using this model a compensator transfer function will be designed and subsequently the closed loop performance will be simulated.

To fully verify the practical design a number of extra steps will need to be undertaken. This involves actual realization of the compensator transfer function into an appropriate circuit which is then used in a circuit level simulation to assess the performance achieved. If deemed not satisfactory then this process may be iterated. This procedure is represented in the flow chart shown in Figure 6.1.

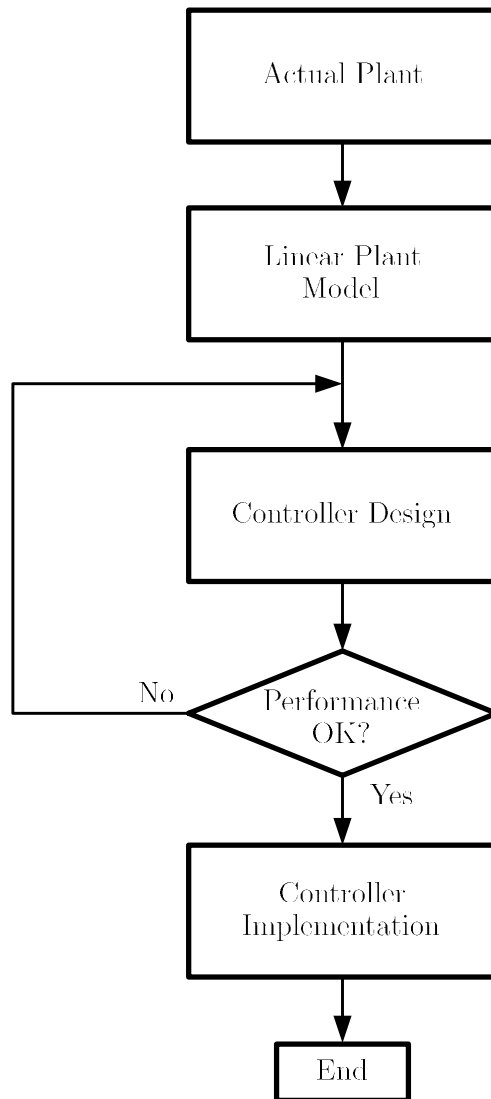


Figure 6.1: Design Flow Diagram

# Chapter 7

## The System

### 7.1 Introduction

In this chapter, the parameters for an example control system will be defined and derived. From the parameters of this example system, subsequent chapters will be devoted to applying control design techniques to optimize system performance parameters.

A control system begins with a model for plant, that has at least one particular parameter to be controlled. To control the plant, the parameter to be controlled is compared to a stable reference value and the difference is input to an error amplifier. The error amplifier then commands the plant, controlling the desired plant parameter. A basic diagram illustrating this architecture is shown in Figure 7.1.

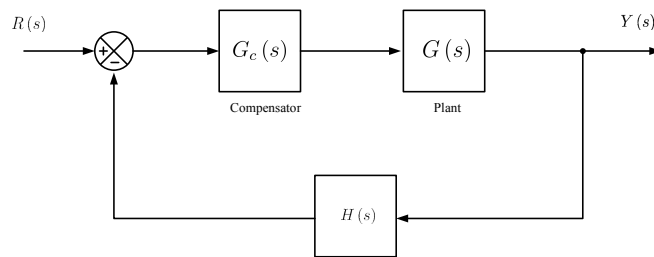


Figure 7.1: Feedback System Block Diagram

## 7.2 The Plant: Buck Converter

### 7.2.1 Introduction

The fundamental item in every control system is the plant, the item that is to be controlled. In this section, the plant will be defined as a buck converter, a switched mode DC power supply.

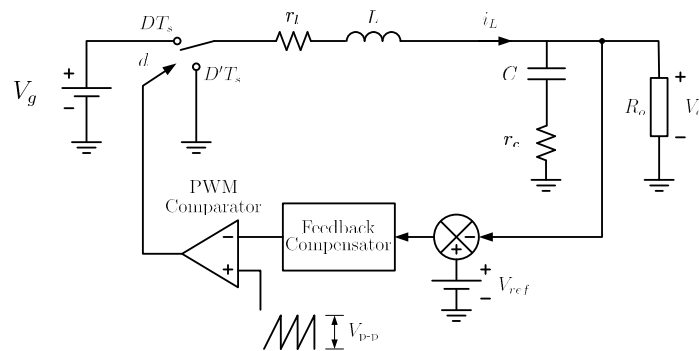


Figure 7.2: Buck Converter Circuit System Diagram

The buck regulator, which is shown complete in Figure 7.2 including circuit losses and feedback compensation, is a basic switched mode power supply. The buck regulator acts to reduce the steady state output voltage based on an applied duty cycle of applied input voltage. The duty cycle is switched at a frequency higher than the resonant frequency LC tank on the output. The output filter allows the circuit to convert the input voltage to a lower output voltage with minimal circuit losses.

The complete system block diagram for the buck regulator is shown in Figure 7.4. The output voltage of the system is fed back to a reference (reduced by  $H(s)$ ), and the difference (error signal) is fed to a compensator which drives a pulse-width modulator to control the output voltage. Additionally, this model includes disturbance inputs in terms of step loading and input voltage variation for design characterization.

### 7.2.2 Transfer Function Derivations

To model the plant based on the diagram of Figure 7.4, three transfer functions are required to be derived. The transfer functions are the control to output  $G_{vd}(s)$ , input voltage to output  $G_{vg}(s)$ , and the output current to output

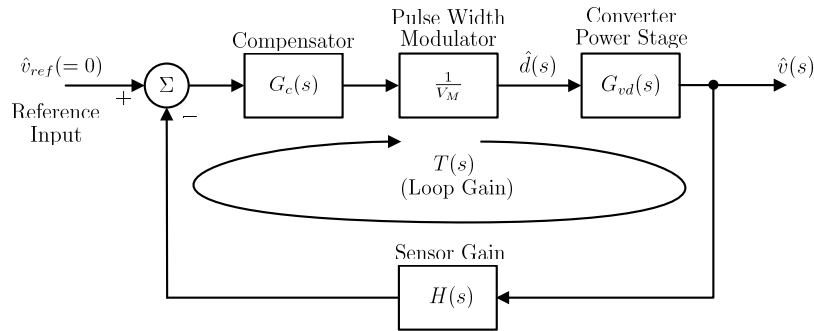


Figure 7.3: Simplified System Diagram

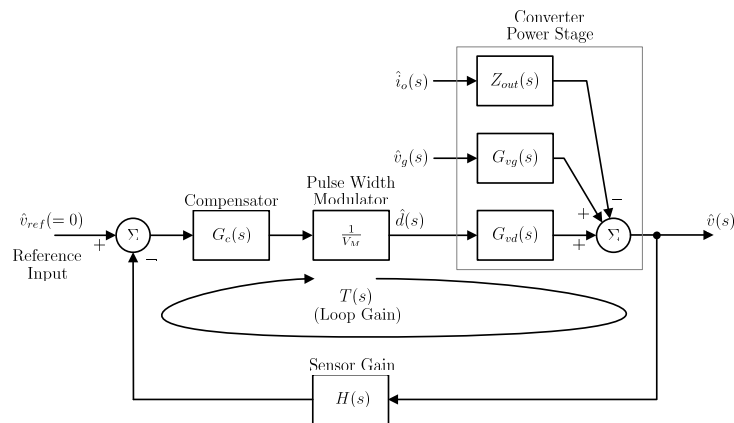


Figure 7.4: Generalized Power System Model

voltage, or open loop output impedance  $Z_{out}(s)$ . Additional transfer functions will be derived in the section, such as the control to inductor current  $G_{id}(s)$ , output current to inductor current  $G_{ii}(s)$ , and the input voltage to inductor current  $G_{ivg}(s)$ . These transfer functions will be utilized in following chapters.

Using the state space analysis approach, the complete set of transfer functions will be derived for the buck converter shown in Figure 7.5.

### $G_{vd}(s)$ Analysis

To analyze the small signal control to output transfer function of the buck converter, an output load change is modeled with a current source, as shown in Figure 7.5. In terms of the state space analysis, this additional source will be modeled as another input variable to the system.

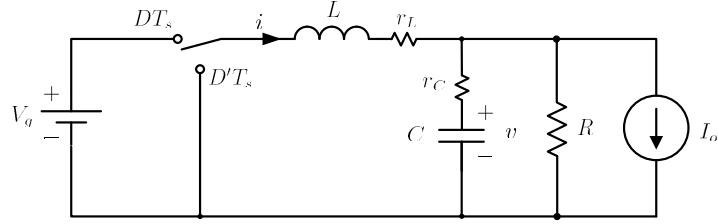


Figure 7.5: Buck Converter Circuit with Non-Ideal Circuit Elements

Listed below are the fundamental equations for state space analysis.  $x$  defines the state variables of the system and the  $u$  variables define the inputs. The number of states is defined by the number of storage elements in the system. For the buck converter, there are two states. The output voltage of the converter is the voltage across the capacitor and the corresponding parasitic resistance.

$$x = \begin{bmatrix} i \\ v \end{bmatrix}$$

$$u = \begin{bmatrix} v_g \\ i_o \end{bmatrix}$$

$$\dot{x} = Ax + Bu$$

$$y = Cx + Eu$$

Applying the principles of superposition to the buck converter, the capacitor current and inductor voltage equations are found and summarized below for both switch positions.

During DTs,

$$L \frac{di_L}{dt} = -(r_l + r_c \parallel R) i_L - \frac{R}{R + r_c} V_c + V_g + I_o (R \parallel r_c)$$

$$C \frac{dV_c}{dt} = \frac{R}{R + r_c} i_L - \frac{1}{R + r_l} V_c + 0V_g - \left( \frac{R}{R + r_c} \right) I_o$$

$$V_{out} = (r_c \parallel R) i_L + \frac{R}{R + r_c} V_c + 0V_g - (R \parallel r_c) I_o$$

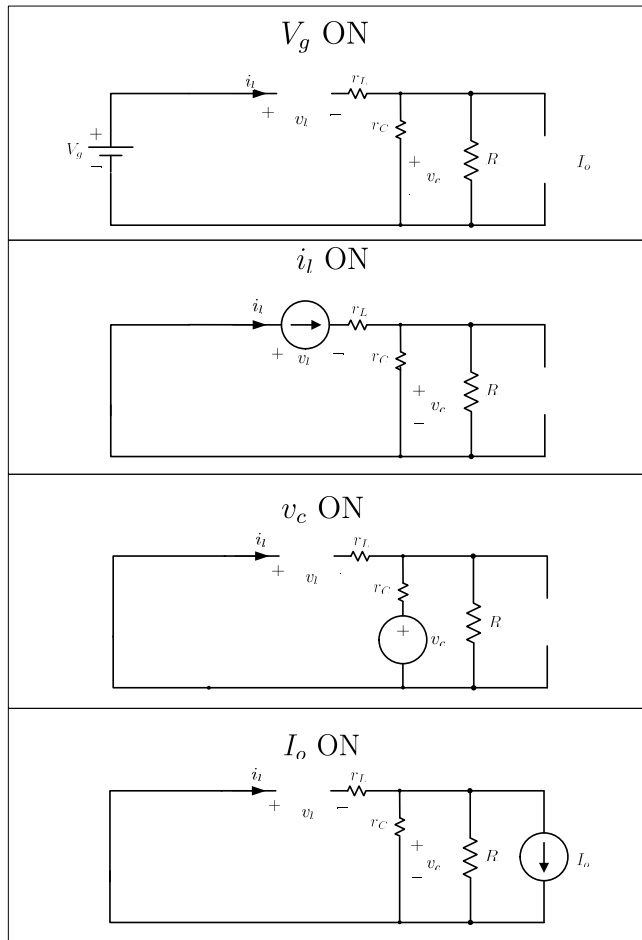


Figure 7.6: Buck Converter Superposition Analysis: DTs

During D'Ts,

$$L \frac{di_L}{dt} = -(r_l + r_c \parallel R) i_L - \frac{R}{R + r_c} V_c + 0V_g + I_o (R \parallel r_c)$$

$$C \frac{dV_c}{dt} = \frac{R}{R + r_c} i_L - \frac{1}{R + r_l} V_c + 0V_g - \left( \frac{R}{R + r_c} \right) I_o$$

$$V_{out} = (r_c \parallel R) i_L + \frac{R}{R + r_c} V_c + 0V_g - (R \parallel r_c) I_o$$

With the circuit defined over the two subintervals, the A, B, C, and E matrices can be defined as shown below:

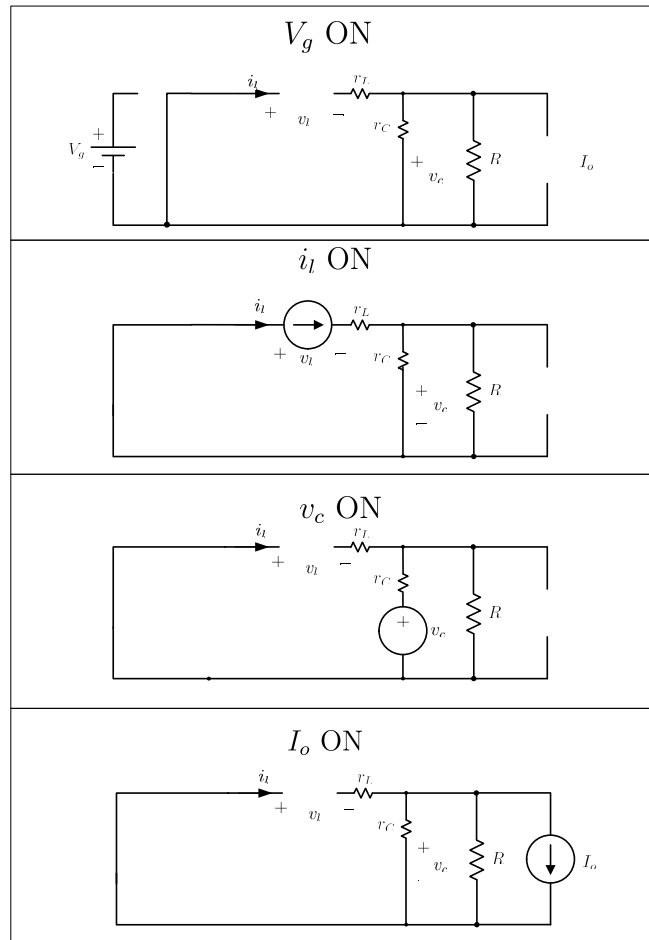


Figure 7.7: Buck Converter Superposition Analysis: D'Ts

$$A_1 = A_2 = A = \begin{bmatrix} -\frac{(r_c \parallel R + r_l)}{L} & -\frac{R}{(R+r_c)L} \\ \frac{R}{(r_c+R)C} & -\frac{1}{(r_c+R)C} \end{bmatrix}$$

$$B_1 = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{D}{L} \\ 0 \end{bmatrix}$$



$$C_1 = C_2 = C = \left[ \begin{array}{c|c} (r_c || R) & \frac{R}{R+r_c} \end{array} \right]$$

$$E_1 = E_2 = E = \left[ \begin{array}{c|c} 0 & -(r_c || R) \end{array} \right]$$

With the state space matrices defined, the control to output transfer function can be calculated as  $G_{vd}(s) = C(sI - A)^{-1}Bd + Ed$ , where  $B_d = (A_1 - A_2)X + (B_1 - B_2)U$  and  $E_d = (C_1 - C_2)X + (E_1 - E_2)U$ . Applying basic matrix manipulation techniques,  $G_{vd}(s)$  is calculated below.

$$X = -A^{-1}BU = \left[ \begin{array}{c} \frac{DV_g}{R+r_l} - I_o \left( \frac{R^2}{(R+r_c)(R+r_l)} - \frac{Rr_c}{(R+r_c)(R+r_l)} \right) \\ I_o \left( \frac{R^2 r_c}{(R+r_c)(R+r_l)} + \frac{R(Rr_c + Rr_l + r_c r_l)}{(R+r_c)(R+r_l)} \right) + \frac{DRV_g}{R+r_l} \end{array} \right]$$

$$B_d = (A_1 - A_2)X + (B_1 - B_2)U = \left[ \begin{array}{c} \frac{V_g}{L} \\ 0 \end{array} \right]$$

$$E_d = (C_1 - C_2)X + (E_1 - E_2)U = 0$$

$$G_{vd}(s) = C(sI - A)^{-1}Bd + Ed = \frac{V_g(1 + sRC)}{\frac{r_c + R}{R}s^2 LC + s \left( \frac{L}{R} + (r_l + r_c + \frac{r_c r_l}{R}) C \right) + \frac{(R+r_l)}{R}}$$

### $G_{ii}(s)$ Analysis

When calculating the output load to inductor current transfer function, it can be noticed that the inductor voltage and capacitor current equations will be identical to those used in calculating  $G_{vd}(s)$  above. Using this fact, only the output equation is needed to be derived to find the C and E matrices.

$$y = i$$

$$C_1 = C_2 = C = \left[ \begin{array}{c|c} 1 & 0 \end{array} \right]$$

$$E_1 = E_2 = E = \left[ \begin{array}{c|c} 0 & 0 \end{array} \right]$$

Using the values found above, the output load to inductor current function is equal to  $G_{ii}(s) = -(C(sI - A)^{-1}B + E)$ , with the negative sign due to the defined current direction.

$$G_{ii}(s) = C(sI - A)^{-1}B + E = \frac{\left( \frac{R-r_c}{R+r_l} + sr_c C \right)}{\frac{r_c + R}{R}s^2 LC + s \left( \frac{L}{R} + (r_l + r_c + \frac{r_c r_l}{R}) C \right) + \frac{(R+r_l)}{R}}$$

### $G_{id}(s)$ , $G_{vv_g}(s)$ , and $G_{iv_g}(s)$ Analysis

To calculate  $G_{id}(s)$ , the control to inductor current transfer function,  $G_{vv_g}(s)$ , the input voltage to output voltage transfer function and  $G_{iv_g}(s)$ , the input voltage to inductor current transfer function, the same parameters derived above can be used to solve each equation.

$$G_{id}(s) = C(sI-A)^{-1}Bd+Ed = \frac{V_g}{R} \frac{(1 + s(R + r_c)C)}{\frac{r_c+R}{R}s^2LC + s\left(\frac{L}{R} + (r_l + r_c + \frac{r_cr_l}{R})C\right) + \frac{(R+r_l)}{R}}$$

$$G_{vv_g}(s) = C(sI-A)^{-1}B+E = \frac{D(1 + sr_cC)}{\frac{r_c+R}{R}s^2LC + s\left(\frac{L}{R} + (r_l + r_c + \frac{r_cr_l}{R})C\right) + \frac{(R+r_l)}{R}}$$

$$G_{iv_g}(s) = C(sI-A)^{-1}B+E = \frac{\frac{D}{R}(1 + s(R + r_c)C)}{\frac{r_c+R}{R}s^2LC + s\left(\frac{L}{R} + (r_l + r_c + \frac{r_cr_l}{R})C\right) + \frac{(R+r_l)}{R}}$$

### $Z_o(s)$ Analysis

To calculate the output impedance of the Buck converter, it is possible to use state space analyses techniques. However, due to the input voltage connection of the Buck converter we can take advantage of the fact that the impedance of the buck on the output looks the same regardless of the switch location.

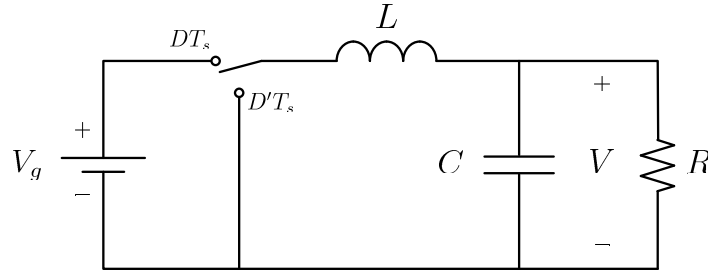


Figure 7.8: Ideal Buck Converter Circuit

$$Z_o(s) = sL \parallel \frac{1}{sC} \parallel R = \frac{sL}{1 + s\frac{L}{R} + s^2LC}$$

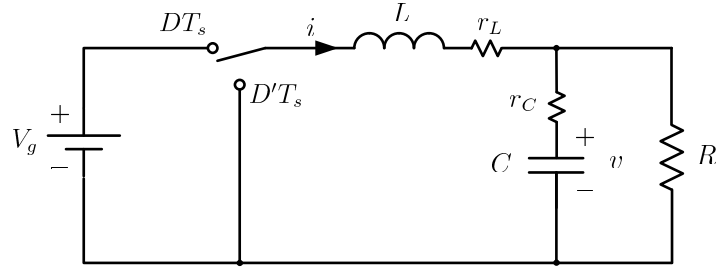


Figure 7.9: Buck Converter Circuit with Non-Ideal Circuit Elements

One way to incorporate losses into the impedance function is to replace the energy storage element in the impedance equation with the sum of the element and its non-ideal resistive component. Starting with the inductor, every instance of  $sL$  is replaced with  $sL + r_l$  in  $Z_o(s)$ .

$$Z_o(s) = \frac{sL + r_l}{1 + \frac{r_l}{R} + s\left(\frac{L}{R} + r_l C\right) + s^2 LC}$$

Assuming  $\frac{r_l}{R} \ll 1$

$$Z_o(s) = \frac{sL + r_l}{1 + s\left(\frac{L}{R} + r_l C\right) + s^2 LC}$$

Now, adding the ESR of the capacitor:

$$sC \Rightarrow \frac{sC}{1 + sRr_c}$$

$$Z_o(s) = \frac{(sL + r_l)(1 + sr_c C)}{1 + s\left(\frac{L}{R} + (r_l + r_c)C\right) + s^2\left(1 + \frac{r_c}{R}\right)LC}$$

The complete output impedance of the Buck converter is found to be:

$$Z_o(s) = \frac{r_l \left(1 + s\frac{L}{r_l}\right) (1 + sr_c C)}{1 + s\left(\frac{L}{R} + (r_l + r_c)C\right) + s^2 LC}$$

Assuming  $\left(\frac{r_c}{R} \ll 1\right)$ .

To summarize the results of this section, Figure 7.10 shows the results of the state space analysis for the buck converter presented in this chapter. The results of the analysis will be leveraged throughout the text.

Buck Converter Transfer Functions		
	Ideal Case	Losses Included
	$Q = \frac{R}{L}\sqrt{LC}$ $r_l = 0$ $\omega_0 = \frac{1}{\sqrt{LC}}$ $r_c = 0$	$Q = \frac{\sqrt{LC}}{(r_c+r_l)C+\frac{L}{R}}$ $r_l \neq 0, \frac{r_l}{R} \ll 1$ $\omega_0 = \frac{1}{\sqrt{LC}}$ $r_c \neq 0, \frac{r_c}{R} \ll 1$
	$\Delta(s) = 1 + \frac{s}{\omega_0 Q} + \left(\frac{s}{\omega_0}\right)^2$	
$G_{vd} \triangleq \frac{\hat{v}}{\hat{d}}$	$\frac{V_g}{\Delta(s)}$	$\frac{V_g(1+sr_cC)}{\Delta(s)}$
$G_{vv_g} \triangleq \frac{\hat{v}}{\hat{v}_g}$	$\frac{D}{\Delta(s)}$	$\frac{D(1+sr_cC)}{\Delta(s)}$
$Z_{out} = G_{vi_o} \triangleq \frac{\hat{v}}{\hat{i}_o}$	$\frac{sL}{\Delta(s)}$	$\frac{r_l(1+\frac{sL}{R})(1+sr_cC)}{\Delta(s)}$
$G_{id} \triangleq \frac{\hat{i}}{\hat{d}}$	$\frac{\frac{V_g}{R}}{\Delta(s)}$	$\frac{\frac{V_g}{R}(1+sr_cC)}{\Delta(s)}$
$G_{iv_g} \triangleq \frac{\hat{i}}{\hat{v}_g}$	$\frac{\frac{D}{R}(1+sRC)}{\Delta(s)}$	$\frac{\frac{D}{R}(1+sRC)}{\Delta(s)}$
$G_{iio} \triangleq \frac{\hat{i}}{\hat{i}_o}$	$\frac{1}{\Delta(s)}$	$\frac{(1+sr_cC)}{\Delta(s)}$

Figure 7.10: Summary of Plant Transfer Functions

### 7.3 Pulse-width Modulator

With a complete model in place for the Buck regulator, the next item in the system diagram to derive is the pulse-width modulator.

From inspection of Figure 13.2, it can be approximated that the duty cycle can be represented by the following relationship:

$$d(t) = \frac{V_c(t)}{V_M} \text{ for } 0 \leq V_c(t) \leq V_M$$

Rearranging,  $\frac{d(t)}{V_c(t)} = \frac{1}{V_M}$ . For a complete derivation confirming the PWM conversion gain can be approximated to this equation, refer to the analysis presented in [1].

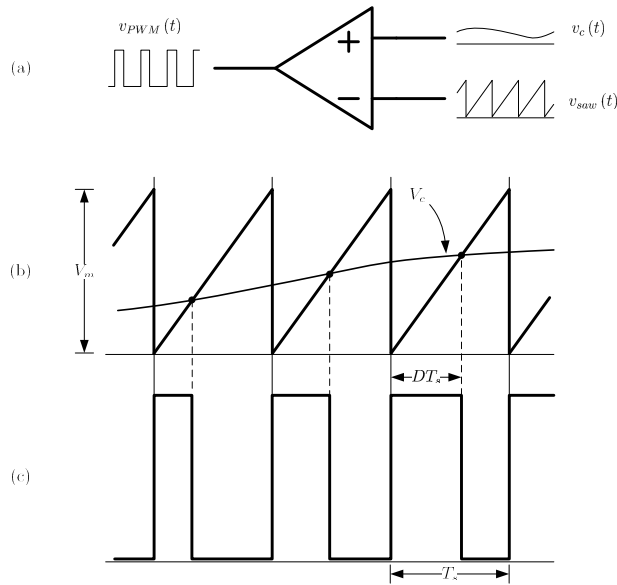


Figure 7.11: PWM Conversion Diagram

## 7.4 Summary

Now that the basic system has been defined, the final block to be derived in the Buck converter system model is the compensator,  $G_c(s)$ . The compensator will be the primary topic of the next several chapters, as it will be leveraged to improve the closed-loop performance of the derived buck regulator system. The equations in this chapter will be heavily leveraged in the remainder of the text.



# References

- [1] R. Middlebrook, "Predicting modulator phase lag in pwm converter feedback loops," in *Powercon*, 1981.





## Chapter 8

# Single Loop Voltage Mode Control

This chapter develops a buck converter design example using different compensation methods to ensure closed loop stability and to optimize system performance. Various compensators are designed using asymptotic Bode plots based primarily on loop bandwidth and stability margins. Computer simulation results are included to show time domain step response behavior and to verify performance improvements.

### 8.1 Introduction

The buck converter is a switch mode, DC-DC, power supply. It accepts a source voltage,  $V_g$  and produces a lower output voltage,  $V$  with high efficiency. An important component of a practical buck converter is control feedback which assures a constant output voltage and attenuates unwanted disturbances. The feedback loop of a buck converter presents several challenges which are explored in the compensation examples.

In this paper we present a series of example buck converter feedback compensation approaches. The design of the buck converter circuit is kept constant to allow comparison of the effects of different compensation schemes. The primary tool that will be applied to evaluate the different compensation approaches are asymptotic Bode plots which are drawn based on corner frequencies of each block in the regulator system. This methodology provides a quick and efficient assessment of circuit performance and an intuitive sense for the trade offs for each compensation approach. Bode plots also directly illuminate the two critical loop stability characteristics, gain and phase margin (GM and PM respectively).

Additional analysis of each compensation approach is undertaken through computer simulation. The PECS [1] circuit simulator is used to evaluate the

effects of  $V_g$  transients, a common problem in real power supply designs. A Matlab [2] simulation is also performed to validate the manual Bode analysis and to determine the exact gain and phase margin. Finally a closed loop Matlab simulation is used to show the ability of the feedback system to attenuate undesired effects as a function of frequency.

## 8.2 Buck Converter System Models

### 8.2.1 General Model

Figure 8.1 is a block diagram of the system components of a buck converter with feedback. The converter power stage accepts  $V_g$  as its power source and the control input  $d(s)$  to produce the output voltage  $V$ . The feedback sensor  $H(s)$ , monitors the converter output voltage which is then compared with a reference voltage  $V_{ref}$ . The difference output of these two voltages is provided to the feedback compensation circuit  $G_c(s)$  and then to the pulse width modulator (PWM) which produces the control waveform for the switching converter  $d(s)$ . The resulting loop gain is thus given by

$$T(s) = G_c(s) \left( \frac{1}{V_M} \right) G_{vd}(s)H(s) \quad (8.1)$$

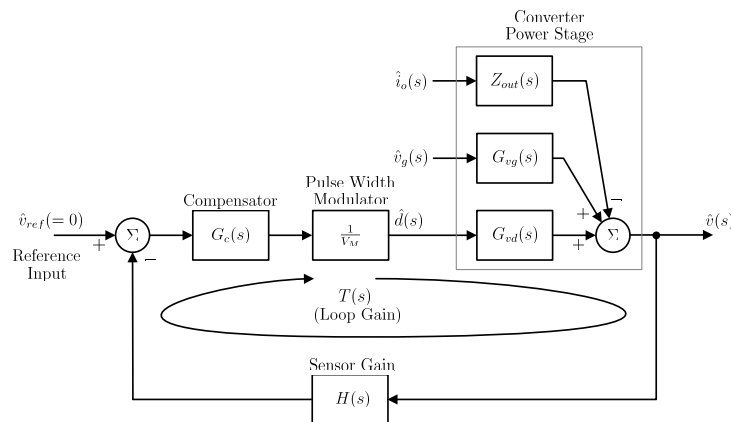


Figure 8.1: Generalized Power System Model

### 8.2.2 Simplified System Model

The general buck converter block diagram provides a complete model for analysis of converter. However, for our analysis we will use a simplified model show in Fig. 8.2 which includes only the elements required for the analysis we will provide. We do not evaluate any source of disturbance except  $V_g$  transients.

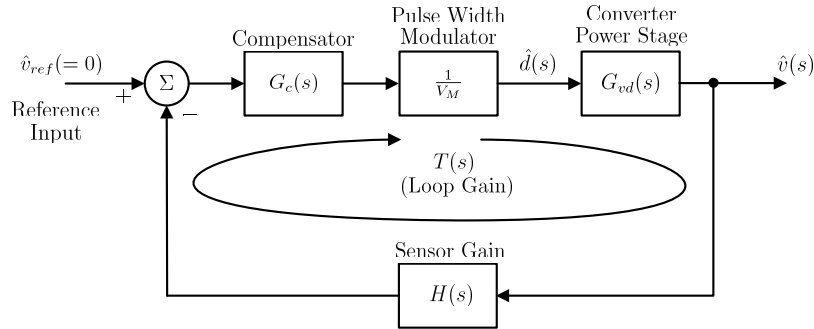


Figure 8.2: Simplified System Diagram

### 8.2.3 Design Targets

To facilitate easy comparison between the selected compensation schemes, the design of the buck converter is fixed with specified values. These values are specified in Table 8.1.

Table 8.1: Specified values

Name	Value	Description
$V_g$	28V	Input Voltage
$V$	15V	Output Voltage
$I_{load}$	5A	Load current
$L$	50uH	Buck inductor value
$C$	500uF	Buck capacitor value
$V_m$	4V	PWM ramp amplitude
$H(s)$	1/3	Sensor gain
$f_s$	100kHz	PWM frequency

### 8.2.4 Buck Converter Model Analysis

Figure 8.3 shows a schematic model for the power converter block. The LCR is a second order circuit with a transfer function described by equation (8.2). It has a resonant frequency value,  $\omega_o = 6.28\text{k rad/s}$  or  $f_o(= \frac{\omega_o}{2\pi}) = 1.0\text{ kHz}$  from (8.3) and a  $Q$  of 9.5 from (8.4). The low frequency gain of the converter is equal to  $V_g$  which is specified to be 28V.

$$G_{vd}(s) = V_g \frac{1}{1 + \frac{s}{Q\omega_0} + \left(\frac{s}{\omega_0}\right)^2} \quad (8.2)$$

$$\omega_o = \frac{1}{\sqrt{LC}} \quad (8.3)$$

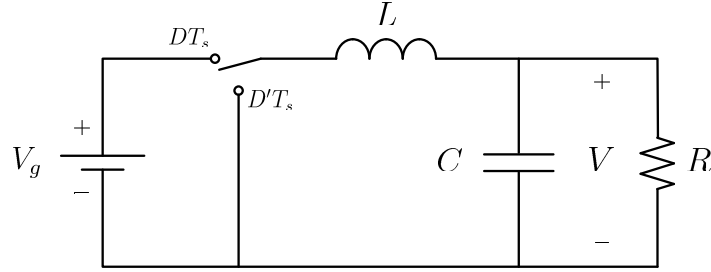


Figure 8.3: Converter Power Stage

$$Q = R\sqrt{\frac{C}{L}} \quad (8.4)$$

Consider the transfer function  $v(s)/v_d(s)$  of the low pass filter formed by the LCR network. The switching frequency  $f_s = 100\text{kHz}$  is much higher than the resonant frequency  $f_0 = 1\text{kHz}$  of the LCR network. During circuit operation, the switch toggles the LCR input between  $V_g$  and ground with a duty cycle  $D$  determined by the feedback loop. A Fourier analysis of the LCR input waveform includes an average DC component  $V = DV_g$  and an  $f_s$  fundamental component and its harmonics as typified by a rectangular waveform. The LCR acts as a low pass filter with a cut off frequency equal to  $f_o$ . It passes the DC component to the output but attenuates  $f_s$  and its harmonics.

### 8.3 Uncompensated System

It is instructive to start our evaluation by examining the uncompensated loop gain, that is with a  $G_c(s) = 1$ . The loop gain is then given from (8.1) as

$$T(s) = \frac{T_o}{1 + \frac{s}{Q\omega_0} + \left(\frac{s}{\omega_o}\right)^2} \quad (8.5)$$

where

$$T_o = \frac{V_g H(0)}{V_m} \quad (8.6)$$

To construct a Bode plot we use the values from equations (8.2)-(8.4) to establish the shape of the Bode magnitude plot. The low frequency gain given by (8.6) has a value of 2.33. The magnitude around  $f_o$  peaks due to the resonant Q of 9.5. At frequencies above  $f_o$  the gain declines at -40dB/decade.

The Bode phase plot is determined only by  $G_{vd}(s)$ . It has a low frequency phase shift of  $0^\circ$ . At  $f_o 10^{-\frac{1}{2Q}}$  or 886Hz ( $\approx 900$ Hz), the phase turns negative and at  $f_o$  the phase has reached  $-90^\circ$ . The phase continues to become more negative until it reaches  $-180^\circ$  at  $10^{\frac{1}{2Q}}$  or 1129Hz ( $\approx 1.1$ kHz). At frequencies higher than 1.1kHz the phase remains at  $-180^\circ$ .

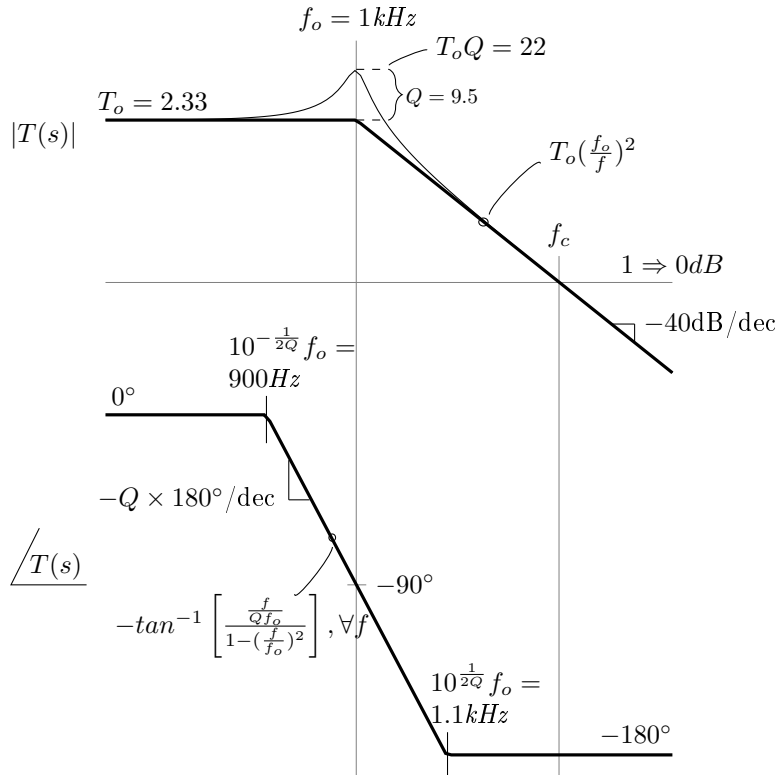


Figure 8.4: Uncompensated Gain and Phase Plot

From the Bode plot it can be determined that unity gain occurs at a frequency,  $f = f_c$  such that

$$T_o \left( \frac{f_o}{f_c} \right)^2 = 1 \quad (8.7)$$

which with  $T_o=2.33$  and  $f_o=1$ kHz, results in  $f_c=1.5$ kHz. At this frequency the phase is  $-180^\circ$  providing zero phase margin. The phase asymptotes show that phase does not cross the  $-180^\circ$  phase level (but is asymptotic to it) which implies that the gain margin is infinite. Figure 8.5 is a Matlab margin plot indicating the actual unity gain frequency to be 1.8 kHz with a phase margin of  $4.7^\circ$ . Also, the Matlab analysis indicates an infinite gain margin.

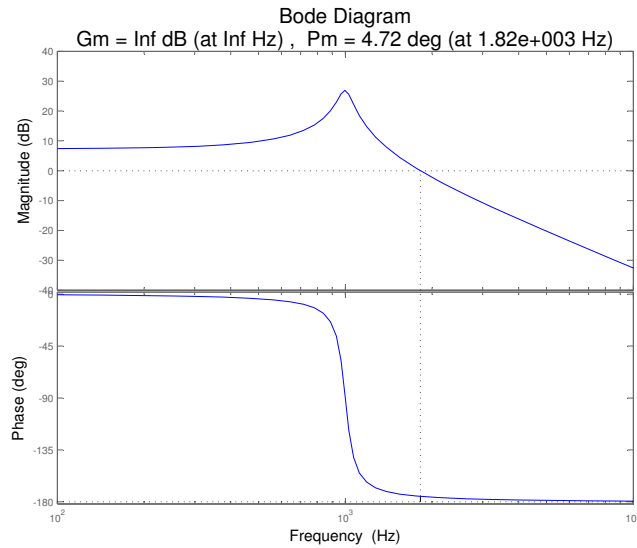


Figure 8.5: Matlab Uncompensated Bode plot

Figure 8.6 shows a PECS implementation of the open loop buck converter system. The input to the modulator is set to 2.1V which results in the target steady state duty ratio of  $D = \frac{V}{V_g} = \frac{15}{28} = 0.54$  required to set the output voltage at  $V = 15\text{V}$  for a nominal input voltage of  $V_g = 28\text{V}$ .

Figure 8.7 shows the output voltage response of the open loop system shown in Figure 8.6 for voltage steps in  $V_g$  of  $28\text{V} \rightarrow 30\text{V} \rightarrow 28\text{V}$ . The response is indicative of the high resonance  $Q$  of 9.5 at the resonant frequency  $f_o = 1\text{kHz}$ . Note also that at an input voltage of  $V_g = 30\text{V}$  the output voltage settles at  $V = DV_g = 0.54 \times 30 = 16.2\text{V}$ , as shown in Fig. 8.7.

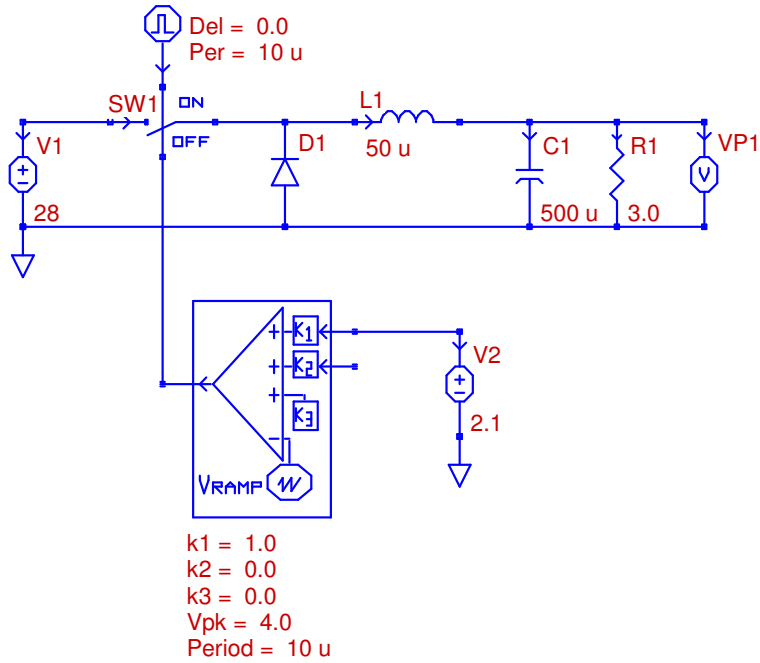


Figure 8.6: PECS Schematic of Open Loop System

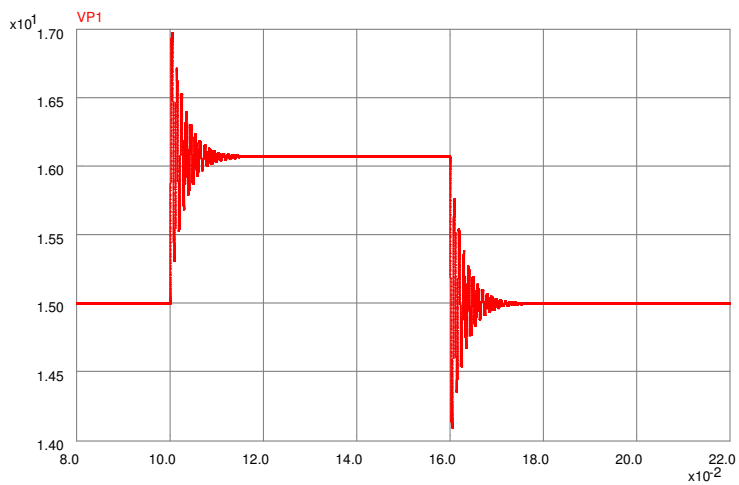


Figure 8.7: PECS Simulation of Open Loop System

## 8.4 Dominant Pole Compensation

Dominant pole compensation is one of the simplest and most common forms of feedback compensation. The motivating idea behind this type of feedback control is to shape the loop gain of the system such that two objectives are achieved:

1. High gain is achieved at DC and low frequencies. This condition ensures low steady state error. In the case of an integrator where the pole appears at zero frequency, i.e. DC, the error is zero.
2. The gain at the plant's lowest frequency pole is less than or equal to 0dB. This condition ensures a positive phase margin and, consequently, stability.

For the case of dominant pole compensation, these objectives are achieved using a compensator consisting of a single pole at a frequency well below those of the plant's poles. For the purposes of this example, an integrator is employed

$$G_c(s) = \frac{\omega_I}{s} \quad (8.8)$$

where  $\omega_I (= 2\pi f_i)$  is an appropriately chosen design constant. Figure 8.8 shows the Bode plot asymptotes for the magnitude and phase of this compensator.

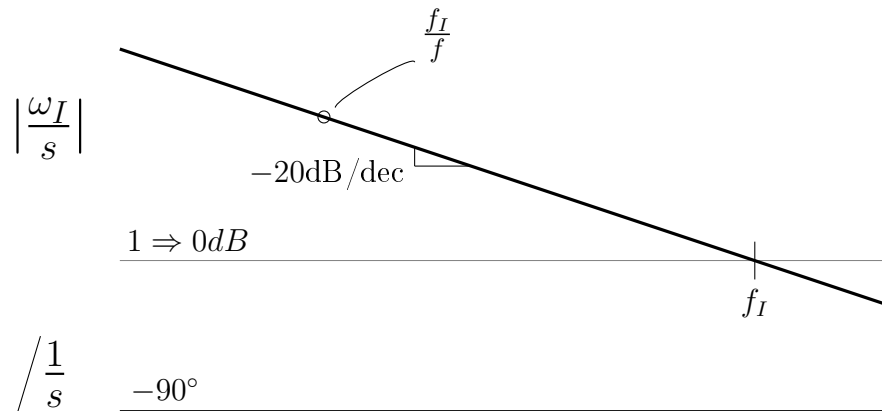


Figure 8.8: Bode Plot of Dominant Pole Compensator

Design of the compensator now consists of selecting an appropriate compensator parameter,  $f_I$ . Following the previously stated criteria, this is a matter of choosing the largest compensator gain such that the total gain at the lowest frequency plant pole(s) is less than 0dB. The loop gain of the system with this compensator is given by



$$T(s) = \frac{\omega_I T_o}{s \left[ 1 + \frac{s}{Q\omega_0} + \left( \frac{s}{\omega_0} \right)^2 \right]} \quad (8.9)$$

Figure 8.9 shows the graphical construction of the phase asymptotes for the loop gain with the compensator. Note that because the plant's dominant pole is second order, it contributes a phase shift of  $-180^\circ$  at high frequencies and a shift of exactly  $-90^\circ$  at  $f_o$ . Furthermore, the compensator contributes its own  $-90^\circ$  phase shift and does so for all frequencies. Consequently, the total phase shift of the compensated loop gain transfer function is  $-180^\circ$  at the dominant pole frequency,  $f_o$ . For this reason it is prudent to design in some additional gain margin. A value of 3dB is initially chosen for this analysis.

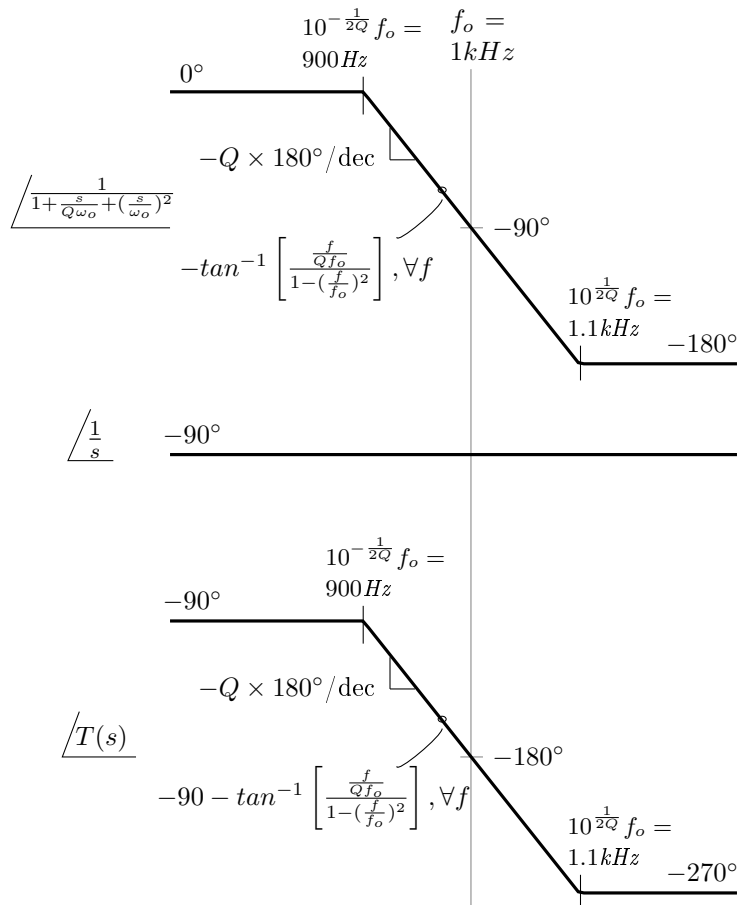


Figure 8.9: Graphical Construction of Phase Asymptotes for Dominant Pole Compensated Loop Gain

Figure 8.10 shows how the plant and compensator transfer functions combine to produce the gain of the compensated open loop. To achieve a loop gain that is -3dB at  $f_o$ , we require the magnitude at  $f_o$  to equal 0.7

$$\frac{f_I T_o Q}{f_o} = 0.7 \quad (8.10)$$

For  $T_o = 2.33$  and  $f_o = 1.0\text{kHz}$ , we find  $f_I = 32$ .

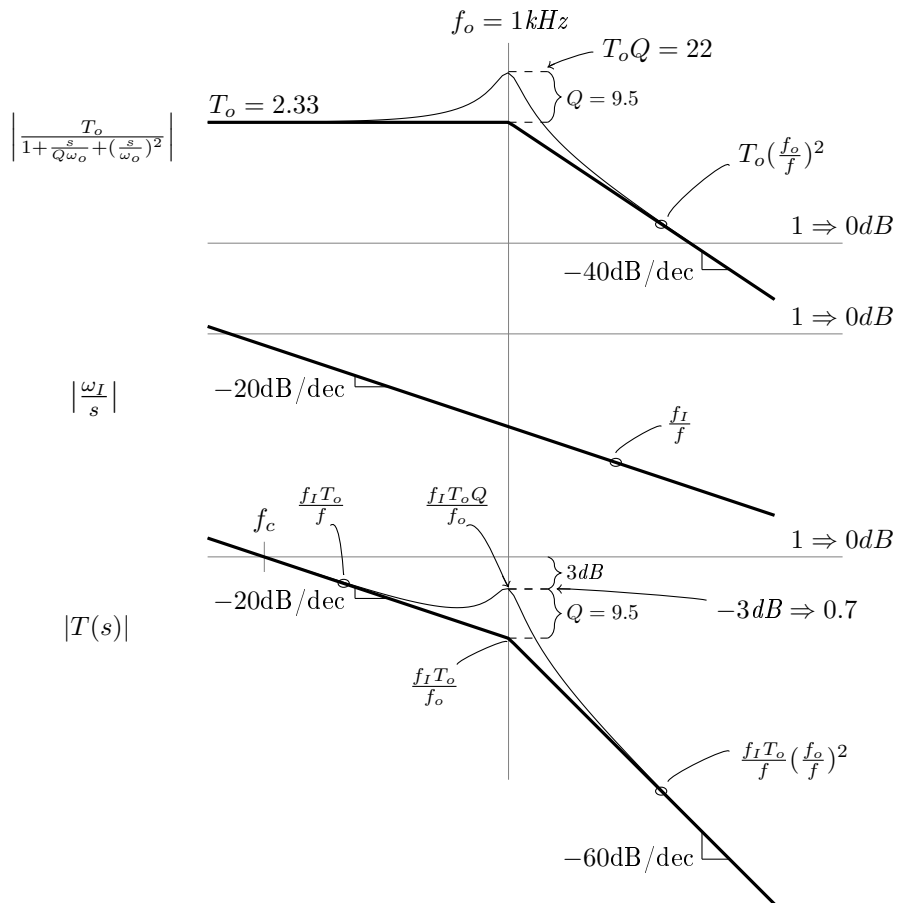


Figure 8.10: Graphical Construction of Gain Asymptotes for Dominant Pole Compensated Loop Gain

Figure 8.11 shows the Bode plot of the resulting gain and phase asymptotes and Figure 8.12 shows a Matlab margin analysis which confirms the design.

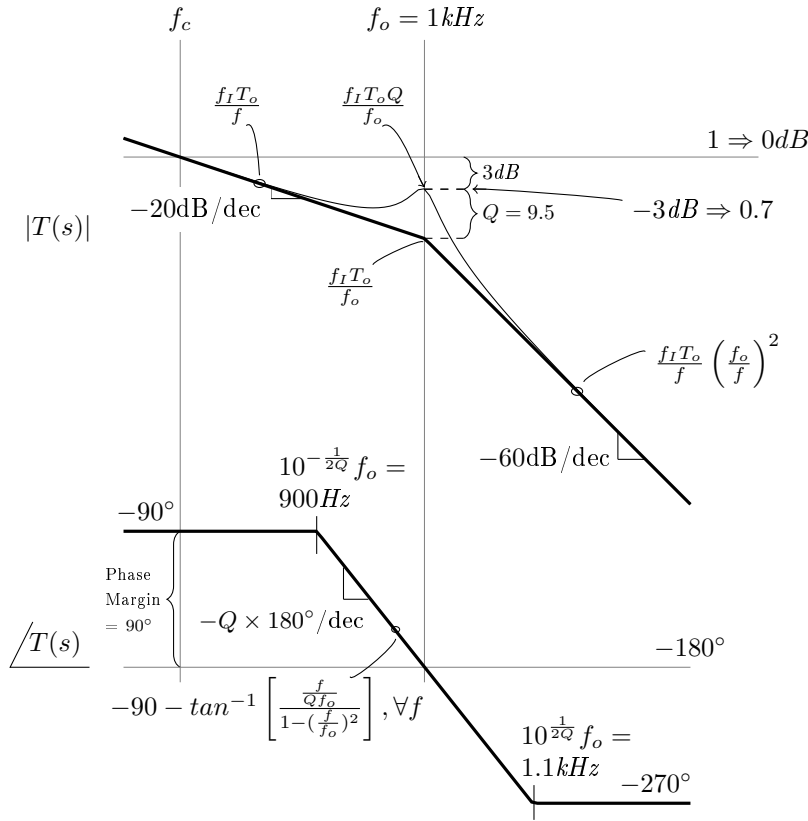


Figure 8.11: Dominant Pole Compensated Loop Gain Asymptotic Bode Plot

With a compensator designed and verified via Matlab, the next stage is to design a circuit that implements the compensator. Figure 8.13 shows the general form of an operational amplifier in an integrator configuration. The transfer function for this circuit is given by:

$$G(s) = \frac{-1}{(s/\omega_o)} \quad (8.11)$$

where

$$\omega_o = \frac{1}{RC} \quad (8.12)$$

where  $\omega_o$  is the frequency at which the integrator gain is unity.

A capacitor value of 50nF is chosen for  $C$ . This value is within the range of low-cost, commercially available ceramic capacitors and is small enough to avoid

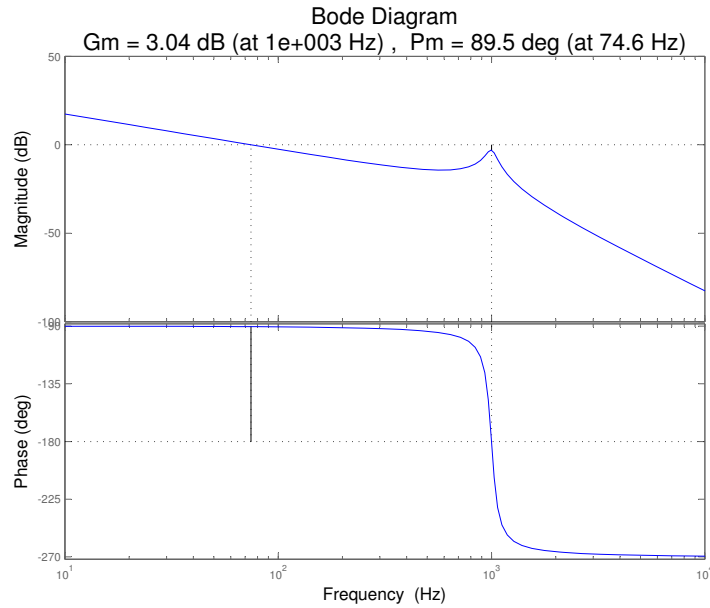


Figure 8.12: Matlab Analysis of Dominant Pole Compensator

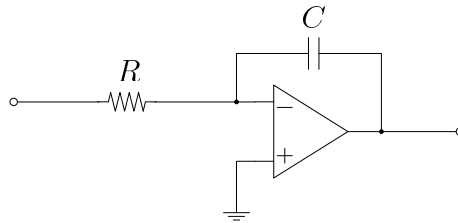


Figure 8.13: Op-Amp Integrator Circuit

any op-amp slew rate issues. Equating  $\omega_o$  with the compensator parameter,  $\omega_I$  ( $= 2\pi f_I$ ) and solving for  $R$  gives

$$R = \frac{1}{\omega_I C} = \frac{1}{2\pi(32)(50\text{nF})} \approx 100\text{k}\Omega \quad (8.13)$$

A PECS simulation is created to verify the time domain performance of the implementation. Figure 8.14 shows the complete PECS circuit model for the design.

Figure 8.15 shows the results of the PECS simulation for a 2V disturbance on the supply voltage,  $V_g$ . The input voltage steps are 28V  $\rightarrow$  30V  $\rightarrow$  28V. The simulation exhibits several undesirable characteristics:

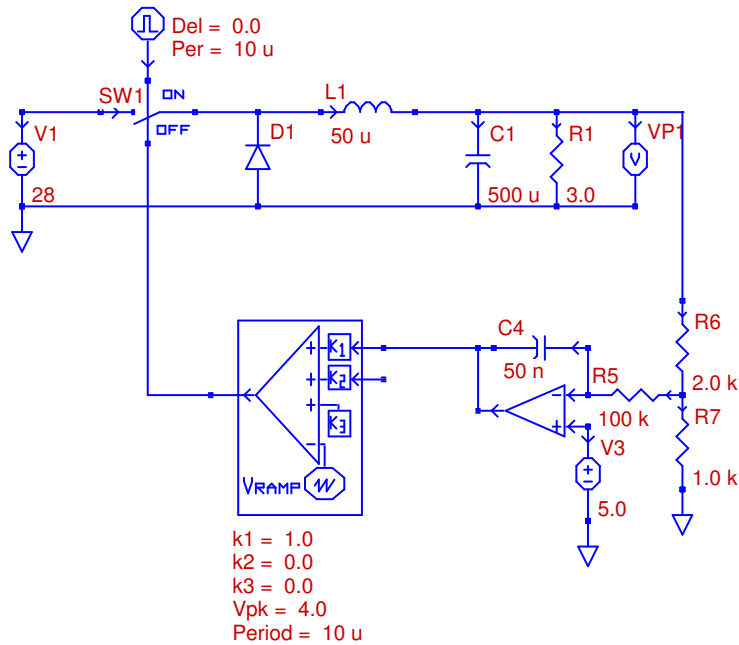


Figure 8.14: PECS Schematic of Dominant Pole Compensated System

1. The regulator does a poor job of rejecting the input voltage disturbance. Nearly all of the input voltage excursion shows up as a transient on the output.
2. The regulator exhibits a substantial amount of ringing in response to the input disturbance. Closer examination of the ringing, as shown in Figure 8.16, reveals that the frequency of the oscillations is the same as the resonant frequency of the plant,  $f_o$ , and is not the result of defective control loop design.

It is clear from the simulation results that, although the design is stable and exhibits zero steady-state error, there is much room for improvement, particularly with respect to its transient response.

One additional experiment is performed using the dominant pole compensation scheme. The  $Q$  of the plant's dominant pole is reduced by placing a large capacitor in series with a small damping resistance. Figures 8.17 and 8.18 show the PECS circuit schematic and simulation results, respectively.

One can see clearly that the ringing of the previous design has been eliminated. Unfortunately, the poor rejection of input voltage transients remains.

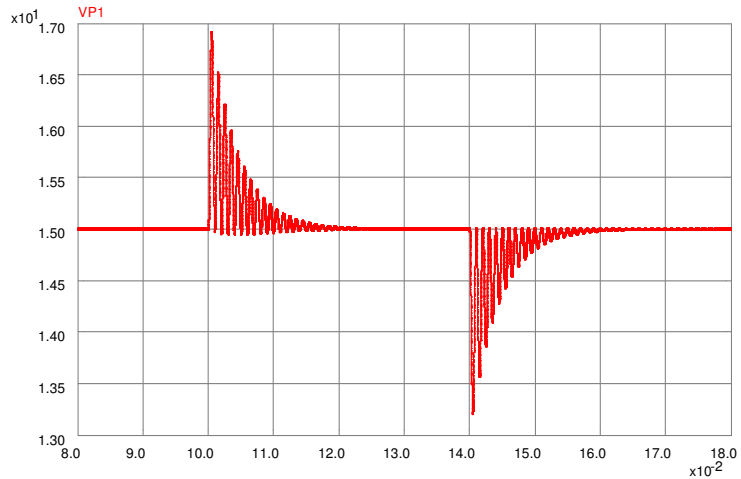


Figure 8.15: PECS Simulation of Dominant Pole Compensated System

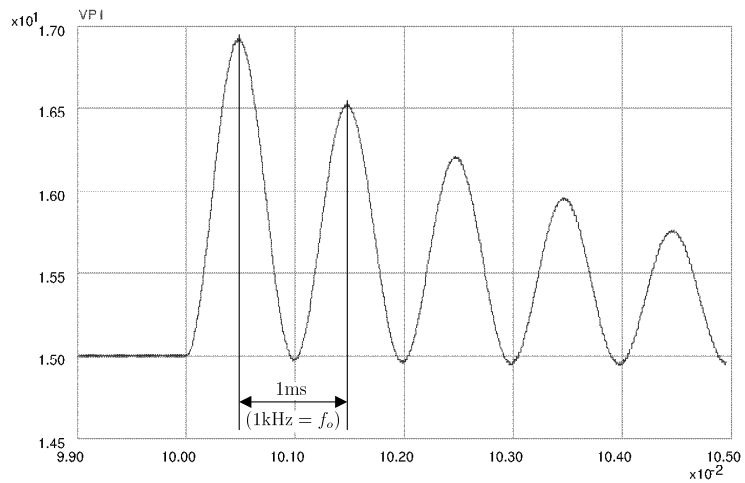


Figure 8.16: Pole of Dominant Pole Simulation Showing Oscillation at Resonant Frequency

Furthermore, this is probably not an ideal solution from a practical standpoint. The large value capacitor will be relatively expensive in terms of component price and physical space. Alternate compensation schemes still offer the potential for better performance at lower cost.

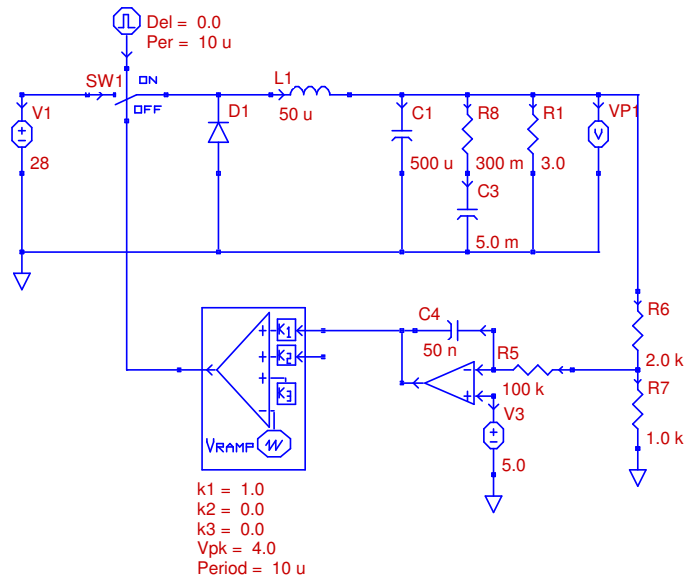


Figure 8.17: PECS Schematic of Dominant Pole with Damping

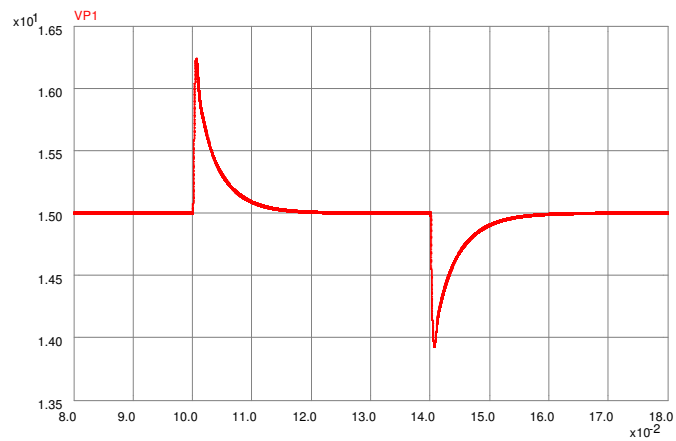


Figure 8.18: PECS Simulation of Dominant Pole Compensated System with Added Output Filter Damping

## 8.5 Dominant Pole Compensation with Zero

The dominant pole compensator of the previous section, while stable and having zero steady state error, exhibits several undesirable characteristics including poor rejection of input supply voltage excursions and pronounced ringing in response to transients. One might assume that these issues are related to the minimal, 3dB, gain margin for which the compensator was designed. This section explores that line of reasoning by modifying the compensator of the previous section in order to substantially increase the gain margin.

The dominant pole compensator is modified by adding a zero at the resonant frequency of the plant and by reducing the gain to -10dB. Overall gain margin is improved in two ways:

1. by directly increasing the gain margin at the resonant frequency,  $f_o$ , from 3dB to 10dB.
2. by shifting the frequency at which the phase reaches  $-180^\circ$  beyond the resonant frequency and the gain peak due to the plant's Q.

The form of the modified compensator transfer function is:

$$G_c(s) = \omega_I \frac{1 + s/\omega_z}{s} \quad (8.14)$$

We will use  $\omega_z = \omega_o$  or  $f_z (= \frac{\omega_z}{2\pi}) = f_o$ . Which results in a loop gain of

$$T(s) = \omega_I \frac{1 + s/\omega_o}{s} \frac{T_o}{1 + \frac{s}{Q\omega_o} + \left(\frac{s}{\omega_o}\right)^2} \quad (8.15)$$

Figure 8.19 shows the resulting Bode plot asymptotes. We would like to set the gain at  $f_o$  to  $1/\sqrt{10}$  (which corresponds to -10dB). From the magnitude plot we see that we want

$$\frac{f_I T_o Q}{f_o} = \frac{1}{\sqrt{10}} \quad (8.16)$$

which given  $T_o = 2.33$ ,  $Q = 9.5$ ,  $f_o = 1kHz$ , results in  $f_I = 14.3$ . Figure 8.20 shows a Matlab confirmation of the Bode plot. Note that a gain margin of 11dB is predicted at a phase cross-over frequency of 1.06kHz, slightly higher than the plant's resonant frequency.

Figure 8.21 shows a standard op-amp implementation with the desired transfer function. The transfer characteristics of the circuit are given by:

$$G(s) = -A \frac{1 + s/\omega_1}{\frac{s}{\omega_1}} \quad (8.17)$$

where:



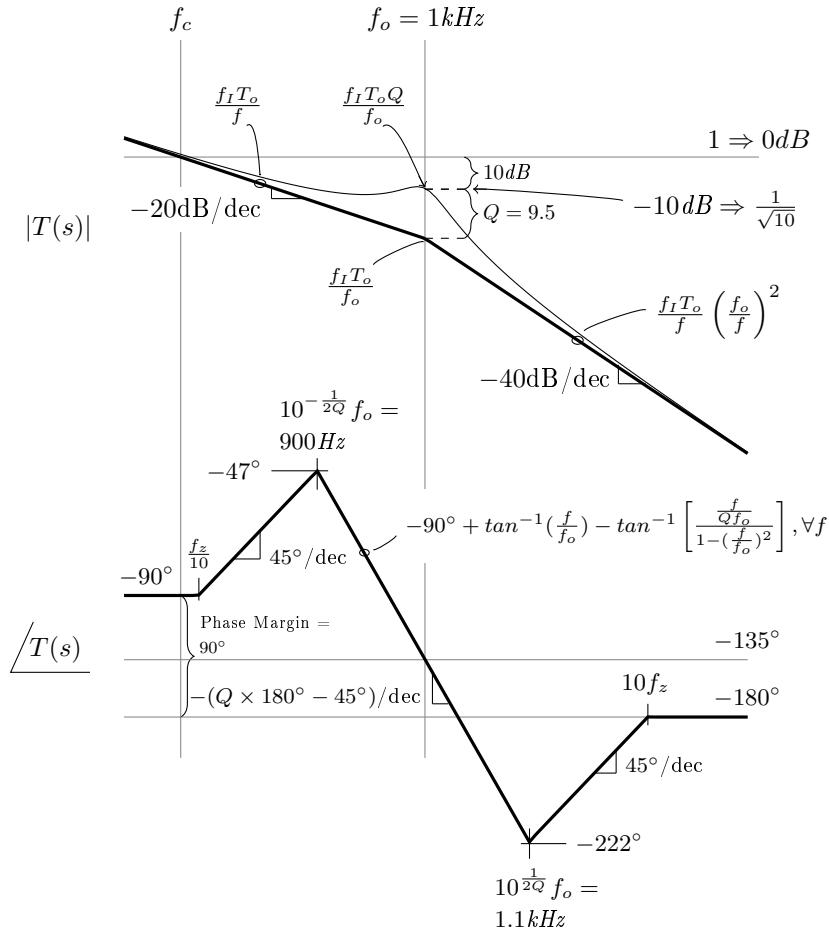


Figure 8.19: Open Loop System Gain and Phase with pole-zero Compensation (10dB GM)

$$A = \frac{R_2}{R_1} \quad \text{and} \quad \omega_1 = \frac{1}{R_2 C_1} \quad (8.18)$$

Equating  $f_1 (= \frac{\omega_1}{2\pi})$  to the plant resonant frequency,  $f_o$  and  $\omega_I$  to  $A\omega_1$  provides two equations with three unknowns. Choosing, somewhat arbitrarily, a value of 100k for  $R_1$ , leads to the following values.

$$R_2 = \frac{f_I R_1}{f_1} = \frac{(14.3)(100\text{k}\Omega)}{(1\text{k}\Omega)} = 1.4\text{k}\Omega \quad (8.19)$$

$$C_1 = \frac{1}{\omega_1 R_2} = \frac{1}{2\pi(1\text{kHz})(1.4\text{k}\Omega)} = 110\text{nF} \quad (8.20)$$

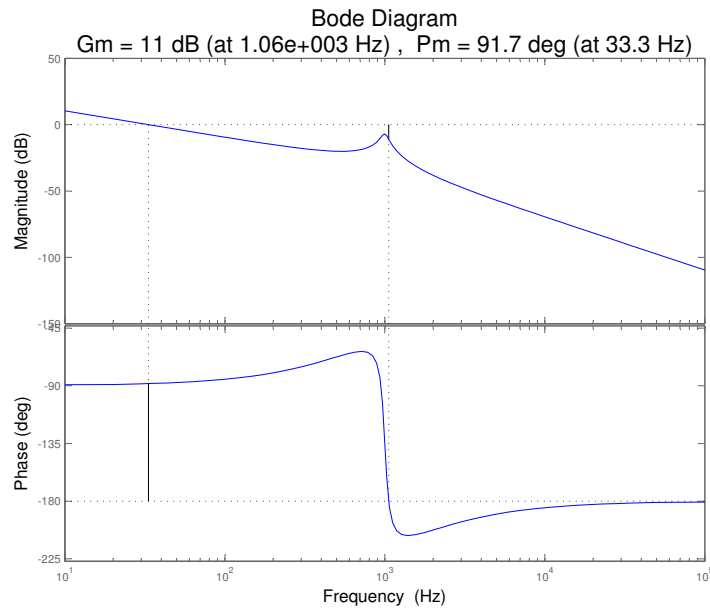


Figure 8.20: Matlab Analysis of Dominant Pole Compensator with Zero

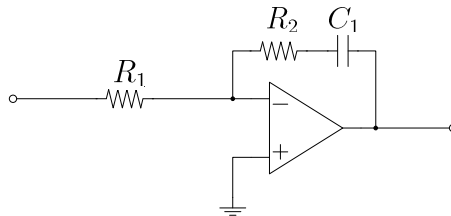


Figure 8.21: op-amp Integrator with Zero

Figure 8.22 shows a PECS circuit implementation of the system with the new compensator. Figure 8.23 shows the response of the system to a transient on the input voltage.

The modified compensator shows little improvement over the original circuit. It still fails to provide good rejection of input voltage transients and the previously observed ringing is still present.

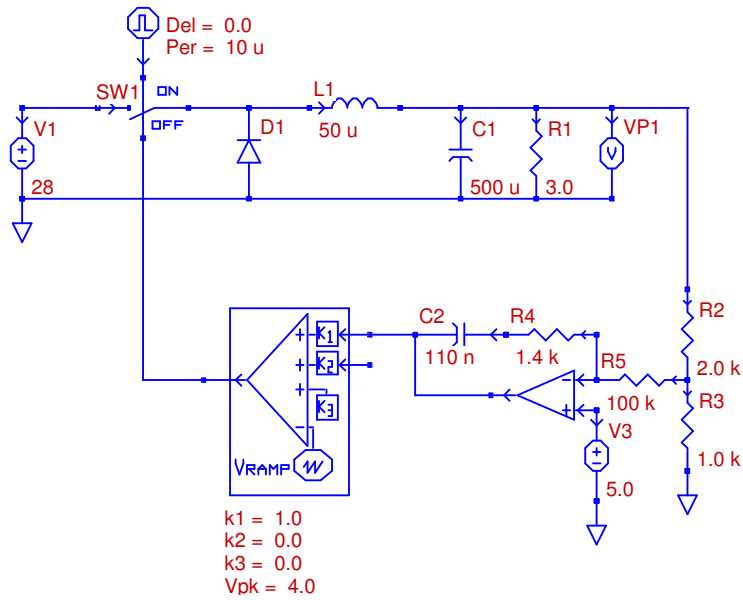


Figure 8.22: PECS Schematic of Dominant Pole with Zero Compensation (10 dB GM)

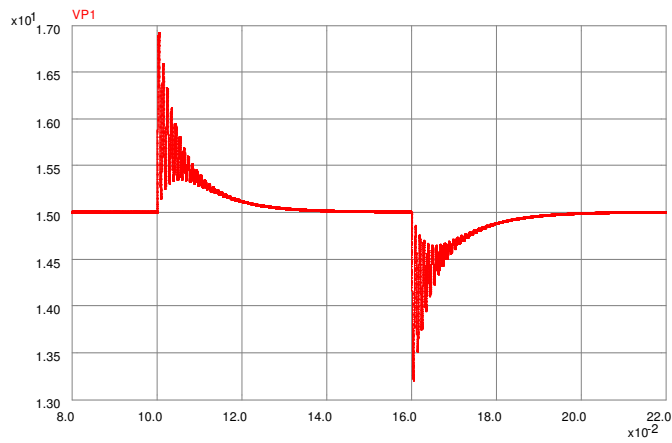


Figure 8.23: PECS Simulation of Dominant Pole with Zero Compensation (10 dB GM)

## 8.6 Lead Compensation

A more sophisticated way to improve the performance of the buck converter is with a lead compensator. The transfer function of this compensator is

$$G_c(s) = G_{c_o} \frac{\left(1 + \frac{s}{\omega_z}\right)}{\left(1 + \frac{s}{\omega_p}\right)}, \quad (8.21)$$

where  $\omega_z < \omega_p$ . As can be seen from the plot of the transfer function shown in Figure 8.24, the lead compensator provides both a phase boost that is adjustable based on the pole and zero frequencies, and a gain boost at higher frequencies that can result in a higher crossover frequency for a lead-compensated buck converter. Generally, a lead compensator is used to provide a phase boost, the level of which is chosen to improve the phase margin to a desired value. The new crossover frequency can be chosen arbitrarily. The design shown here will be to obtain a  $45^\circ$  phase margin and a crossover frequency of 5 kHz for the loop gain with a lead compensator.

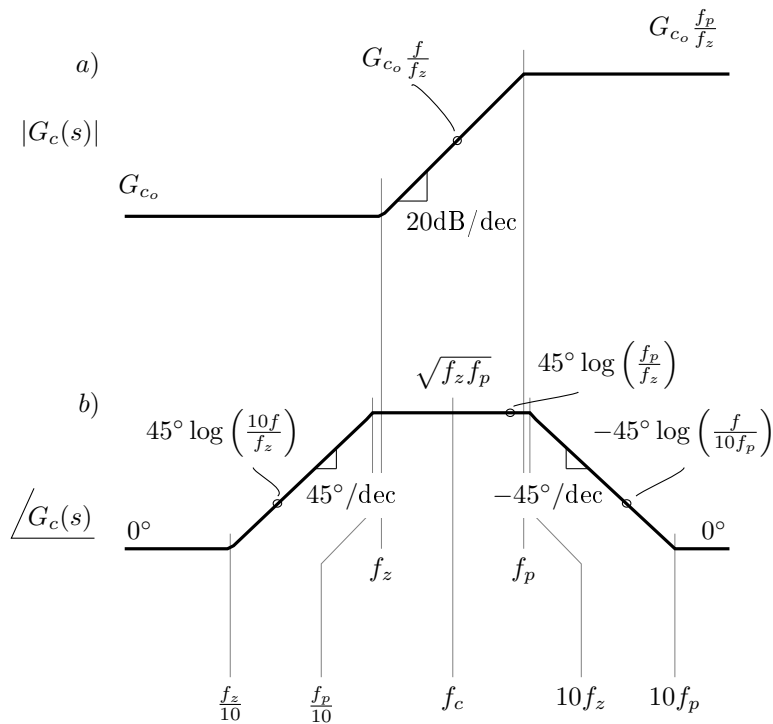


Figure 8.24: Bode Plot of Lead Compensator

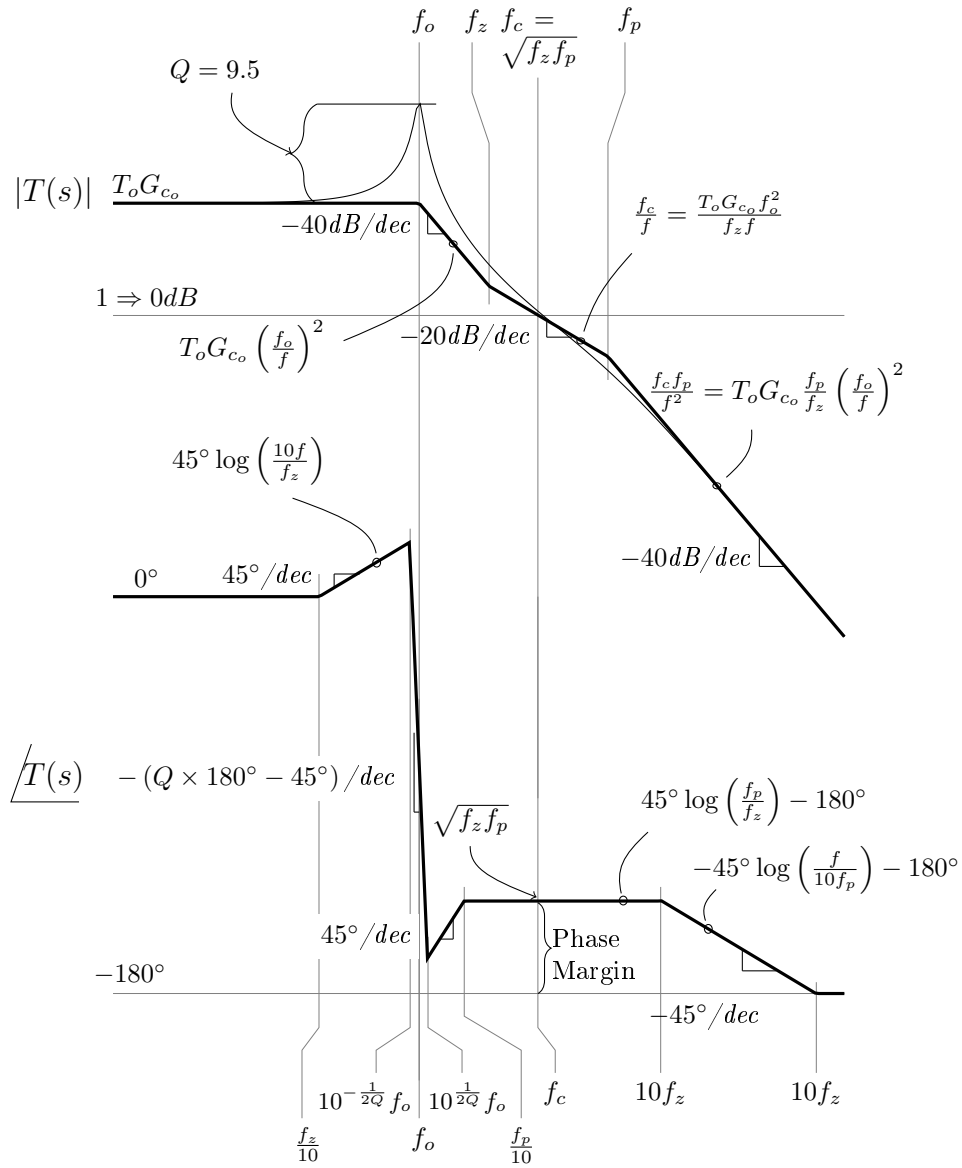


Figure 8.25: Bode Plot of Lead Compensated System

When the compensator is placed in the loop, the loop gain of the buck converter system becomes

$$T(s) = T_0 G_{c_0} \frac{\left(1 + \frac{s}{\omega_z}\right)}{\left(1 + \frac{s}{\omega_p}\right) \left(1 + \frac{s}{Q\omega_0} + \left(\frac{s}{\omega_0}\right)^2\right)} \quad (8.22)$$

The asymptotic Bode plot of this loop gain is shown in Fig. 8.25. The expressions shown can be used to place the pole and zero frequencies of the compensator to obtain the desired phase margin and unity-gain crossover frequency. As can be seen, the phase margin of the lead compensated system is given by

$$\phi_M = 45^\circ \log \left( \frac{f_p}{f_z} \right)$$

For a desired phase margin of  $45^\circ$  we have

$$45^\circ = 45^\circ \log \left( \frac{f_p}{f_z} \right)$$

or

$$f_p = 10f_z$$

Also, the crossover frequency,  $f_c$  will necessarily be the geometric mean of the pole and the zero frequency. Since the phase margin condition gives a relationship between the pole and zero frequencies, this can be used to solve for both.

$$\begin{aligned} f_c &= \sqrt{f_z f_p} \\ 5 \text{ kHz} &= \sqrt{10f_z^2} \\ f_z &= \frac{5 \text{ kHz}}{\sqrt{10}} \\ f_z &= 1.58 \text{ kHz and } f_p = 15.8 \text{ kHz} \end{aligned}$$

These relationships result in the pole and zero frequencies for the lead compensator. To complete the design, the required low-frequency gain  $G_{c_o}$  of the compensator to place the unity-gain point at the appropriate frequency must be determined. This can be found by equating the values of the gain asymptotes at  $f_z$ .

$$T_0 G_{c_0} \left( \frac{f_0}{f_z} \right)^2 = \frac{f_c}{f_z}$$

Substituting the values of  $f_o$  and  $T_o$  for the example converter, and the values of  $f_z$  and  $f_c$  as previously calculated, the gain  $G_{c_o}$  of the compensator is

$$\begin{aligned} G_{c_o} &= \frac{1}{T_0} \left( \frac{f_z}{f_0} \right)^2 \frac{f_c}{f_z} \\ G_{c_o} &= \frac{1}{2.33} \left( \frac{1.58 \text{ kHz}}{1 \text{ kHz}} \right)^2 \frac{5 \text{ kHz}}{1.58 \text{ kHz}} \\ G_{c_o} &= 3.4 \end{aligned}$$

As seen in previous designs and now in the phase plot of Fig. 8.25, the phase response is asymptotic to  $-180^\circ$  at high frequencies and so does not cross through this level which implies an infinite gain margin.

In summary, we have designed a lead compensator which, using asymptotic Bode plot approximations, result in a  $45^\circ$  phase margin with a unity gain frequency of 5kHz and an infinite gain margin. The Matlab simulation in Figure 8.26 verifies the results, above.

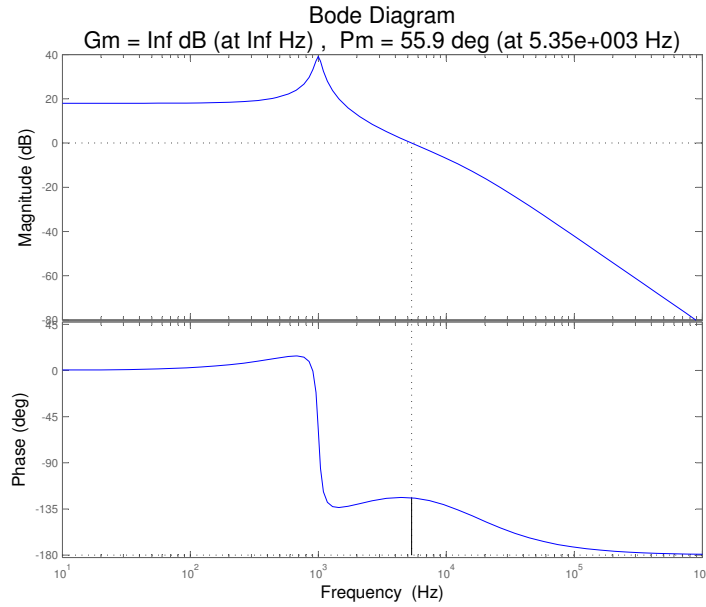


Figure 8.26: Matlab Lead Compensator

With all of the parameters of the lead compensator determined, what remains is to implement the compensator using an op-amp circuit and simulate the closed-loop converter to evaluate its performance. A general circuit that can be used to implement any lead or lag compensator is shown in Fig. 8.27. The transfer function of this circuit is

$$G_c(s) = G_{c_o} \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{\omega_p}} \quad (8.23)$$

where

$$G_{c_o} = -\frac{R_2}{R_1} \quad (8.24)$$

$$f_z = \frac{\omega_z}{2\pi} = \frac{1}{2\pi R_1 C_1} \quad (8.25)$$

$$f_p = \frac{\omega_p}{2\pi} = \frac{1}{2\pi R_2 C_2} \quad (8.26)$$

The resistor ratio sets the low frequency gain, and the two resistor-capacitor pairs set the pole and zero frequencies. Using standard resistor values of  $R_1 = 100 \text{ k}\Omega$  and  $R_2 = 330 \text{ k}\Omega$  results in the required low frequency gain of close to  $G_{c_o} = 3.4$ . Using (8.23) we find

$$C_1 = \frac{1}{2\pi (1.58 \text{ kHz}) (100 \text{ k}\Omega)} \Rightarrow C_1 = 1.0 \text{ nF}$$

$$C_2 = \frac{1}{2\pi (15.8 \text{ kHz}) (330 \text{ k}\Omega)} \Rightarrow C_2 = 33 \text{ pF}$$

It is also necessary to derive a value for the reference voltage on the non-inverting input of the op-amp. The sensed voltage from the output will be 5 V in steady-state as before, and the control voltage should be 2.14 V. Using these in combination with the resistor values for the lead compensator, the reference voltage can be found.

$$V_{ref} = \frac{R_2}{R_1 + R_2} V_{sense} + \frac{R_1}{R_1 + R_2} V_{control}$$

$$V_{ref} = \frac{330 \text{ k}\Omega}{100 \text{ k}\Omega + 330 \text{ k}\Omega} (5 \text{ V}) + \frac{100 \text{ k}\Omega}{100 \text{ k}\Omega + 330 \text{ k}\Omega} (2.14 \text{ V}) = 4.33 \text{ V}$$

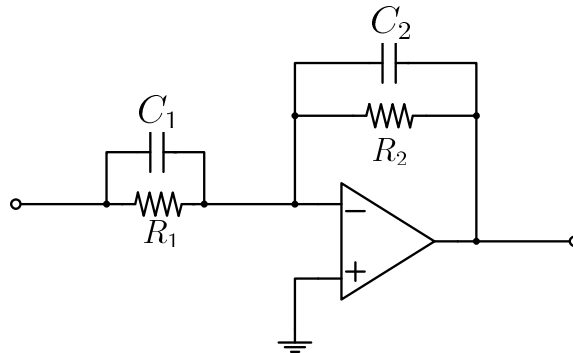


Figure 8.27: op-amp circuit implementation of lead compensator

Using these values in the PECS simulator (see Figure 8.28 for PECS schematic), the response of the lead-compensated buck converter to a step in the input voltage was simulated as before. The results of the simulation are shown in Figure 8.29. The lead compensator is quite effective in increasing the phase margin of the system. The oscillatory behavior evident in the output voltage of the uncompensated converter is not present, and the magnitude of the steady-state error due to the step is reduced, though not eliminated. Thus, the system with the lead compensator is very stable, but will still exhibit steady-state errors to



a step disturbance. To fix this problem, the system type number must be increased by adding a pole at  $s = 0$ , as was seen previously. This is the approach taken in the design of the subsequent compensators.

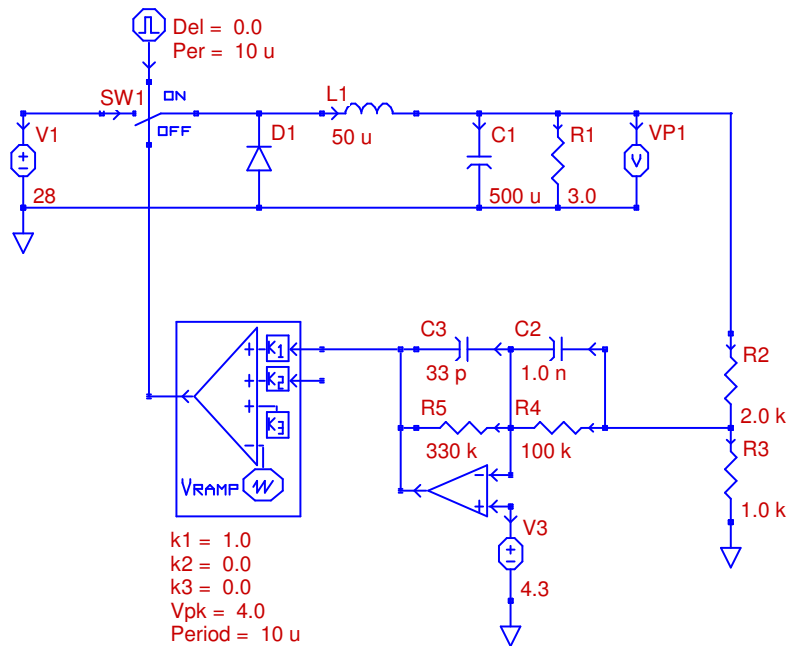


Figure 8.28: PECS Schematic of Lead System

## 8.7 Dominant Pole with Lead Compensation

So far we have seen that with a dominant pole integral compensation a zero steady state error can be achieved at the expense of limited bandwidth with resulting large overshoot in the step response. In contrast lead compensation is able to extend bandwidth thus reducing step response overshoot. However, due to severely curtailed low frequency loop gain, a non-zero steady state error is seen.

In this section a compensator which is a composite of the two previous compensators is examined. The exact form of the compensator is:

$$G_c(s) = \frac{\omega_I \left(1 + \frac{s}{\omega_1}\right) \left(1 + \frac{s}{\omega_z}\right)}{s \left(1 + \frac{s}{\omega_p}\right)} \quad (8.27)$$

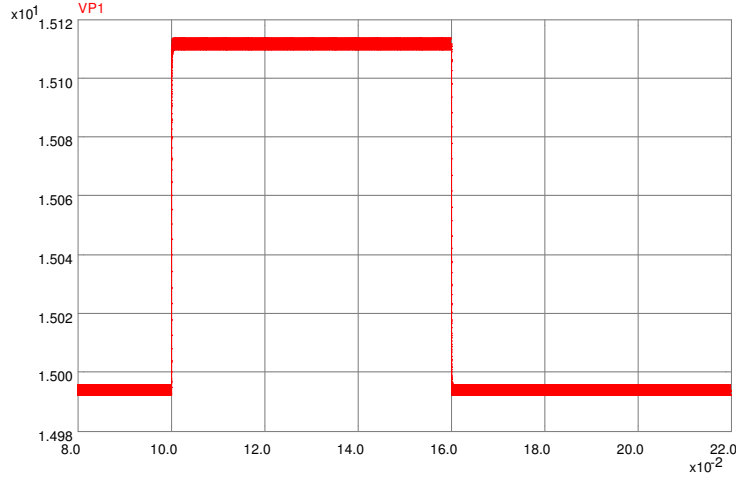


Figure 8.29: PECS Simulation of Lead System

Effectively, to the lead compensator design of the previous section we are adding an integrator pole, i.e. a pole at zero frequency, and a zero at  $f_1 (= \frac{\omega_1}{2\pi})$ . In the following we will consider two different values for the zero frequency  $f_1$ .

In the first case  $f_1$  will be chosen to be the largest frequency which, based on the phase asymptote, contributes  $+90^\circ$  to the crossover frequency  $f_c$ , thus fully canceling the  $-90^\circ$  contribution from the integrator pole. This effectively leaves the phase margin unchanged from the lead compensator design of the previous section. From the phase asymptotes plots of a zero, we see that the zero frequency  $f_1$  should be at  $\frac{f_c}{10}$  which is 500Hz.

In the second design considered here we will lower the zero frequency to  $f_1 = 150\text{Hz}$  and examine the effect on the closed loop performance.

In either case the expression for the loop gain is

$$T(s) = T_o \frac{\omega_I \left(1 + \frac{s}{\omega_1}\right) \left(1 + \frac{s}{\omega_z}\right)}{s \left(1 + \frac{s}{\omega_p}\right) \left[1 + \frac{s}{Q\omega_o} + \left(\frac{s}{\omega_o}\right)^2\right]} \quad (8.28)$$

where  $f_1$  is either 500Hz or 150Hz, as discussed above and  $f_I (= \frac{\omega_I}{2\pi})$  is the only design variable to be determined.

### 8.7.1 Design 1: Zero $f_1 = 500\text{Hz}$

As before, asymptotic plots for the loop gain are drawn. As the construction of the phase plot is more involved than that of the magnitude plot, its construction is shown separately in Fig. 8.30. In Fig. 8.30, the top plot is that of the previous lead compensation design, as seen in Fig. 8.25. The plot of the phase of the component  $\frac{1}{s} \left(1 + \frac{s}{\omega_1}\right)$  is shown in the center plot where  $f_1 = 500\text{Hz}$ . The final phase plot for the new loop gain is shown in the bottom plot. Both magnitude and phase plots for the new loop gain are shown together in Fig. 8.31.

To determine,  $f_I$ , the one unknown variable in the loop gain, we note that at the frequency  $f_1$  the magnitude is set equal to the low frequency loop gain of the lead compensation design of the last section.

$$T_0 \frac{f_I}{f_1} = T_0 G_{c_o}|_{lead} \quad (8.29)$$

For  $f_1 = 500$  we find  $f_I = 1770$ . Thus the expression for the compensator is as given in (8.27) with the following values

$$\begin{aligned} \omega_I &= 2\pi(1770) \\ \omega_1 &= 2\pi(800) \\ \omega_z &= 2\pi(1580) \\ \omega_p &= 2\pi(15800) \end{aligned} \quad (8.30)$$

To confirm the accuracy of the design, the Bode plot of the exact loop gain was evaluated using Matlab. This is shown in Fig. 8.32. Our asymptotic design values of crossover frequency  $f_c$  and phase margin of  $5\text{kHz}$  and  $45^\circ$ , respectively were determined by Matlab as given by the Matlab "margin" command to be more precisely  $5,370\text{Hz}$  and  $50.5^\circ$ , respectively, thus confirming the design procedure.

A compensator which realizes the transfer function is shown in Fig. 8.33 where we find

$$\begin{aligned} \omega_I &= \frac{1}{R_1(C_2 + C_3)} \\ \omega_1 &= \frac{1}{R_2 C_2} \\ \omega_z &= \frac{1}{R_1 C_1} \\ \omega_p &= \frac{1}{R_2 \frac{C_2 C_3}{C_2 + C_3}} \end{aligned} \quad (8.31)$$

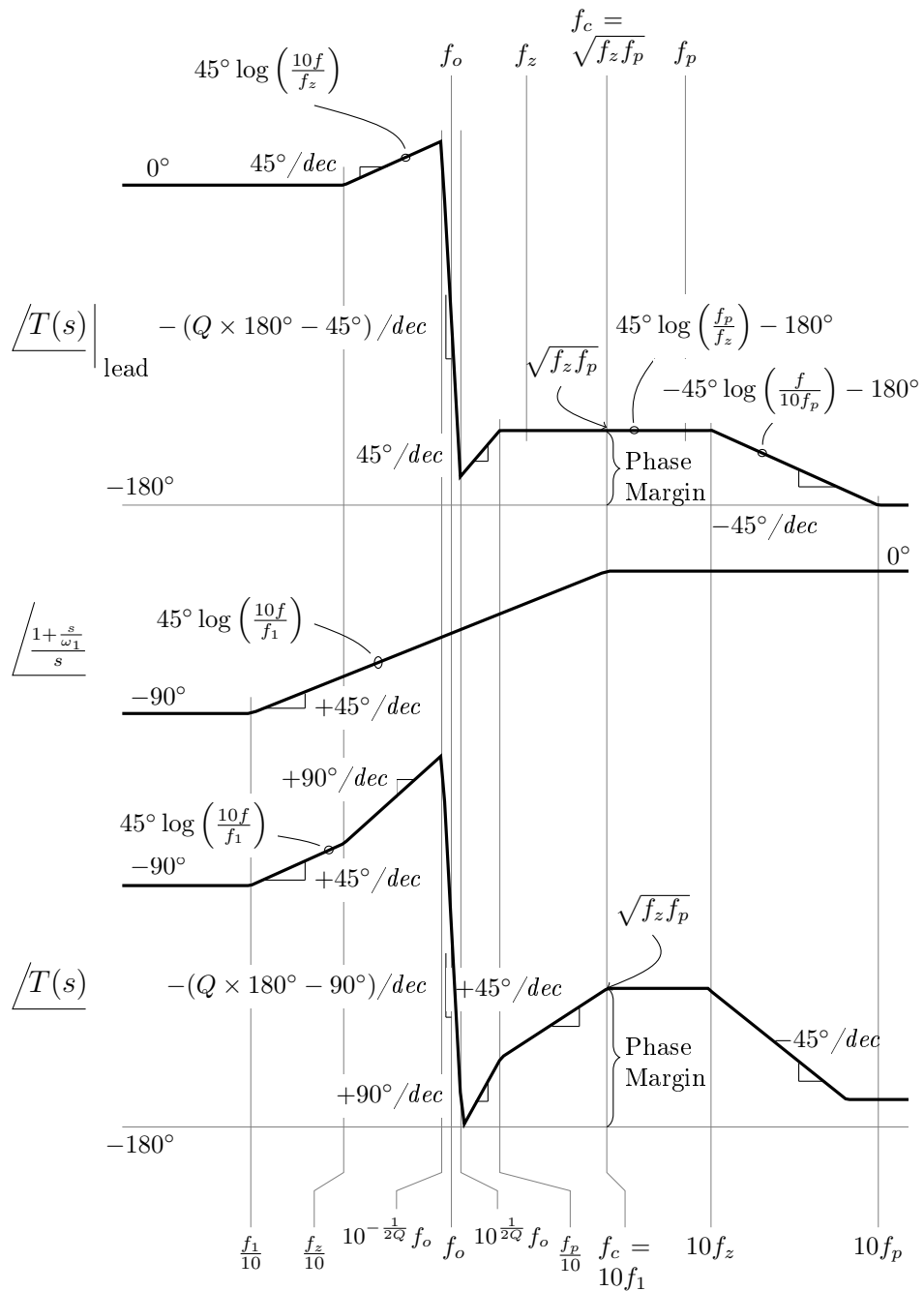


Figure 8.30: Bode Plot of Lead Compensator (500Hz)

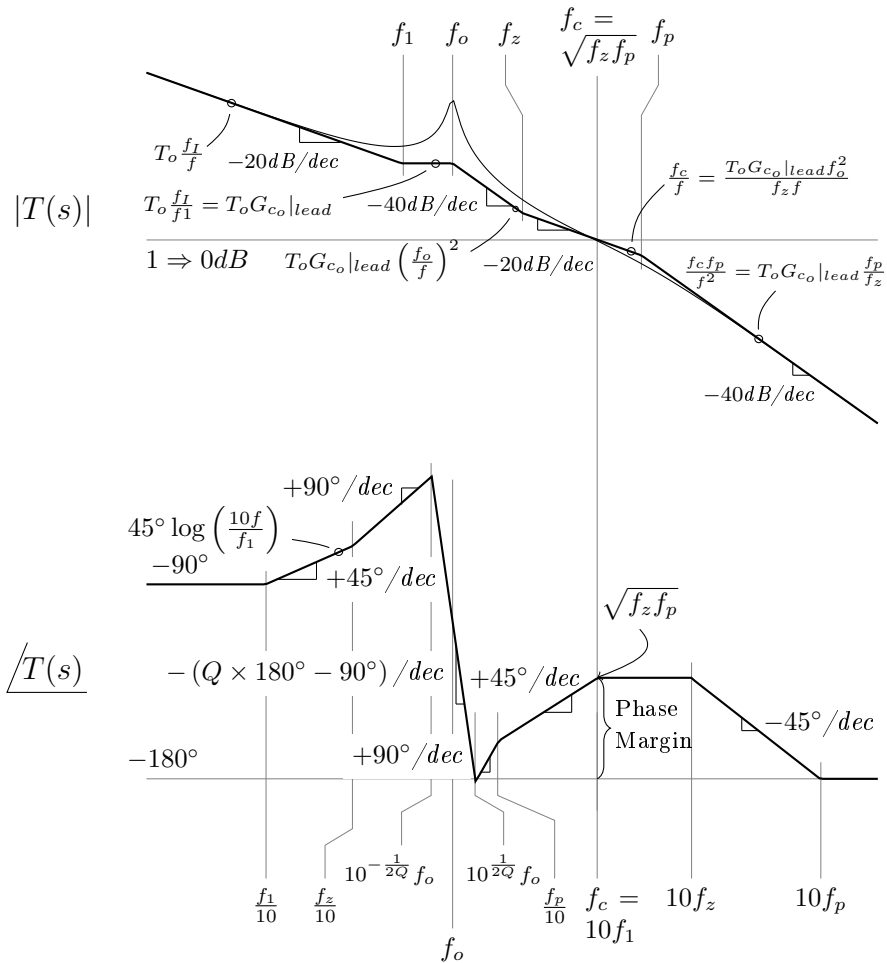


Figure 8.31: Bode Plot of System with Lead plus Integral Compensation (500Hz)

Setting  $R_1 = 100\text{K}$  and using the approximation  $C_3 \ll C_2$  we find the component values:

$$\begin{aligned}
 C_1 &= \frac{1}{\omega_z R_1} = 2.2\text{nF} \\
 C_2 &= \frac{1}{\omega_I R_1} = 1\text{nF} \\
 R_2 &= \frac{1}{\omega_1 C_2} = 330\text{k}\Omega \\
 C_3 &= \frac{1}{\omega_p R_2} = 33\text{pF}
 \end{aligned}
 \tag{8.32}$$

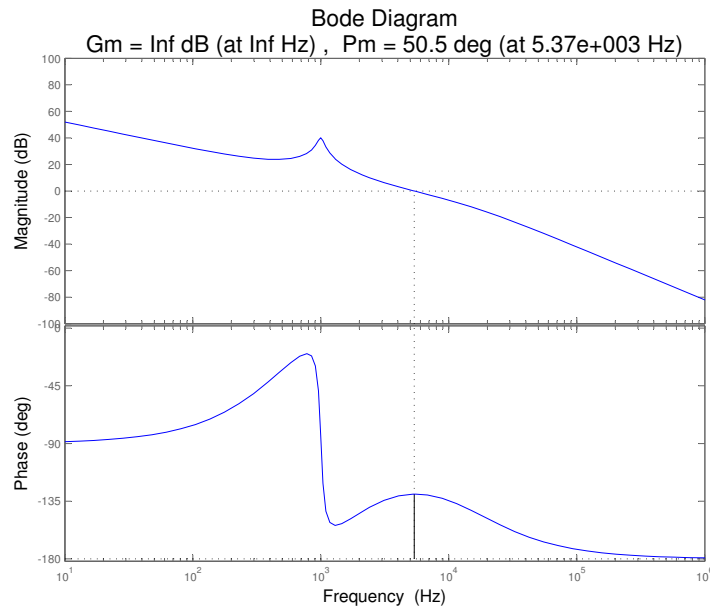


Figure 8.32: Matlab Lead Compensator with Integrator and Zero at 500Hz

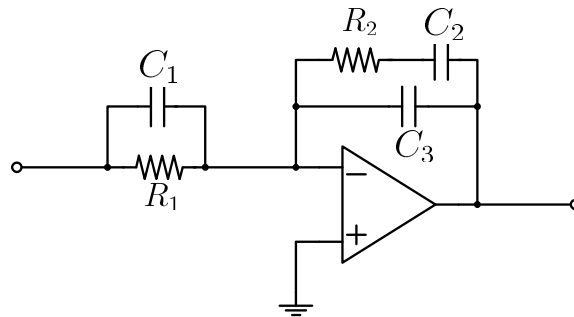


Figure 8.33: Compensator Circuit for Dominant Pole with Lead Compensation

A PECS implementation of the closed loop system is shown in Fig. 8.34. The simulated response of input voltage steps  $26V \rightarrow 30V \rightarrow 28V$  is shown in Fig. 8.35. Clearly seen here is the zero steady state error and a maximum voltage deviation of around 80 mV with a settling time of around 1 ms.

### 8.7.2 Design 2: Zero $f_1 = 150$ Hz

The above design procedure will now be repeated for the case of the zero  $f_1 = 150$  Hz. The resulting asymptotic phase plot construction is shown in Fig.

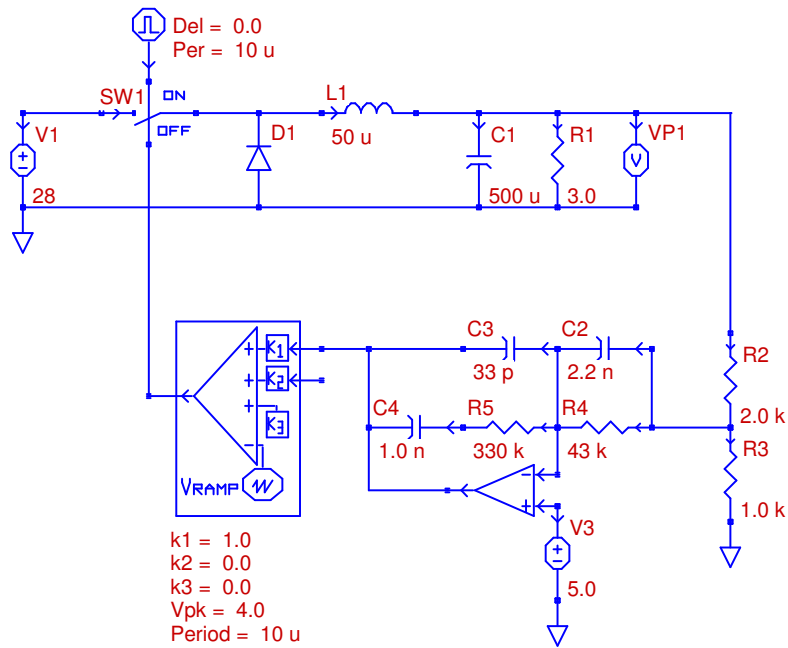


Figure 8.34: PECS Schematic of Lead Compensated System with Zero at 500Hz

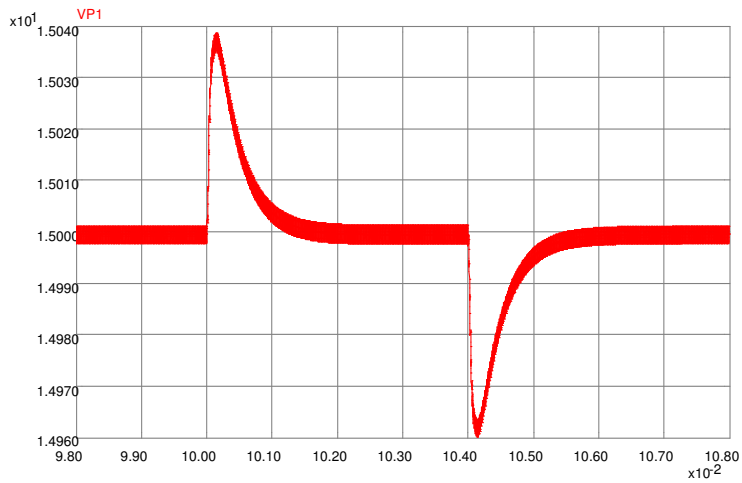


Figure 8.35: PECS Simulation of Lead Compensated System with Zero at 500Hz

8.36. The final magnitude and phase asymptotic plots are given in Fig. 8.37.

The new  $f_I$  is now found to be from (8.33)

$$f_I = f_1 T_o = 150 \times 3.4 = 351 \quad (8.33)$$

Using the new values of  $f_I = 351$  and  $f_1 = 150$ , a more precise value of crossover frequency and phase margin is found from Matlab to be 5,350Hz and  $54.3^\circ$ , respectively, as seen in Fig. 8.38. Recall that the asymptotic plots indicate 5kHz and  $45^\circ$ , respectively.

The change of  $f_1 = 150\text{Hz}$  to  $f_1 = 150\text{Hz}$  results in only a change in one capacitor value in the compensator. The resulting PECS implementation is shown in Fig. 8.39 along with the response of input voltage steps of  $28V \rightarrow 30V \rightarrow 28V$ , in Fig. 8.40. We now see that the peak voltage variation has slightly increased to 90 mV but the settling time has tripled to around 3ms.

## 8.8 Extended Bandwidth Design

In the following we examine the performance of a compensator (closely related to the previous two) which is designed to produce an extended loop bandwidth. To this end a unity gain crossover frequency  $f_c = 40\text{kHz}$  is, somewhat arbitrarily, chosen. The compensator used is

$$G_c(s) = \frac{\omega_I \left(1 + \frac{s}{\omega_{z_1}}\right) \left(1 + \frac{s}{\omega_{z_2}}\right)}{s} \quad (8.34)$$

The zeros  $f_{z_1} (= \frac{\omega_{z_1}}{2\pi})$  and  $f_{z_2} (= \frac{\omega_{z_2}}{2\pi})$  are simply chosen as follows. Zero  $f_{z_2}$  is set so  $f_{z_2} = f_o$  so as to counter the effects of the plant complex pole pair. The lower frequency zero  $f_{z_1}$  is set so that  $f_{z_1} = \frac{f_o}{10}$  to minimize the phase drop at  $f_o$ . The resulting loop gain expression is given by

$$T(s) = \omega_I T_o \frac{\left(1 + \frac{s}{\omega_{z_1}}\right) \left(1 + \frac{s}{\omega_{z_2}}\right)}{s \left[1 + \frac{s}{Q\omega_o} + \left(\frac{s}{\omega_o}\right)^2\right]} \quad (8.35)$$

The asymptotic magnitude and phase responses of the resulting loop gain are shown in Fig. 8.41, where the phase contributions of the different factors are individually drawn and then summed at the bottom plot to produce the overall asymptotic loop gain phase plot.

To determine the quantity  $\omega_I (= 2\pi f_I)$  in (8.34) the high frequency asymptotes of the magnitude plot is used. At the crossover frequency  $f_c$  we have

$$\frac{T_o f_I f_o}{f_{z_1} f_c} = 1 \quad (8.36)$$



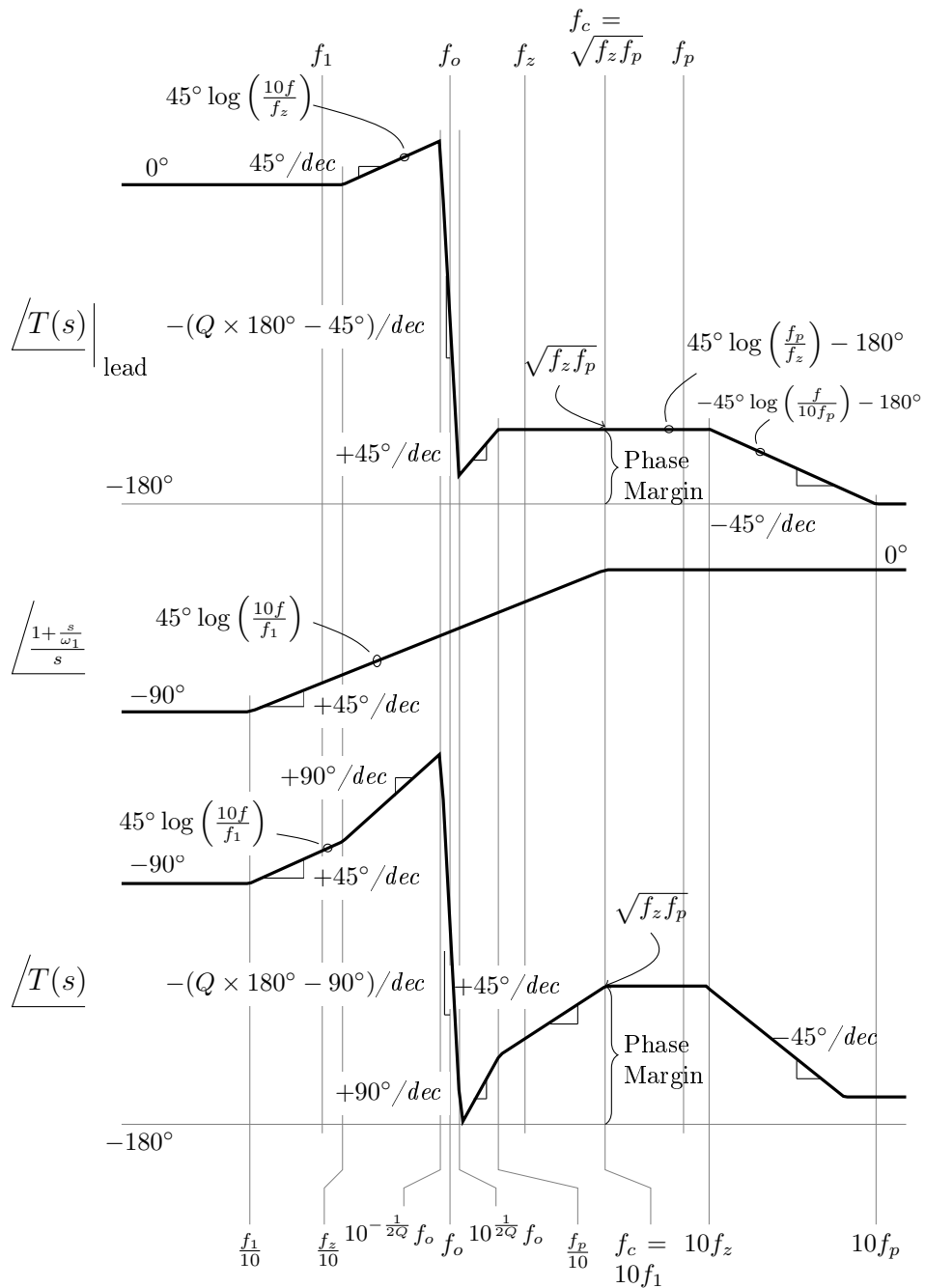


Figure 8.36: Bode Plot of Lead Compensator (150Hz)

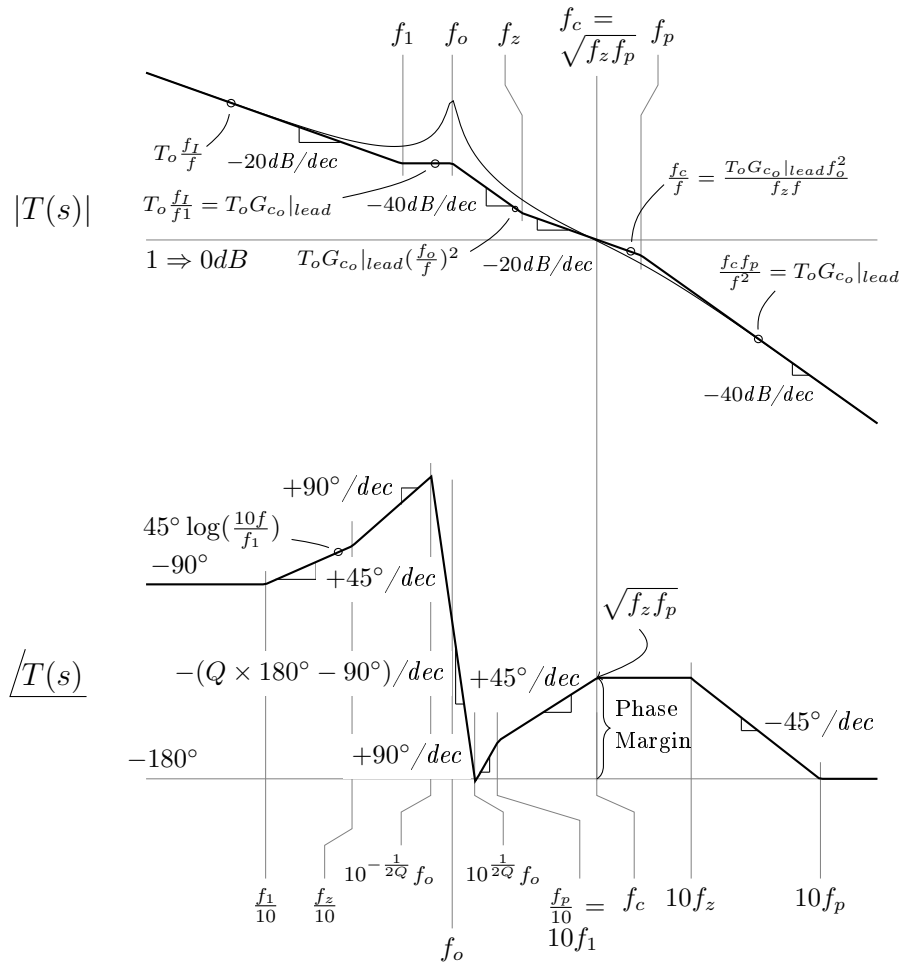


Figure 8.37: Bode Plot of System with Lead plus Integral Compensation (150Hz)

so that we have

$$f_I = \frac{f_{z1} f_c}{T_o f_o} \quad (8.37)$$

with the values at hand we find

$$f_I = 172 \quad (8.38)$$

From the phase asymptotic plot of Fig. 8.41 we can clearly see that the expected phase margin is  $90^\circ$ . Using Matlab we more precisely find with the design values used  $f_c = 40\text{kHz}$  and phase margin is  $88.6^\circ$  as shown in Fig. 8.42.

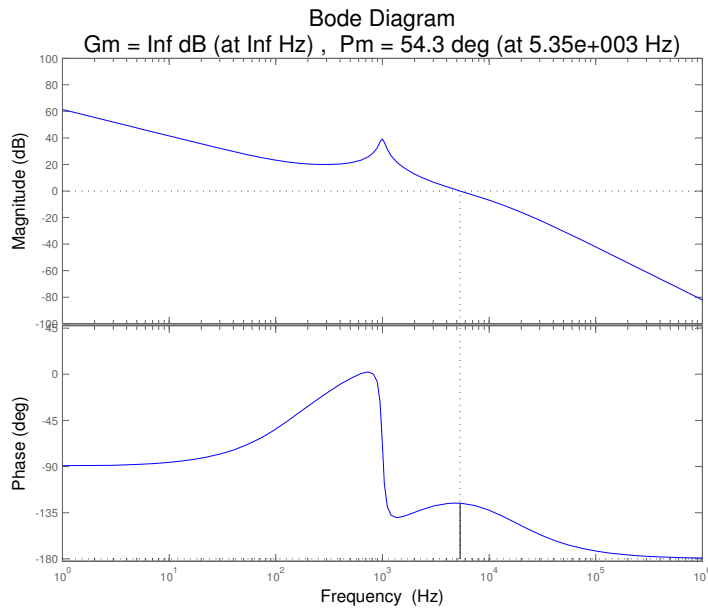


Figure 8.38: Matlab Lead Compensator with Integrator and Zero at 150Hz

The resulting PECS implementation is shown in Fig. 8.43 along with the response of input voltage steps of  $28V \rightarrow 30V \rightarrow 28V$ , in Fig. 8.44. We now see that the peak voltage variation has greatly reduced to just 30mV.

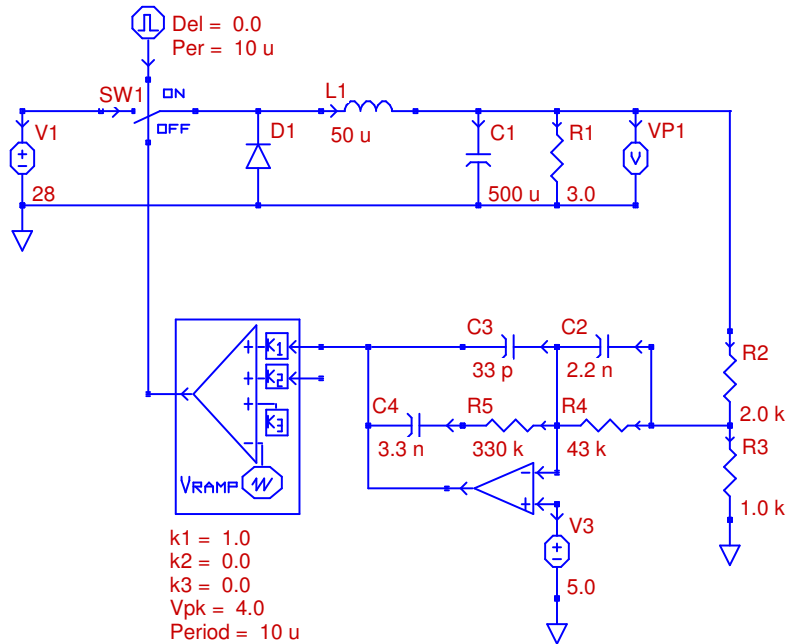


Figure 8.39: PECS Schematic of Lead Compensated System with Zero at 150Hz

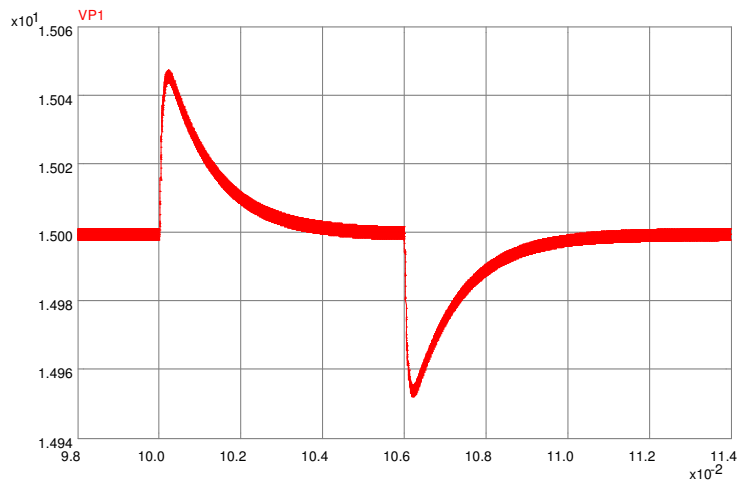


Figure 8.40: PECS Simulation of Lead Compensated System with Zero at 150Hz

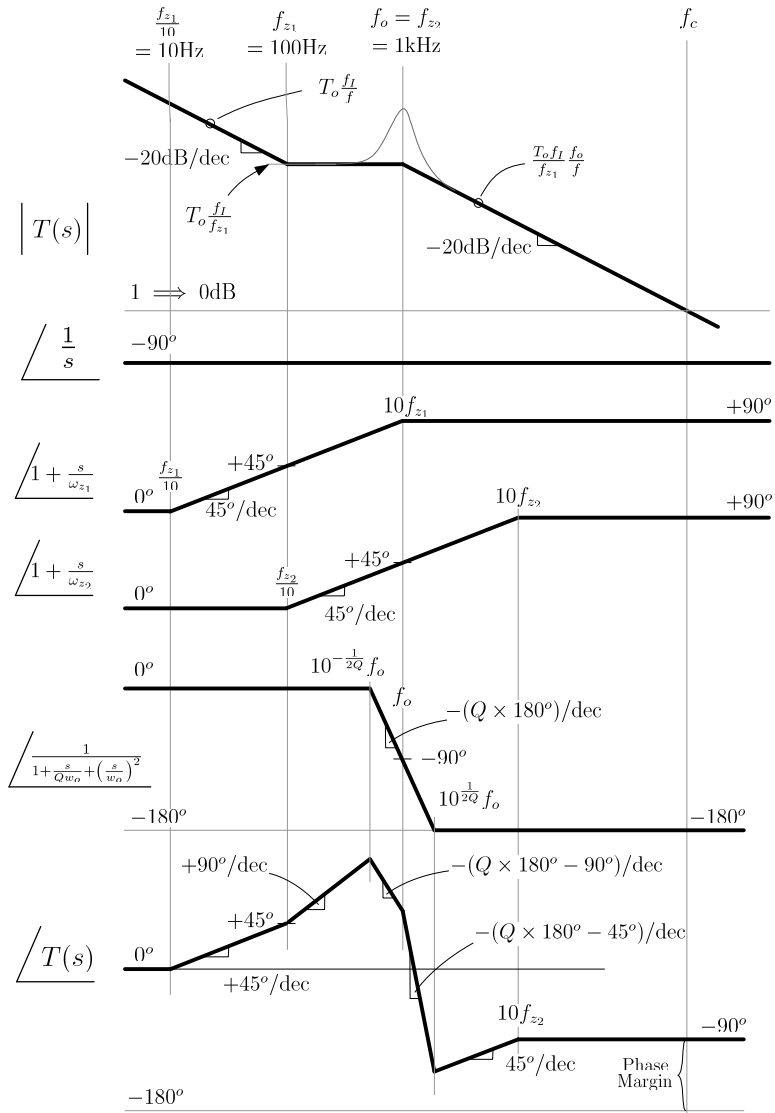


Figure 8.41: Extended Bandwidth Bode Plot and Phase Construction

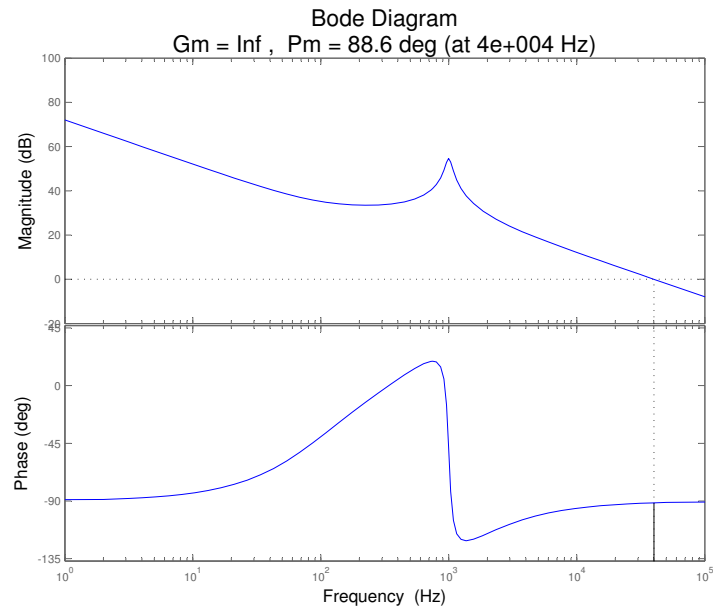


Figure 8.42: Matlab Analysis of Extended Compensator

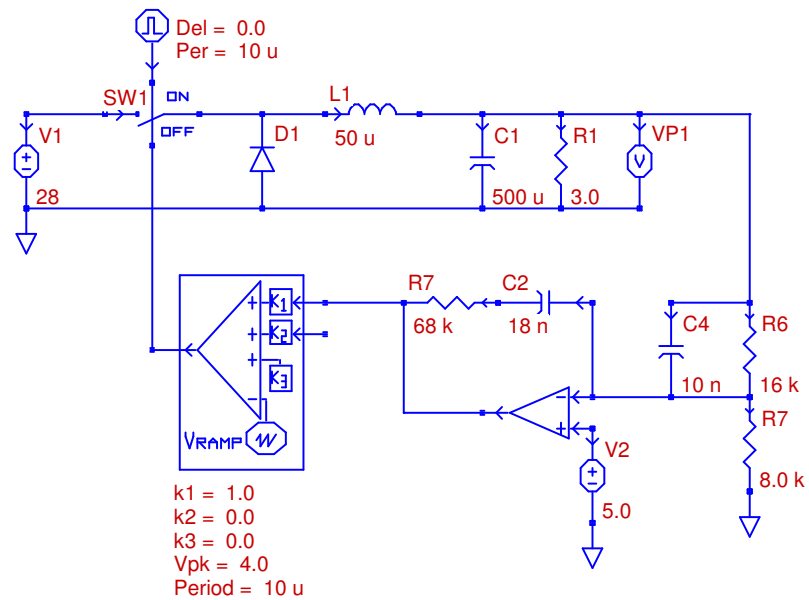


Figure 8.43: PECS Schematic of Extended System

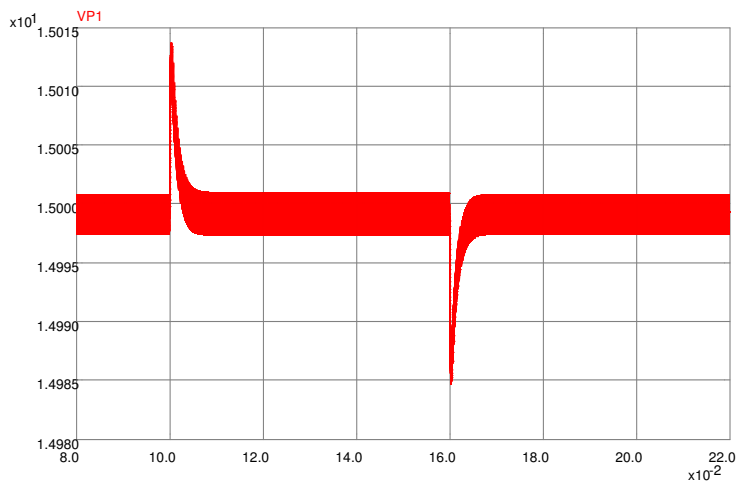


Figure 8.44: PECS Simulation of Extended System

## 8.9 Conclusion

The following table shows the summary of all of the results.



Table 8.2: Summary of Compensators

$$\text{Loop Gain } T(s) = G_c(s) \frac{T_o}{1 + \frac{s}{Q\omega_o} + (\frac{s}{\omega_o})^2},$$

where  $T_o = 2.33$ ,  $Q = 9.5$ , and  $\omega_o = 2\pi(1kHz)$

Compensator Design	Compensator Transfer Function $G_c(s)$	$\phi_M$ Asymptote (Matlab) (degrees)	$f_c$ Asymptote (Matlab) (kHz)	$\Delta v$ (mV)
Uncompensated Open Loop	$G_{c_o}$ $G_{c_o} = 1$	0 (5)	1.5 (1.82)	2,800
Dominant Pole (3dB gain margin)	$\frac{\omega_I}{s}$ $\omega_I = 2\pi(32)$	90 (90)	0.0744 (0.0746)	3,700
Dominant Pole + Zero (10db gain margin)	$\frac{\omega_I(1 + \frac{s}{\omega_1})}{s}$ $\omega_I = 2\pi(14.3)$ $\omega_1 = 2\pi(1,000)$	90 (92)	0.0333 (0.033)	3,700
Lead	$G_{c_o} \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{\omega_p}}$ $G_{c_o} = 3.4$ $\omega_z = 2\pi(1,580)$ $\omega_p = 2\pi(15,800)$	45 (56)	5.0 (5.35)	120
Lead + Integrator + Zero at 500Hz	$\frac{\omega_I(1 + \frac{s}{\omega_1})(1 + \frac{s}{\omega_z})}{s(1 + \frac{s}{\omega_p})}$ $\omega_I = 2\pi(1,770)$ $\omega_1 = 2\pi(500)$ $\omega_z = 2\pi(1,580)$ $\omega_p = 2\pi(15,800)$	45 (51)	5.0 (5.37)	80
Lead + Integrator + Zero at 150Hz	$\frac{\omega_I(1 + \frac{s}{\omega_1})(1 + \frac{s}{\omega_z})}{s(1 + \frac{s}{\omega_p})}$ $\omega_I = 2\pi(351)$ $\omega_1 = 2\pi(150)$ $\omega_z = 2\pi(1,580)$ $\omega_p = 2\pi(15,800)$	45 (54)	5.0 (5.35)	90
Extended Bandwidth	$\frac{\omega_I(1 + \frac{s}{\omega_{z1}})(1 + \frac{s}{\omega_{z2}})}{s}$ $\omega_I = 2\pi(172)$ $\omega_{z1} = 2\pi(100)$ $\omega_{z2} = 2\pi(1000)$	90 (89)	40 (40)	30



# References

- [1] R. W. Erickson and D. Maksimović, *Fundamentals of Power Electronics*, 2nd ed. New York, NY: Springer Science+Business Media, LLC, 2001.
- [2] R. Tymerski, “PECS Simulator ©1999-2009,” Portland, OR.
- [3] MATLAB, *R2012b*. Natick, MA: The Mathworks, Inc., 2012.

## Appendix

### 8.9.1 Compensator Circuits

	Control Action	$G(s) = \frac{E_o(s)}{E_i(s)}$	Operational Amplifier Circuits
1	P Proportional	$-A$ where: $A = \frac{R_2}{R_1}$	
2	I Integral	$-\frac{1}{s/\omega_0}$ where: $\omega_0 = \frac{1}{R_1 C_1}$	
3	PD Proportional-Derivative	$-A(1 + \frac{s}{\omega_1})$ where: $A = \frac{R_2}{R_1}, \quad \omega_1 = \frac{1}{R_1 C_1}$	
4	PI Proportional-Integral	$-A \frac{1 + \frac{s}{\omega_1}}{s}$ where: $A = \frac{R_2}{R_1}, \quad \omega_1 = \frac{1}{R_2 C_1}$	
5	PID Proportional-Integral-Derivative	$-A \frac{(1 + \frac{s}{\omega_1})(1 + \frac{s}{\omega_2})}{s}$ where: $A = \frac{R_2}{R_1}, \quad \omega_1 = \frac{1}{R_1 C_1}, \quad \omega_2 = \frac{1}{R_2 C_2}$	
6	Lead if $\omega_1 < \omega_2$ or Lag if $\omega_1 > \omega_2$	$-A \frac{1 + \frac{s}{\omega_1}}{1 + \frac{s}{\omega_2}}$ where: $A = \frac{R_2}{R_1}, \quad \omega_1 = \frac{1}{R_1 C_1}, \quad \omega_2 = \frac{1}{R_2 C_2}$	

	Control Action	$G(s) = \frac{E_o(s)}{E_i(s)}$	Operational Amplifier Circuits
7	Lead-Lag	$-A \frac{(1 + \frac{s}{\omega_1})(1 + \frac{s}{\omega_2})}{(1 + \frac{s}{\omega_3})(1 + \frac{s}{\omega_4})}$ where: $A = \frac{R_4}{R_3},$ $\omega_1 = \frac{1}{(R_1 + R_3)C_1}, \quad \omega_2 = \frac{1}{R_2 C_2}$ $\omega_3 = \frac{1}{R_1 C_1}, \quad \omega_4 = \frac{1}{(R_2 + R_4)C_2}$	
8	Integrator with Lead or Lag	where: $-\frac{\omega_0 (1 + \frac{s}{\omega_1})}{s (1 + \frac{s}{\omega_2})}$ $\omega_0 = \frac{1}{R_1(C_1 + C_2)}$ $\omega_1 = \frac{1}{R_2 C_2}$ $\omega_2 = \frac{1}{R_2 \frac{C_1 C_2}{C_1 + C_2}}$	
9	Integrator with Lead	where: $-\frac{\omega_0 (1 + \frac{s}{\omega_1})(1 + \frac{s}{\omega_2})}{s (1 + \frac{s}{\omega_3})}$ $\omega_0 = \frac{1}{R_1(C_2 + C_3)}$ $\omega_1 = \frac{1}{R_2 C_2}, \quad \omega_2 = \frac{1}{R_1 C_1}$ $\omega_3 = \frac{1}{R_2 \frac{C_2 C_3}{C_2 + C_3}}$	
10	Integrator with Lead-Lag	where: $-\frac{\omega_0 (1 + \frac{s}{\omega_1})(1 + \frac{s}{\omega_2})}{s (1 + \frac{s}{\omega_3})(1 + \frac{s}{\omega_4})}$ $\omega_0 = \frac{1}{R_1(C_2 + C_3)}$ $\omega_1 = \frac{1}{R_2 C_2}, \quad \omega_2 = \frac{1}{C_1(R_1 + R_3)},$ $\omega_3 = \frac{1}{R_3 C_1}, \quad \omega_4 = \frac{1}{R_2 \frac{C_2 C_3}{C_2 + C_3}}$	

## 8.10 MATLAB Code

```
1 clear all
2 close all
3
4 f0 = 1000;
5 Q = 9.5;
6 T0 = 2.33;
7 w0 = 2*pi*f0;
8
9 s = tf('s');
10 Ts = T0/((s/w0)^2 + s/(Q*w0) + 1);
11
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 % Open loop
14
15 figure(1)
16 margin(Ts)
17 h = gcr;
18 h.AxesGrid.Xunits = 'Hz';
19 h.AxesGrid.TitleStyle.FontSize = 16;
20 h.AxesGrid.XLabelStyle.FontSize = 12;
21 h.AxesGrid.YLabelStyle.FontSize = 12;
22
23 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
24 % Lead compensation
25
26 Gc0 = 3.4;
27 wz = 2*pi*1500;
28 wp = 2*pi*15000;
29
30 Gc1 = Gc0*(1+s/wz)/(1+s/wp);
31
32 Ts1 = Gc1*Ts;
33
34 figure(2)
35 margin(Ts1)
36 h = gcr;
37 h.AxesGrid.Xunits = 'Hz';
38 h.AxesGrid.TitleStyle.FontSize = 16;
39 h.AxesGrid.XLabelStyle.FontSize = 12;
40 h.AxesGrid.YLabelStyle.FontSize = 12;
41
42
43 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
44 % 500
45
46 wi = Gc0*2*pi*500;
47 wz2 = 2*pi*500;
48 wz = 2*pi*1500;
49 wp = 2*pi*15000;
50
51 Gc2 = wi/s*(1+s/wz2)*(1+s/wz)/(1+s/wp);
52 Ts2 = Gc2*Ts;
53
```

```

54 figure(3)
55 margin(Ts2)
56 h = gcr;
57 h.AxesGrid.Xunits = 'Hz';
58 h.AxesGrid.TitleStyle.FontSize = 16;
59 h.AxesGrid.XLabelStyle.FontSize = 12;
60 h.AxesGrid.YLabelStyle.FontSize = 12;
61
62
63
64
65
66 %%%%%%%%%%%
67 % 150
68
69 wi = Gc0*2*pi*150;
70 wz2 = 2*pi*150;
71 wz = 2*pi*1500;
72 wp = 2*pi*15000;
73
74 Gc3 =wi/s*(1+s/wz2)*(1+s/wz)/(1+s/wp);
75 Ts3 = Gc3*Ts;
76
77 figure(4)
78 margin(Ts3)
79 h = gcr;
80 h.AxesGrid.Xunits = 'Hz';
81 h.AxesGrid.TitleStyle.FontSize = 16;
82 h.AxesGrid.XLabelStyle.FontSize = 12;
83 h.AxesGrid.YLabelStyle.FontSize = 12;
84
85
86 %%%%%%%%%%%
87 % Extended bandwidth
88
89 %wi = Gc0*2*pi*100;
90 wi = 2*pi*100*40000/(T0*1000); % fc = 40000
91 wz1 = 2*pi*100;
92 wz2 = 2*pi*1000;
93
94 Gc4 = wi/s*(1+s/wz2)*(1+s/wz1);
95 Ts4 = Gc4*Ts;
96
97 figure(5)
98 margin(Ts4)
99 h = gcr;
100 h.AxesGrid.Xunits = 'Hz';
101 h.AxesGrid.TitleStyle.FontSize = 16;
102 h.AxesGrid.XLabelStyle.FontSize = 12;
103 h.AxesGrid.YLabelStyle.FontSize = 12;
104
105 %%%%%%%%%%%
106 % Dominant pole
107
108 Gcd0 = 200;
109
110 Gc5 = Gcd0/s;

```

```

111
112 Ts5 = Gc5*Ts;
113
114 figure(6)
115 margin(Ts5)
116 h = gcr;
117 h.AxesGrid.Xunits = 'Hz';
118 h.AxesGrid.TitleStyle.FontSize = 16;
119 h.AxesGrid.XLabelStyle.FontSize = 12;
120 h.AxesGrid.YLabelStyle.FontSize = 12;
121
122 %%%%%%%%%%%%%%%
123 % Dominant pole with zero
124
125 Gcdz0 = 1/sqrt(10)*w0/(T0*Q); % 89.76
126 wz = 2*pi*1000;
127
128 Gc6 = Gcdz0*(1+s/wz)/s;
129
130 Ts6 = Gc6*Ts;
131
132 figure(7)
133 margin(Ts6)
134 h = gcr;
135 h.AxesGrid.Xunits = 'Hz';
136 h.AxesGrid.TitleStyle.FontSize = 16;
137 h.AxesGrid.XLabelStyle.FontSize = 12;
138 h.AxesGrid.YLabelStyle.FontSize = 12;

```



## Chapter 9

# Droop and Multi-Loop Control

## Abstract

Constant output voltage is an important feature of a DC voltage regulator. This paper describes three compensation techniques including voltage droop, inductor current droop, and voltage compensation to minimize voltage deviation to changes in the output load of the system. Through simulation it is confirmed that droop compensation improves voltage deviation by a factor of two in comparison to traditional voltage compensation.

## 9.1 Introduction

In DC-DC voltage regulators, it is important to supply a constant voltage, regardless of the current load on the output. The goal of this paper is to describe three feedback compensation techniques for the Buck converter to limit the voltage deviation in response to a current step on the output. The designs to be implemented include a voltage droop compensator [1], an inductor current compensator [2], and a conventional voltage compensation circuit. For this design, resistive losses will be included for the inductor and the capacitor to provide a more complete and accurate analysis.

## 9.2 Design

### 9.2.1 Passive Droop Compensation

#### Concept of droop control

As discussed in [1], the basic concept behind droop control is to apply compensation to the Buck converter in a way that creates a constant, closed-loop output impedance. By creating a constant output impedance, any variation in load current will result in a change in output voltage to maintain a constant impedance. Knowing the maximum load current requirements, the maximum droop of the system is simply  $\Delta V = \Delta IR$ .

Reviewing the output section of the buck converter including losses, it can be seen that the open loop output impedance at high frequencies is equal to the parasitic resistance of the output capacitor. For this design, the parasitic resistance of the capacitor will be utilized as the value of the output impedance for the compensated Buck converter.

#### Derivation of the closed-loop output impedance

As shown in Figure 9.4, the output voltage variation of a Buck converter is determined based on the variation of the duty cycle, variations in the source

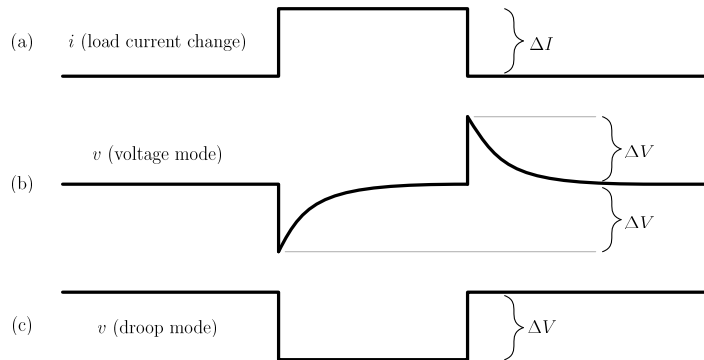


Figure 9.1: Transient Response of System with Droop Control

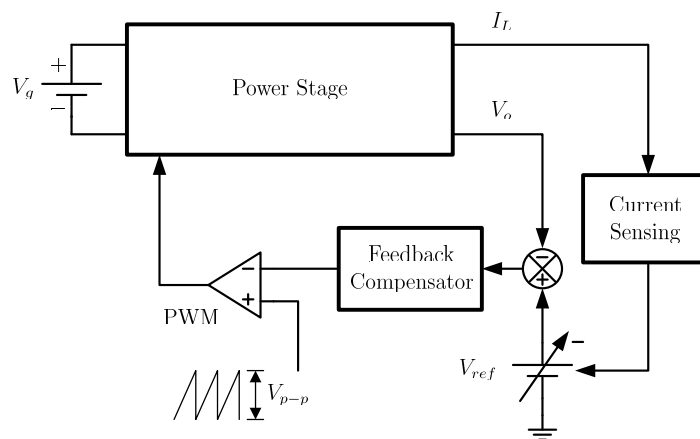


Figure 9.2: Active Droop Control System Block Diagram

voltage, and variations in the load current. For this example, input voltage variations will be neglected.

By definition,

$$Z_{oc}(s) = \frac{Z_o(s)}{1 + T(s)} \quad (9.1)$$

Following Figure 9.4,  $T(s) = F_m H G_{vd}(s) G_{con}(s)$ , where  $F_m$  is the PWM comparator effect,  $G_{vd}(s)$  is the control to output transfer function,  $H$  is the feedback attenuation, and  $G_{con}(s)$  is the compensation block to be designed. To find  $Z_{oc}(s)$  it is necessary to first calculate  $Z_o(s)$  (the open loop output

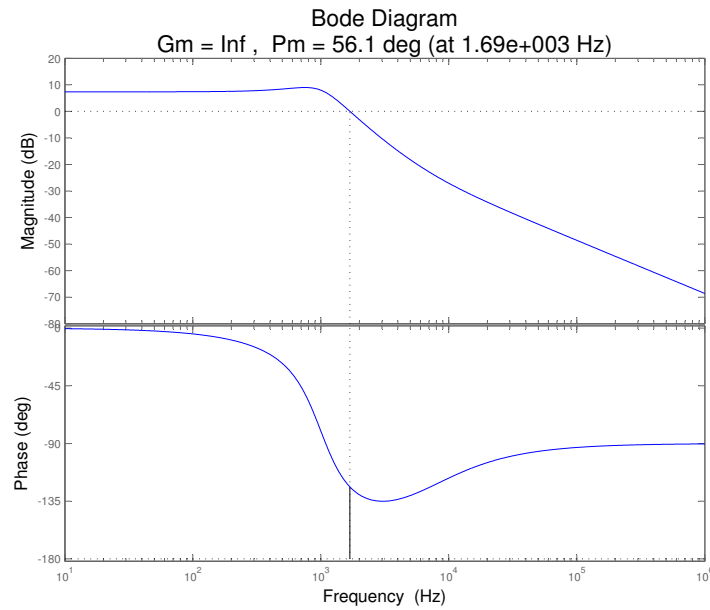


Figure 9.3: MATLAB Uncompensated Bode plot

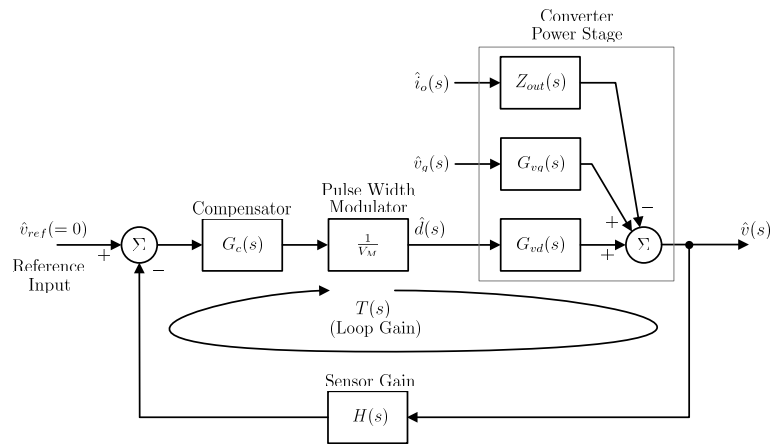


Figure 9.4: Buck converter control loop

impedance) and  $G_{vd}(s)$ . Applying state space averaging methods to calculate the small signal model of  $Z_o(s)$  and  $G_{vd}(s)$ , the functions are realized as shown below.

$$Z_o(s) = r_l \frac{(1 + \frac{s}{\omega_c})(1 + \frac{s}{\omega_l})}{1 + \frac{s}{Q\omega_o} + \left(\frac{s}{\omega_o}\right)^2} \quad (9.2)$$

$$G_{vd}(s) = V_g \frac{1 + \frac{s}{\omega_c}}{1 + \frac{s}{Q\omega_o} + \left(\frac{s}{\omega_o}\right)^2} \quad (9.3)$$

Where  $\omega_o \approx \frac{1}{\sqrt{LC}}$ ,  $\omega_c = \frac{1}{r_c C}$ ,  $\omega_l = \frac{r_l}{L}$ ,  $\omega_R = \frac{1}{RC}$ , and  $Q \approx \frac{\sqrt{LC}}{r_l + r_c}$ .

Recalling that our aim is to derive a compensator circuit that will implement a constant output impedance equal to the parasitic resistance of the capacitor,  $Z_{oc}$  is set equal to  $r_c$ .

$$Z_{oc}(s) = \frac{Z_o(s)}{1 + T(s)} = \frac{Z_o(s)}{1 + F_m G_{vd}(s) G_{con}(s)} = r_c$$

Expanding  $Z_o(s)$  and  $G_{vd}(s)$ :

$$r_c = \frac{r_l \frac{(1 + \frac{s}{\omega_l})(1 + \frac{s}{\omega_c})}{1 + \frac{s}{Q\omega_o} + \frac{s^2}{\omega_o^2}}}{1 + \frac{F_m(1 + \frac{s}{\omega_c})V_g}{1 + \frac{s}{Q\omega_o} + \frac{s^2}{\omega_o^2}} G_{con}(s)}$$

Rearranging the equation,

$$r_c F_m V_g \left(1 + \frac{s}{\omega_c}\right) G_{con}(s) = r_l - r_c + \left[\left(\frac{1}{\omega_l} + \frac{1}{\omega_c}\right)r_l - \frac{r_c}{Q\omega_o}\right]s + \left(\frac{r_l}{\omega_l \omega_c} - \frac{r_l}{\omega_o^2}\right)s^2$$

Expanding the  $s^2$  term,

$$\frac{r_l}{\omega_l \omega_c} - \frac{r_c}{\omega_o^2} = \frac{r_l}{L} \frac{1}{r_c C} - \frac{r_c}{LC} = r_c(LC - LC) = 0$$

$G_{con}(s)$  can then be simplified into the form:

$$G_{con}(s) = K_v \frac{1 + \frac{s}{\omega_{zv}}}{1 + \frac{s}{\omega_{pv}}} \quad (9.4)$$

Where  $K_v = \frac{r_l - r_c}{r_c V_g F_m H}$ ,  $\omega_{pv} = \omega_c$ , and  $\omega_{zv} = \frac{R_l - R_c}{L - R_c^2 C}$ .

Using the circuit parameters defined in the introduction,  $K_v = .171$ ,  $\omega_{zv} = 4.1k \frac{\text{rad}}{\text{s}}$ , and  $\omega_{pv} = 40k \frac{\text{rad}}{\text{s}}$ . For a Matlab calculation of these values, please refer to the MATLAB code provided at the end of the chapter.

Figure 9.5 show the frequency response of the loop gain of the system. Note that in this compensation design, a small gain at low frequencies is implemented, which is contrary to typical feedback designs where nearly infinite DC gain is desired.

In Figures 9.6 and 9.7, the open and closed loop output impedance and the system audio susceptibility are shown as a function of frequency. Note that as expected, the closed loop output impedance is approximately a constant  $-26dB$  or  $50m\Omega$  over the frequency range.

With the compensator circuit now completely defined, it can be realized using an operational amplifier circuit. As seen in Figure 9.8, the circuit can be implemented using a single op-amp.

Since the loop gain of this system is very low, it is not safe to assume that a reference of 5 V ( $V_oH$ ) will work for this system. Looking at the block diagram, if the duty ratio is known, the output voltage of the amplifier is  $\frac{D}{F_m}$ . Using this value, the value of  $V_{ref}$  can be found by applying nodal analysis at  $V_-$ . Using the DC value of D ( $\frac{V}{V_g} = .53$ ),  $V_{ref}$  is found to be equal to approximately 3.9 V.

Through simulation of the droop circuit (Figure 9.8), it is found that the system performs exactly as expected for a step in current of 0.1 A to 5.0 A. With an output impedance of  $50m\Omega$ , the voltage change ideally would be equal to  $\Delta V = (4.9A)(50m\Omega) = .245V$ , which is approximately what is seen in Figure 9.8. In the next section, we will expand upon this design to incorporate additional compensation using the inductor current.

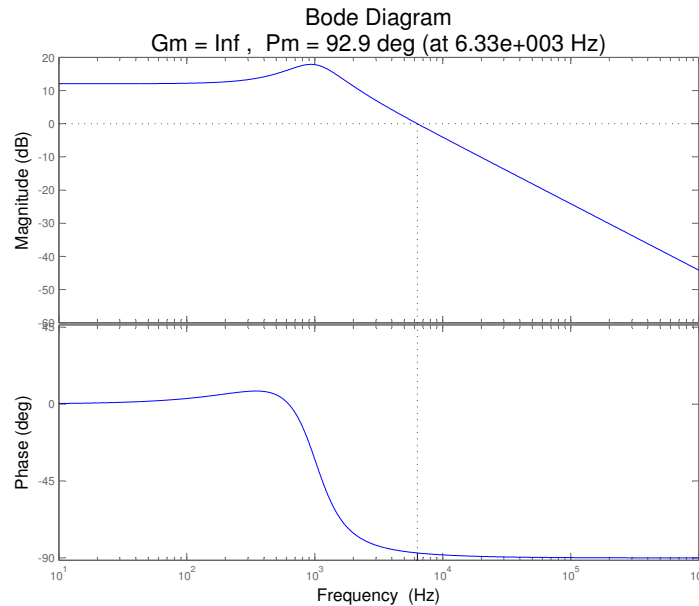


Figure 9.5: MATLAB Bode Plot of Loop Gain with Passive Droop Compensation

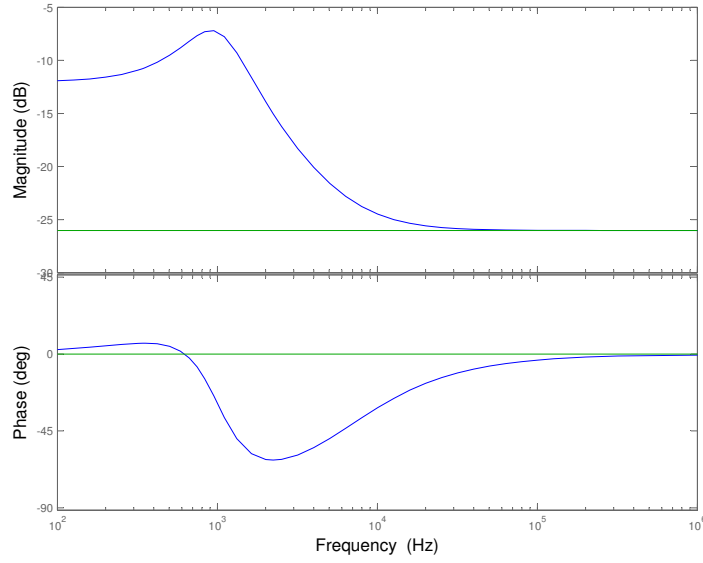


Figure 9.6: MATLAB Bode Plot of  $Z_o$  for Passive Droop Compensated System

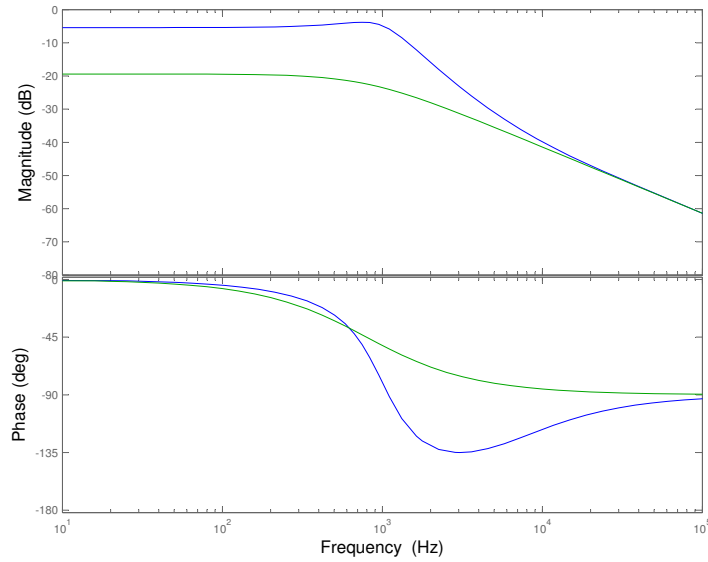


Figure 9.7: MATLAB Bode Plot of  $G_{vg}$  for Passive Droop Compensated System

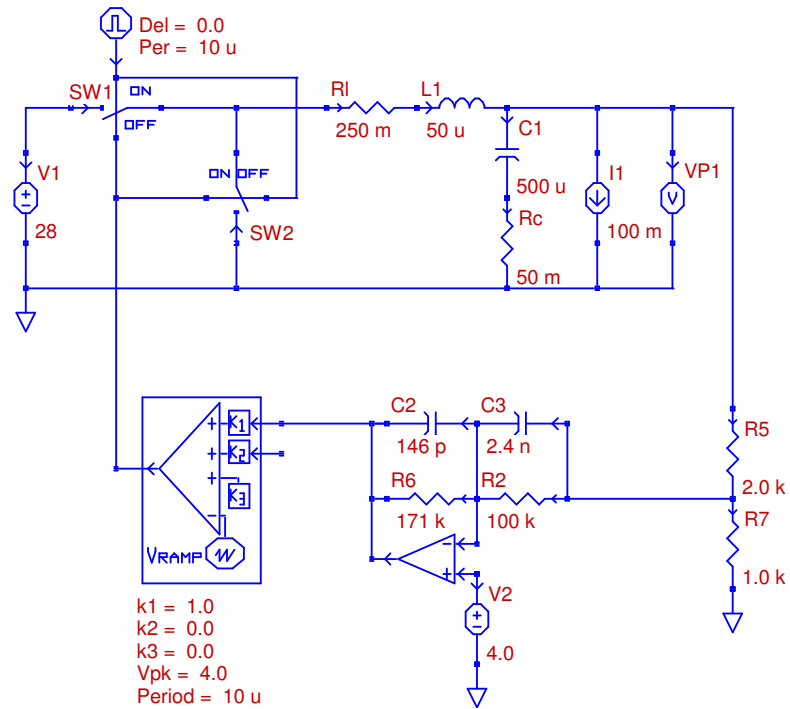


Figure 9.8: PECS Schematic of System with Passive Droop Compensation

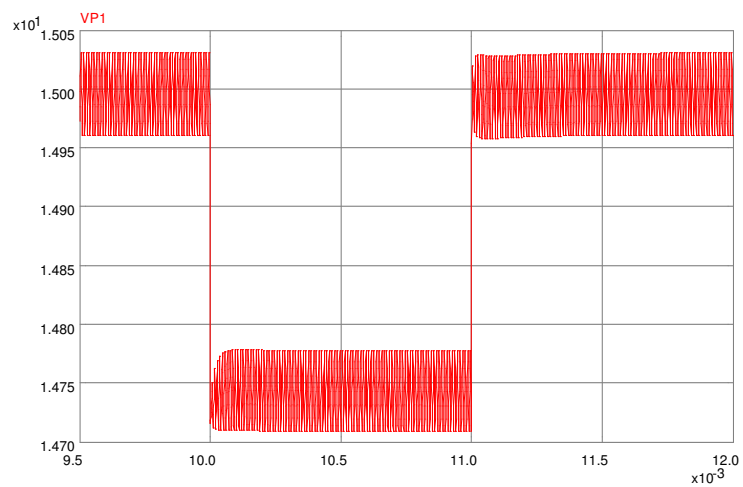


Figure 9.9: PECS Simulation of Passive Droop Compensation Response to a Load Current Step



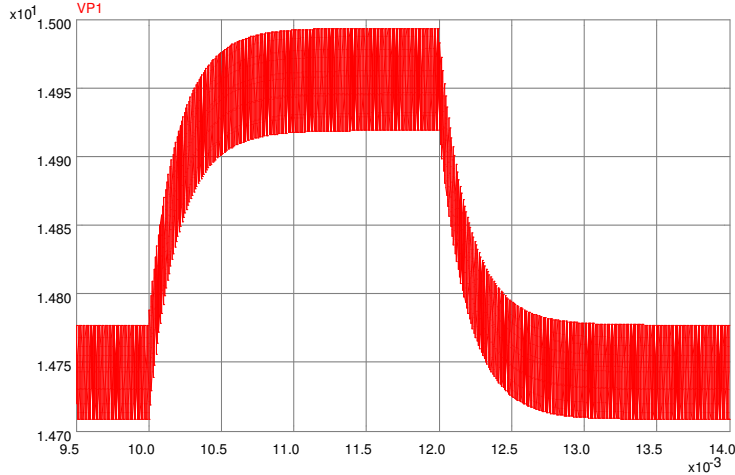


Figure 9.10: PECS Simulation of Passive Droop Compensation Response to a Supply Voltage Disturbance

## 9.2.2 Active Droop Compensation

### Concept of Current Sensing Droop Control

This mode of control implements a constant output impedance equal to the parasitic resistance of the capacitor, similar to that of the previous section. However, this feedback system is composed of two loops. One loop is the voltage loop that was used in the Section 1 design. The second loop is a current sensing loop, which measures the inductor current and provides a feedback voltage proportional to this current. The goal of this design as presented in [2] is to combine these two loops into one compensator to control the deviation in the output voltage. This can be seen in Figure 9.11.

Figure 9.12 shows the current and voltage loops in terms of transfer functions via small-signal block diagram.  $Z_o(s)$ ,  $G_{vd}(s)$ , and  $F_m$  are the same transfer functions used in the voltage droop mode.  $R_i$  is the inductor current sensing gain.  $A_v(s)$  is the transfer function of the feedback compensator, and is synonymous to  $G_{con}(s)$  in the voltage droop mode.  $G_{ii}(s)$  is the transfer function of inductor current to load current.  $G_{id}(s)$  is the transfer function of inductor current to duty cycle. Applying state space averaging methods to calculate the small signal model of  $G_{ii}(s)$  and  $G_{id}(s)$ (see Appendix 1), the functions are realized as shown below.

$$G_{ii}(s) = \frac{1 + \frac{s}{\omega_c}}{1 + \frac{s}{Q\omega_0} + \left(\frac{s}{\omega_0}\right)^2} \quad (9.5)$$

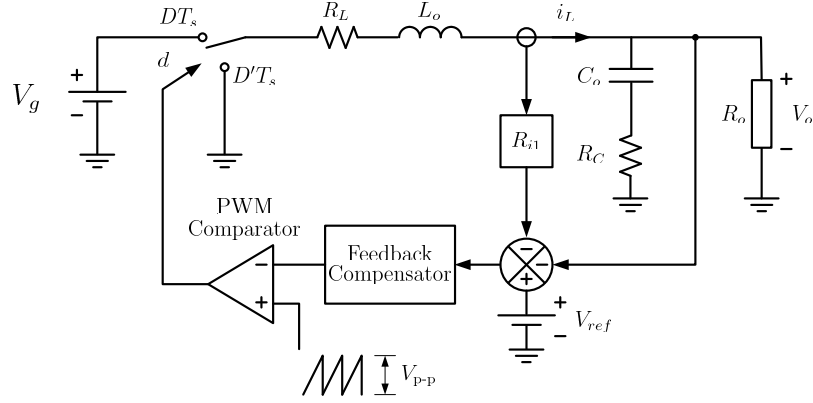


Figure 9.11: Buck converter with the current sensing mode [2]

$$G_{id}(s) = \frac{V_{in}}{R_o} \frac{1 + \frac{s}{\omega_R}}{1 + \frac{s}{Q\omega_0} + \left(\frac{s}{\omega_0}\right)^2} \quad (9.6)$$

where  $\omega_o \approx \frac{1}{\sqrt{CL}}$ ,  $\omega_c = \frac{1}{R_c C}$ ,  $\omega_l = \frac{R_l}{L}$ ,  $\omega_R = \frac{1}{RC}$ , and  $Q \approx \frac{\sqrt{LC}}{R_l + R_c}$ .

### Design of Current Sensing Droop Control

To create a feedback compensator  $A_v(s)$  for the current sensing droop control, the first step is realizing the functions for the current loop  $T_i(s)$  and the voltage loop  $T_v(s)$ . This can be achieved by following each loop in Figure 9.12. For a complete diagram including input voltage disturbances, refer to Figure 9.20.

$$T_i(s) = A_v(s)F_m G_{id}(s)R_i \quad (9.7)$$

$$T_v(s) = A_v(s)F_m G_{vd}(s) \quad (9.8)$$

In order to determine a single compensator  $A_v(s)$  for both loops,  $T_i(s)$  and  $T_v(s)$  need to be related. Because the compensator effects both loops, it is important to design the loops with the same characteristics. This allows both loops to be in unison and not work against each other. This can be achieved by designing both loops to have the same crossover frequency. This happens when the ratio of the two loops is set equal to 1.

$$\frac{T_v(s)}{T_i(s)} = \frac{G_{vd}(s)}{R_i G_{id}(s)} = \frac{R(1 + \frac{s}{\omega_c})}{R_i(1 + \frac{s}{\omega_R})} \quad (9.9)$$

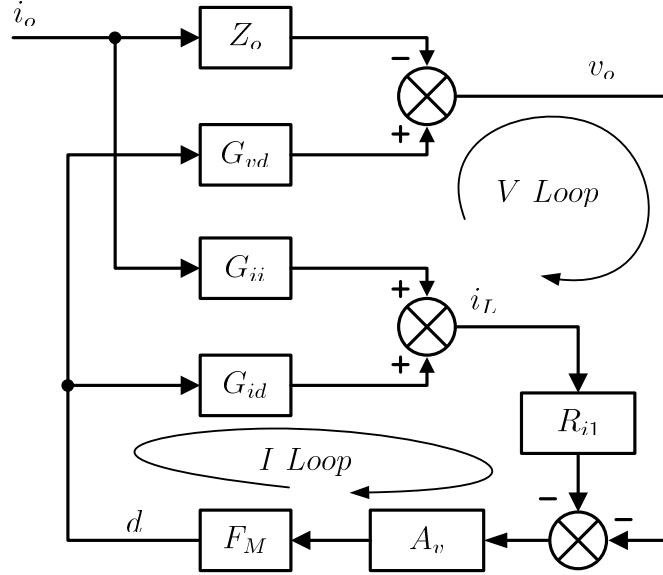


Figure 9.12: Small-signal block diagram of the current sensing mode

It can be seen that this ratio equals 1 when  $R_i = R_c$  and when  $\omega > \omega_c$ , since the zero from the output capacitor is normally much larger than  $\omega_R$ .

The design should be similar to the voltage droop control in terms of having a constant impedance gain equal to the parasitic losses of the output capacitor. The closed-loop output impedance function,  $Z_{oc}(s)$ , is shown below.

$$Z_{oc}(s) = \frac{Z_o(s)(1 + T_i(s)) + T_i(s) \frac{G_{vd}(s)G_{ii}(s)}{G_{id}(s)}}{1 + T_i(s) + T_v(s)} \quad (9.10)$$

Setting the above equation equal to the value of  $R_c$  the transfer function of  $A_v(s)$  can be derived. From [2], the current loop should be stable with a phase margin of around  $90^\circ$ . In order to achieve this, a zero,  $\omega_z$ , is needed to compensate for the power stage double pole. The high frequency switching noise should also be filtered. This requires a pole,  $\omega_p$ , placed well before the switching frequency. The current loop should have a crossover frequency that is higher than the parasitic zero of the output capacitor. This can be done by making the gain,  $\omega_i$ , of the compensator sufficiently large. Since infinite DC gain is required, an integrator is used by placing a pole at zero,  $\frac{1}{s}$ . The compensator transfer function can now be realized.

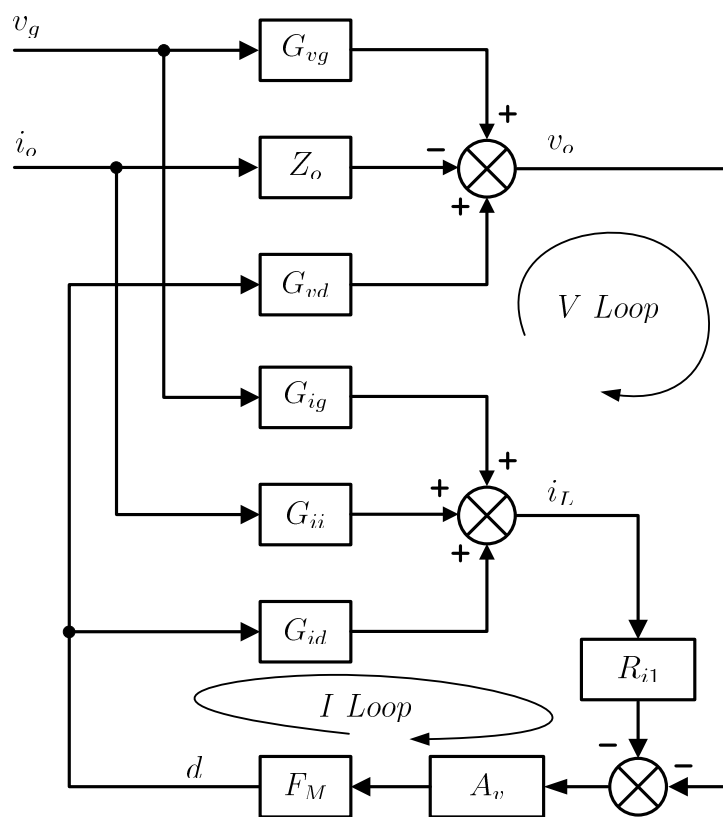


Figure 9.13: Small-signal block diagram of the current sensing mode -General

$$A_v(s) = \omega_i \frac{1 + \frac{s}{\omega_z}}{s(1 + \frac{s}{\omega_p})} \quad (9.11)$$

where  $\omega_z = \omega_o$ ,  $\omega_p = (2\pi)50,000$ , and  $\omega_i = 100,000$

The compensator can now be implemented using an operational amplifier circuit. As seen in Figure 9.17, this circuit can be implemented using a single op-amp. The summing of the two loops is incorporated into the op-amp used by the compensator. Please note the use of a voltage divider in the voltage feedback loop is similar to that in the previous section, only it has been scaled to a ratio of  $\frac{1}{2}$  for component value selection.

The loop gain of this system is a ratio of the current and voltage loops, and once again it's not safe to assume a reference of 5 V will work for this system.  $V_{ref}$  is found to be approximately 3.8 V in this design.

## Loop Gain Analysis

With the design stage complete, the system can be analyzed for stability and crossover frequency verification. The outer loop,  $T_2$ , shown below, determines the system stability.

$$T_2(s) = \frac{T_v(s)}{1 + T_i(s)} \quad (9.12)$$

It can be seen in Figure 9.14 that all three loops have the same crossover frequency. Stability is verified with  $T_2$  having a phase margin of about  $105^\circ$ . This is really close to the design of  $90^\circ$ , with other poles and zeros from  $T_v$  and  $T_i$  accounting for the slight difference.

It can be observed in Figure 9.15 that the impedance gain is nearly constant. The impedance phase is also nearly constant as it changes only slightly over the entire range of frequencies.

In Figure 9.18, the current sensing droop control was simulated with a step in the load current, the same as was performed in Section 1. The results are nearly the same as with the voltage droop control. There are slight transient spikes, but they are small and are on the order of the ripple. These spikes are caused from the slight deviation in the constant output impedance. With the output impedance at  $50m\Omega$ , the ideal voltage change would again be .245 V. Aside from the slight transient spikes, this is what is approximately seen in Figure 9.18. This two loop compensation control can achieve the same results as the simple voltage droop control, yet have the versatility to compensate for both voltage and current. In the next section, a voltage compensation scheme will be presented and compared to the droop control methods.

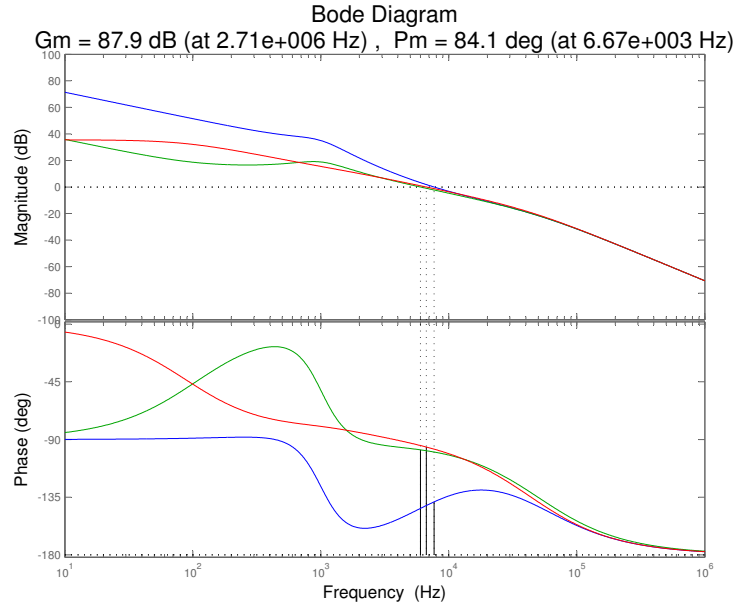


Figure 9.14: MATLAB Bode Plot of Loop Gains with Active Droop Compensation

### 9.2.3 Voltage Mode Compensation

#### Concept of voltage compensation

The final design that will be discussed in this report is a traditional voltage compensation design. Back in Figure 9.4, we found for the purposes of droop control that we would design for changes in output current and neglect any disturbances in the input voltage. For this design, the compensation circuit will be designed to reduce the audio susceptibility of the circuit ( $G_{cl}(s) = \frac{\hat{v}}{\hat{v}_g}$ ). This design will then be tested by applying a step disturbance to the load current, and determining the maximum deviation of the output voltage in response to the disturbance.

By following the loop in Figure 9.4, it can be seen by inspection that  $G_{cl}(s) = \frac{G_{vg}(s)}{1+T(s)}$ , where  $T(s) = G_{vd}(s)(Fm)G_{con}(s)$  and  $G_{vd}(s)$  is the same as in Section 1 and Section 2.  $G_{vg}(s)$  can be found to equal the following expression:

$$G_{vg}(s) = D \frac{1 + \frac{s}{\omega_c}}{1 + \frac{s}{Q\omega_0} + \left(\frac{s}{\omega_0}\right)^2} \quad (9.13)$$

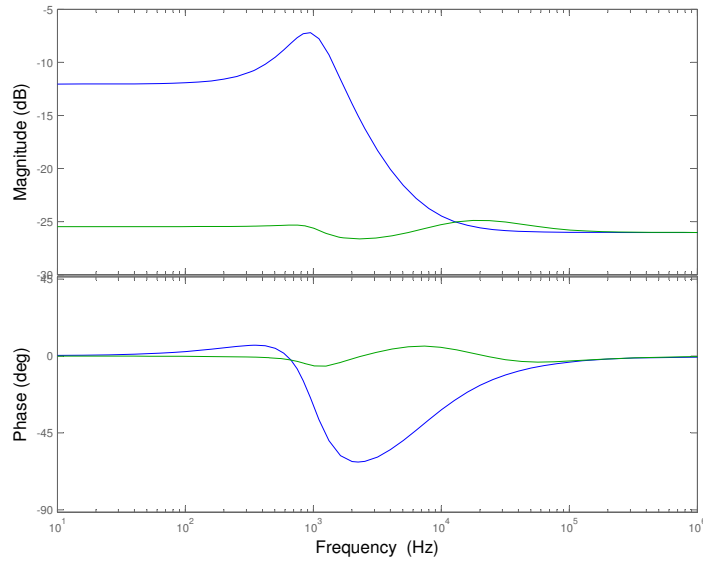


Figure 9.15: MATLAB Bode Plot of  $Z_o$  for Active Droop Compensated System

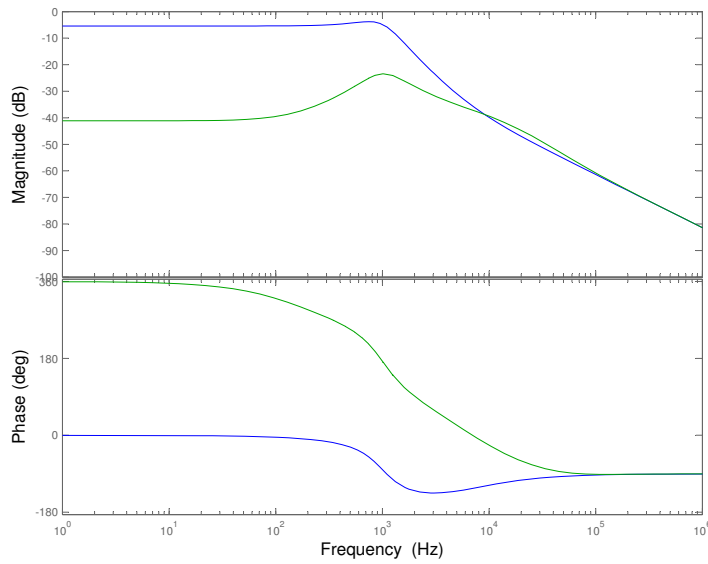


Figure 9.16: MATLAB Bode Plot of  $G_{vg}$  for Active Droop Compensated System

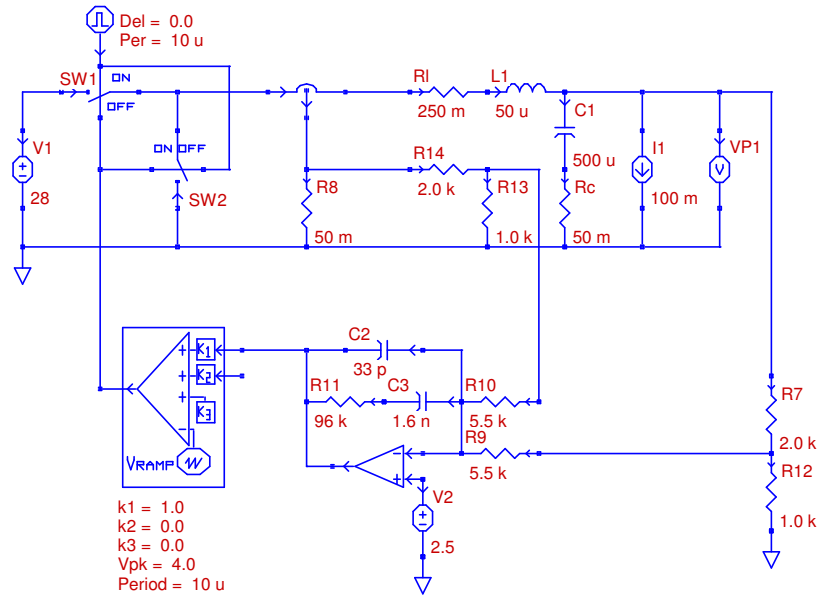


Figure 9.17: PECS Schematic of System with Active Droop Compensation

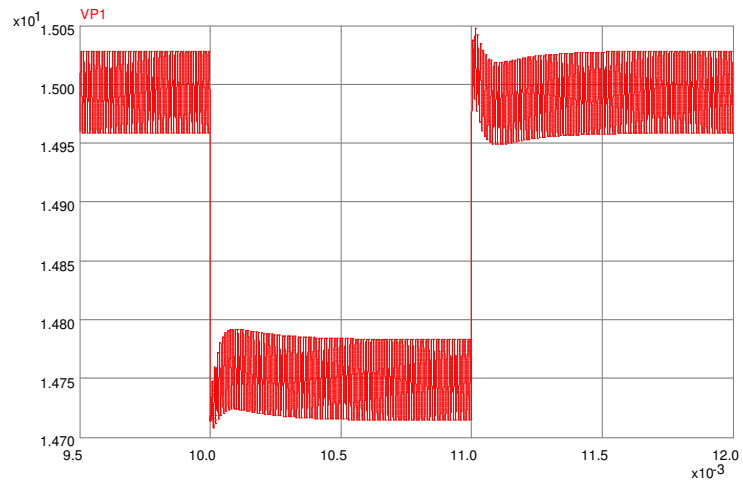


Figure 9.18: PECS Simulation of Active Droop Compensated Response to a Load Current Step



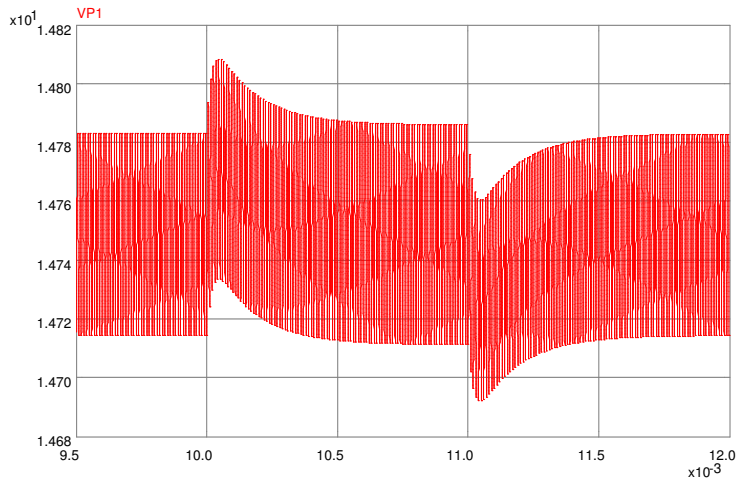


Figure 9.19: PECS Simulation of Active Droop Compensated Response to a Supply Voltage Disturbance

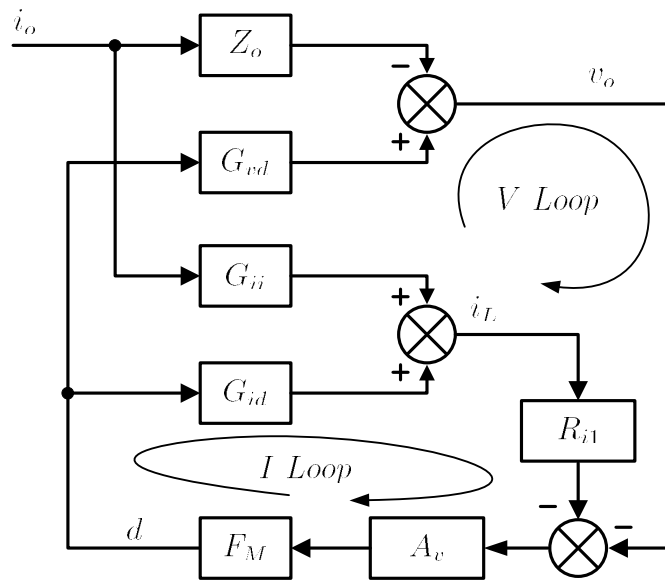


Figure 9.20: Complete Small-signal block diagram of the current sensing mode

Where all parameters of the equation are the same as described in the previous sections.

### Designing loop compensation

The goal of this design is to create a compensation block that will maximally reduce the effect of input voltage disturbances on the output. To improve on the basic integrator compensator design, two zeros are added around the crossover frequency to combat the extra  $-90^\circ$  of phase shift introduced by the integrator. Two poles are also introduced to ensure that the switching frequency noise will be canceled with the introduction of the zero. The equation for this compensation block is shown below:

$$G_{con}(s) = \frac{\omega_I \left(1 + \frac{s}{\omega_{z1}}\right) \left(1 + \frac{s}{\omega_{z2}}\right)}{s \left(1 + \frac{s}{\omega_{p1}}\right) \left(1 + \frac{s}{\omega_{p2}}\right)} \quad (9.14)$$

where  $\omega_I$  is the constant gain and  $\omega_{z1}$ ,  $\omega_{z2}$ ,  $\omega_{p1}$ , and  $\omega_{p2}$  are the new parameters to be designed.

To gain a better understanding of the system, the system transfer functions are plotted for three different cases of zero placement:

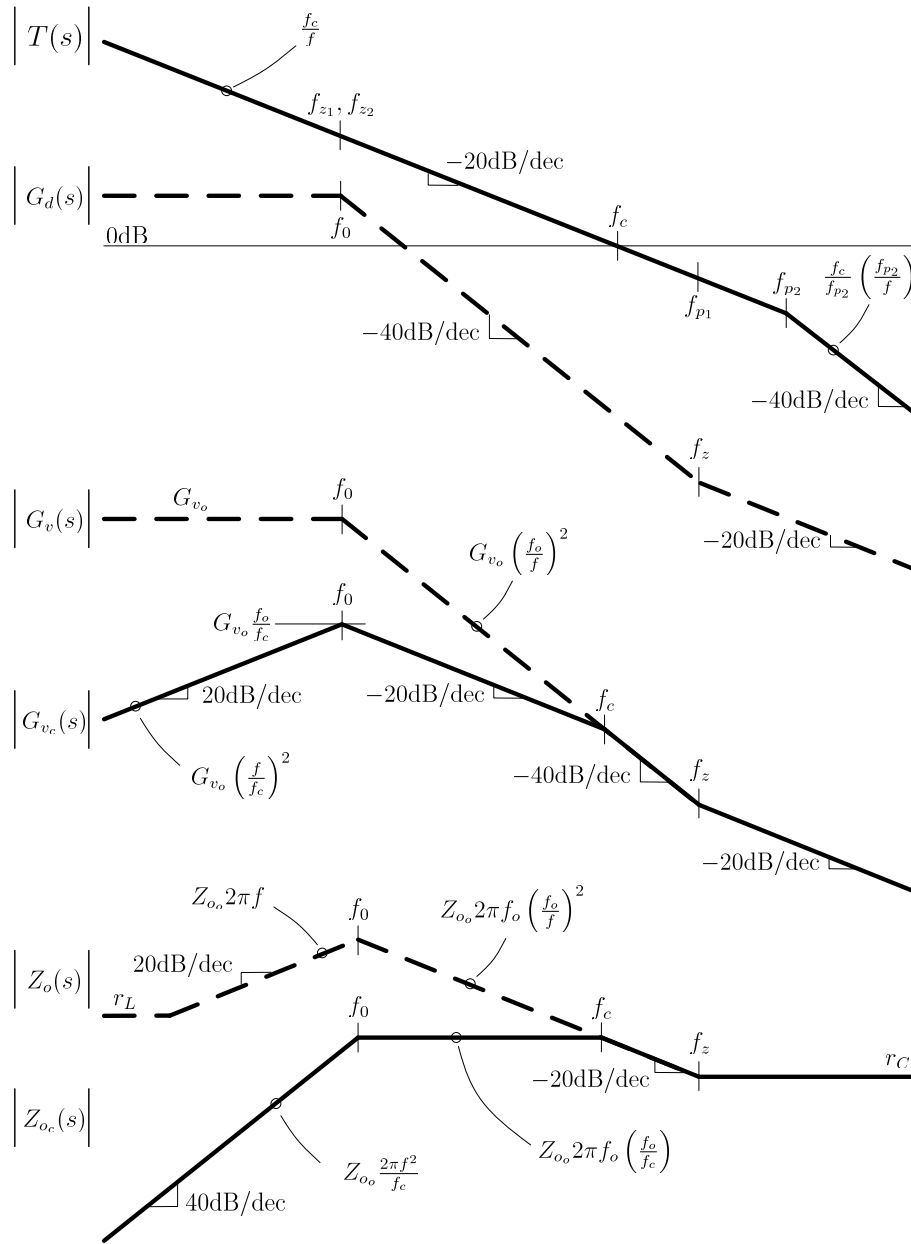


Figure 9.21: Bode Plot Asymptotes for Case 1

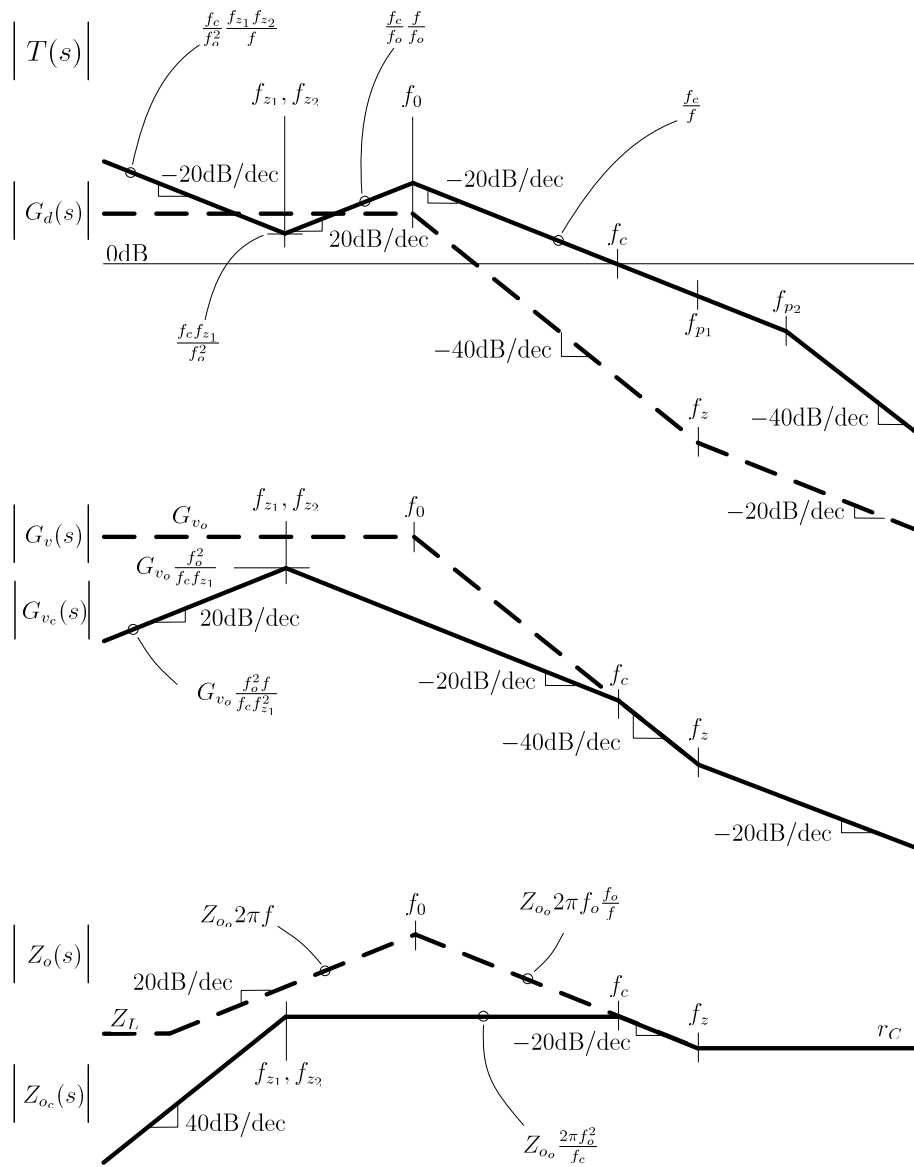


Figure 9.22: Bode Plot Asymptotes for Case 2

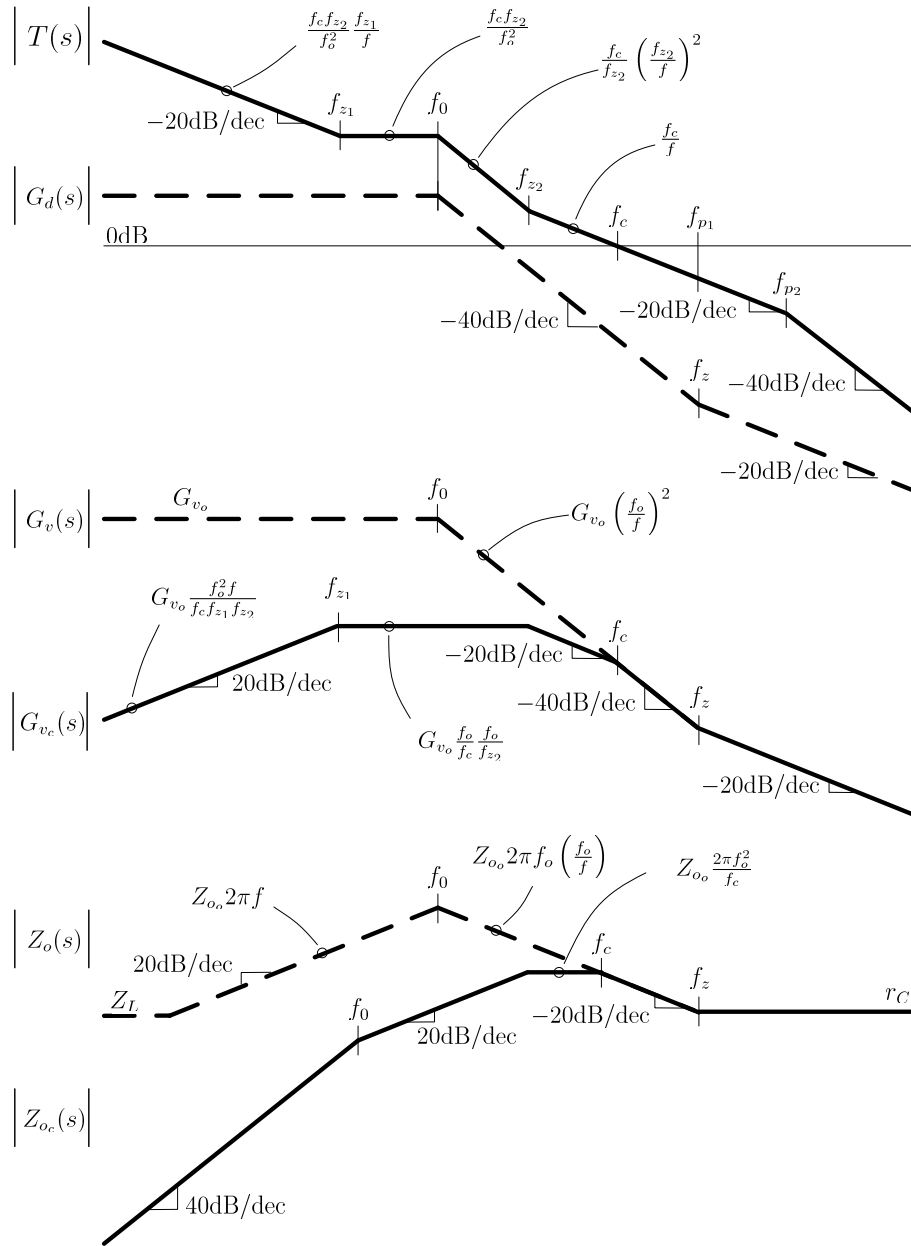


Figure 9.23: Bode Plot Asymptotes for Case 3

To determine a proper location for the introduced components, it is important to first start by improving the phase margin. This is accomplished by having at least one zero below the natural frequency of the system. By introducing one zero rather than both at this point, it will work to exactly cancel the phase shift introduced by the integrator without dropping the loop gain more than necessary. Introducing the second zero after the natural frequency will help extend the loop gain bandwidth, pushing the crossover frequency further toward higher frequency.

Through careful adjustment of parameter location and system response simulation, the system parameters were calculated as shown below. An optimum crossover frequency is found in terms of improving the phase margin, yet also maximizing loop gain bandwidth at  $f_c = 6.31kHz$ . This will be shown in the system response later in this section.

$$\begin{aligned}\omega_{z1} &= (2\pi)5.06 \frac{krad}{s} \\ \omega_{z2} &= (2\pi)8.22 \frac{krad}{s} \\ \omega_{p1} = \omega(z) &= (2\pi)6.37 \frac{krad}{s} \\ \omega_{p2} &= (2\pi)63.6 \frac{krad}{s} \\ \omega_I &= 17k \frac{rad}{s}\end{aligned}$$

Figure 9.24 describe the loop gain of the system as a function of frequency. Note that with the previously discussed changes, high DC gain, switching frequency attenuation, and system stability are all accomplished.

Figure 9.27 is the circuit realization of the compensation scheme designed above. Figure 9.28 shows the simulation results of the circuit in response to the change in output current described in the introduction. Note that the maximum deviation from peak to peak using this voltage compensation scheme is 300 mV. The system shows a spike in voltage when the current is changed, and then quickly reduces to the expected voltage. This clearly shows the difference between this method of design and the constant impedance method, where a constant impedance will have half of the peak to peak voltage deviation; yet, there is a difference in the average output voltage level depending on the load applied. However, since power supply requirements are typically quantified in terms of an output voltage tolerance range, the voltage droop compensation and current droop compensation allow for tighter restrictions on levels, thus making them a better design compared to traditional voltage compensation.

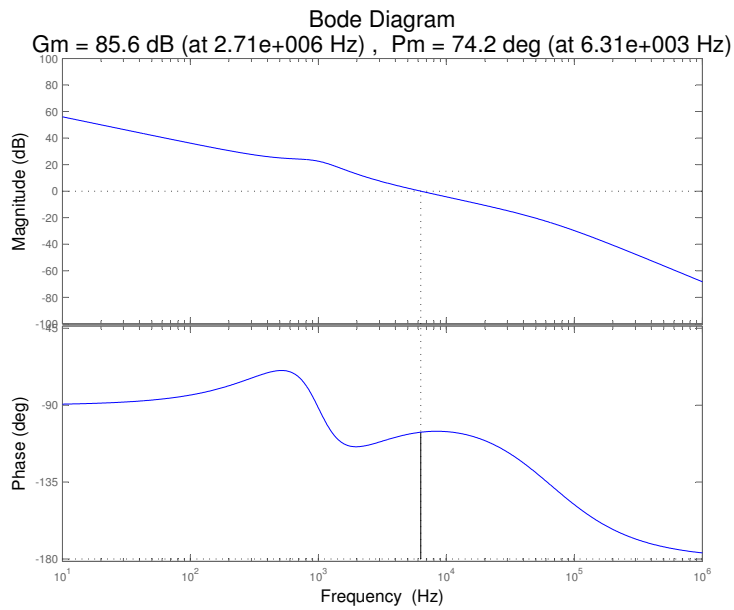


Figure 9.24: MATLAB Bode Plot of Loop Gains with Voltage Mode Compensation

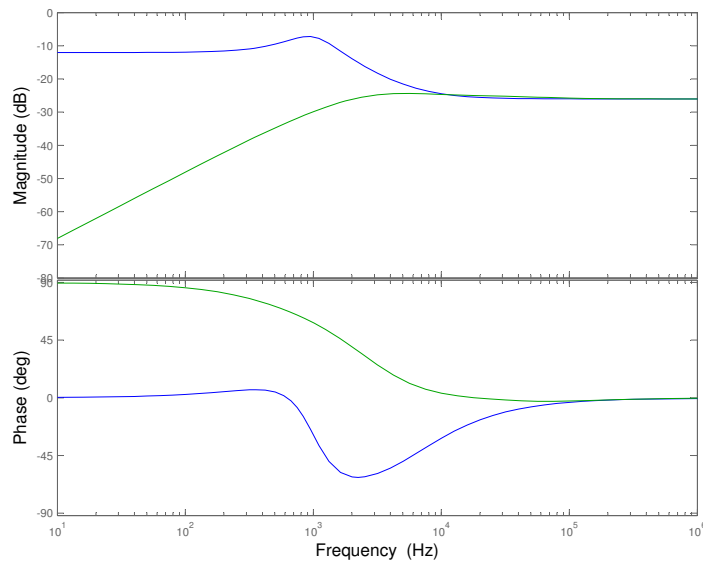


Figure 9.25: MATLAB Bode Plot of  $Z_o$  for Voltage Mode Compensated System

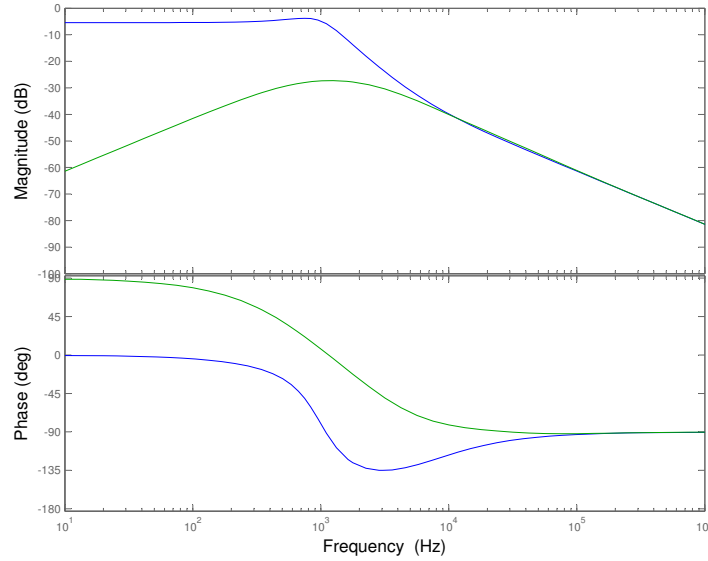


Figure 9.26: MATLAB Bode Plot of  $G_{vg}$  for Voltage Mode Compensated System

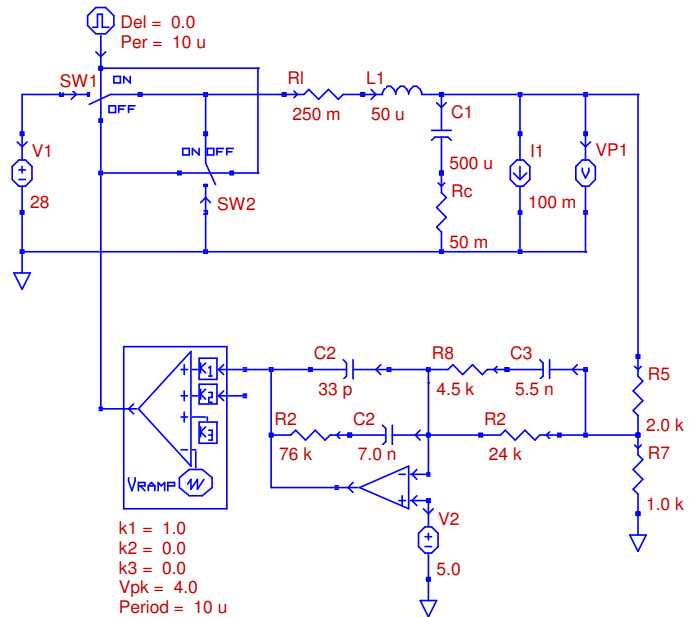


Figure 9.27: PECS Schematic of System with Voltage Mode Compensation



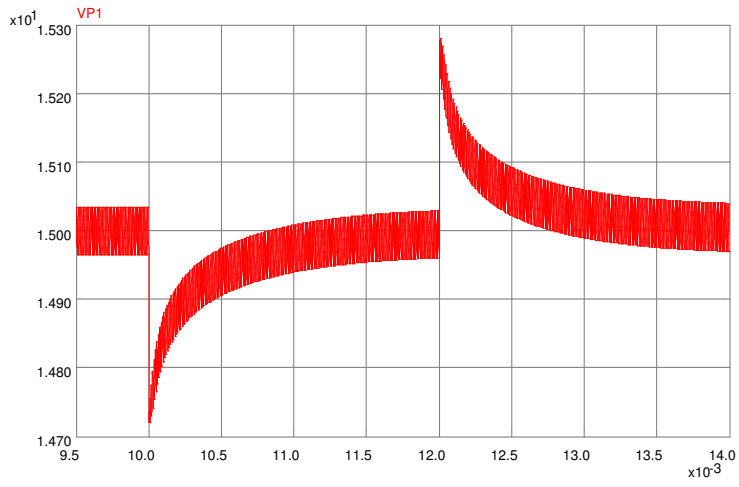


Figure 9.28: PECS Simulation of Voltage Mode Compensated Response to a Load Current Step

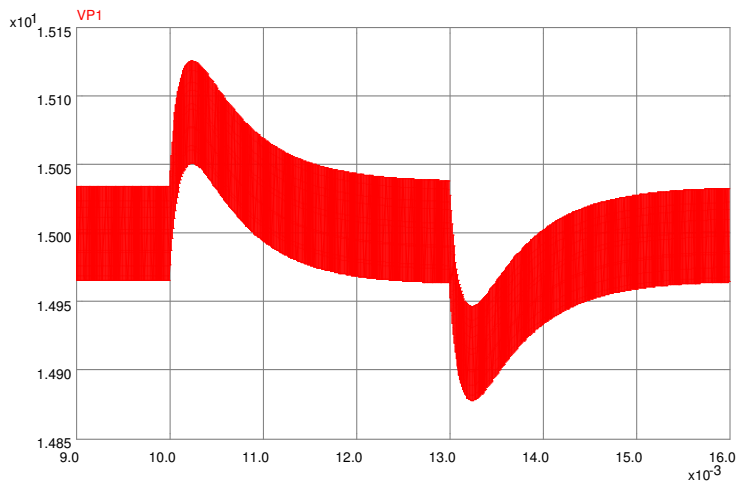


Figure 9.29: PECS Simulation of Voltage Mode Compensated Response to a Supply Voltage Disturbance

## 9.3 Summary

Through implementation of a voltage droop, inductor current droop, and a voltage compensation circuit, it was found that the voltage and current droop compensation techniques were able to have an output voltage deviation of half the size of the voltage compensation method. This reduction in output voltage deviation is an effect of creating a constant closed loop output impedance on the output of the converter. The only downside to the droop compensation methods is that the output voltage changes to the deviation value, rather than introducing a small transient and returning, as is the case with the voltage compensation.

Overall, due to the fact that regulated voltages are defined over an acceptable output range, the droop control methods both allow the smallest deviation that can fit in a given tolerance window. Ideally the current sensing droop control is the most complete method of compensation due to the two loop compensation. Practically, however, the voltage droop method accurately reduces the output voltage deviation and is the simplest to implement in terms of components.

## 9.4 MATLAB Code

```
1 clear
2 close all
3 format compact
4
5 s = tf('s');
6
7 Vg = 28;
8 R = 3;
9 L = 50e-6;
10 C = 500e-6;
11 rl = 0.25;
12 rc = 0.05;
13 Vm = 4;
14 Vo = 15;
15 D = Vo/Vg;
16
17
18 H = 1/3;
19 w0 = 1/sqrt(L*C);
20 f0 = w0/(2*pi);
21 wc = 1/(rc*C);
22 wr = 1/(R*C);
23 wl = rl/L;
24 Q = sqrt(L/C)/(rl+rc);
25 PWM = 1/Vm;
26 T0 = Vg*H*PWM;
27
28
```

```

29 f = logspace(1,6,1000);
30 w = 2*pi*f;
31
32 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
33 %% Plant Open-loop
34
35 Gvg = D*(1+s/wc)/(1+s/(Q*w0)+(s/w0)^2);
36 Gvd = Vg*(1+s/wc)/(1+s/(Q*w0)+(s/w0)^2);
37 Zo = rl*(1+s/wc)*(1+s/wl)/(1+s/(Q*w0)+s^2/w0^2);
38
39 %% Loop gain
40 T = tf(T0*[1/wc 1], [1/w0^2 1/(Q*w0) 1]);
41
42 figure(1)
43 [mag, phase] = bode(T,w);
44 margin(mag, phase, w)
45
46 h = gcr;
47 h.AxesGrid.Xunits = 'Hz';
48 h.AxesGrid.TitleStyle.FontSize = 16;
49 h.AxesGrid.XLabelStyle.FontSize = 12;
50 h.AxesGrid.YLabelStyle.FontSize = 12;
51
52 % Zoc = Zo/(1+T);
53 % Gvgc = Gvg/(1+T);
54 %
55 % figure(2)
56 % bode(Zo);
57 % hold
58 % bode(Zoc)
59 % title('');
60 % h = gcr;
61 % h.AxesGrid.Xunits = 'Hz';
62 % h.AxesGrid.TitleStyle.FontSize = 16;
63 % h.AxesGrid.XLabelStyle.FontSize = 12;
64 % h.AxesGrid.YLabelStyle.FontSize = 12;
65 %
66 %
67 % figure(3)
68 % bode(Gvg);
69 % hold
70 % bode(Gvgc)
71 % title('');
72 % h = gcr;
73 % h.AxesGrid.Xunits = 'Hz';
74 % h.AxesGrid.TitleStyle.FontSize = 16;
75 % h.AxesGrid.XLabelStyle.FontSize = 12;
76 % h.AxesGrid.YLabelStyle.FontSize = 12;
77
78
79
80
81 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
82 %% Droop passive
83
84 Kv = (rl-rc)/(rc*Vg*(1/Vm)*H);
85 wzc = (rl-rc)/(L-rc*rc*C);

```

```

86 wpv = wc;
87
88 Gcdroop_p = tf(Kv*[1/wzc 1], [1/wpv 1]);
89
90 Tdroop_p = Gcdroop_p * T;
91
92 figure(4)
93 [mag, phase] = bode(Tdroop_p,w);
94 margin(mag, phase, w)
95 h = gcr;
96 h.AxesGrid.Xunits = 'Hz';
97 h.AxesGrid.TitleStyle.FontSize = 16;
98 h.AxesGrid.XLabelStyle.FontSize = 12;
99 h.AxesGrid.YLabelStyle.FontSize = 12;
100
101
102 Zoc = Zo/(1+Tdroop_p);
103 Gvgc = Gvg/(1+Tdroop_p);
104
105 figure(2)
106 bode(Zo);
107 hold
108 bode(Zoc)
109 title('');
110 h = gcr;
111 h.AxesGrid.Xunits = 'Hz';
112 h.AxesGrid.TitleStyle.FontSize = 16;
113 h.AxesGrid.XLabelStyle.FontSize = 12;
114 h.AxesGrid.YLabelStyle.FontSize = 12;
115
116
117 figure(3)
118 bode(Gvg);
119 hold
120 bode(Gvgc)
121 title('');
122 h = gcr;
123 h.AxesGrid.Xunits = 'Hz';
124 h.AxesGrid.TitleStyle.FontSize = 16;
125 h.AxesGrid.XLabelStyle.FontSize = 12;
126 h.AxesGrid.YLabelStyle.FontSize = 12;
127
128
129 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
130 %% Droop active
131
132 wi_da = 100000;
133 wz_da = w0;
134 wp_da = 2*pi*50000;
135
136 % wi_da = 7.3529e+004;
137 % wz_da = w0;
138 % wp_da = 2*pi*5.4683e+004;
139
140 Gcvdroop_a = tf(wi_da*[1/wz_da 1], conv([1 0], [1/wp_da 1]));
141
142 Tvdroop_a = Gcvdroop_a * T;

```

```

143
144 figure(5)
145 [mag, phase] = bode(Tvdroop_a,w);
146 margin(mag, phase, w)
147
148 hold
149
150 Ti = tf(Vg/R*[1/wr 1], [1/w0^2 1/(Q*w0) 1]);
151 Tidroop_a = PWM * H *Gcvdroop_a * Ti * rC;
152
153 [mag, phase] = bode(Tidroop_a,w);
154 margin(mag, phase, w)
155
156 T2 = Tvdroop_a/(1+Tidroop_a);
157 [mag, phase] = bode(T2,w);
158 margin(mag, phase, w)
159
160 h = gcr;
161 h.AxesGrid.Xunits = 'Hz';
162 h.AxesGrid.TitleStyle.FontSize = 16;
163 h.AxesGrid.XLabelStyle.FontSize = 12;
164 h.AxesGrid.YLabelStyle.FontSize = 12;
165
166 hold off
167
168
169
170
171 Gii=(1+s/wc)/(1+s/(Q*w0)+s^2/w0^2);
172 Gid=(Vg/R)*(1+s/wr)/(1+s/(Q*w0)+s^2/w0^2);
173 Gig=s*C/(1+s/(Q*w0)+s^2/w0^2);
174
175 % Av=wi*(1+s/w1)/(s*(1+s/w2));
176 % Ti= Av*Fm*Gid*H*Ril;
177 % Tv=Av*Fm*H*Gvd;
178 % T2=Tv/(1+Ti);
179 %Gvgc=Gvg/(1+T2);
180
181 Ti = Tidroop_a;
182 Tv = Tvdroop_a;
183
184 Zoc = (Zo*(1+Ti)+Ti*(Gvd*Gii)/Gid)/(1+Ti+Tv);
185 Gvgc = (Gvg*(1+Ti)-Ti*(Gvd*Gig)/Gid)/(1+Ti+Tv);
186
187 figure(9)
188 bode(Zo);
189 hold
190 bode(Zoc)
191 title('');
192 h = gcr;
193 h.AxesGrid.Xunits = 'Hz';
194 h.AxesGrid.TitleStyle.FontSize = 16;
195 h.AxesGrid.XLabelStyle.FontSize = 12;
196 h.AxesGrid.YLabelStyle.FontSize = 12;
197
198
199 figure(10)

```

```

200 bode(Gvg);
201 hold
202 bode(Gvgc)
203 title('');
204 h = gcr;
205 h.AxesGrid.Xunits = 'Hz';
206 h.AxesGrid.TitleStyle.FontSize = 16;
207 h.AxesGrid.XLabelStyle.FontSize = 12;
208 h.AxesGrid.YLabelStyle.FontSize = 12;
209
210
211
212
213 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
214 %%% Voltage mode
215
216 wi = 17000;
217 wz1 = 0.8*w0;
218 wz2 = 1.3*w0;
219 wp1 = wc;
220 wp2 = 10*wp1;
221
222 Gcvmode = tf(wi*conv([1/wz1 1], [1/wz2 1]), conv([1 0], ...
           conv([1/wp1 1], [1/wp2 1])));
223
224 Tvmode = Gcvmode * T;
225
226 figure(6)
227 [mag, phase] = bode(Tvmode,w);
228 margin(mag, phase, w)
229 h = gcr;
230 h.AxesGrid.Xunits = 'Hz';
231 h.AxesGrid.TitleStyle.FontSize = 16;
232 h.AxesGrid.XLabelStyle.FontSize = 12;
233 h.AxesGrid.YLabelStyle.FontSize = 12;
234
235 Zoc = Zo/(1+Tvmode);
236 Gvgc = Gvg/(1+Tvmode);
237
238 figure(7)
239 bode(Zo);
240 hold
241 bode(Zoc)
242 title('');
243 h = gcr;
244 h.AxesGrid.Xunits = 'Hz';
245 h.AxesGrid.TitleStyle.FontSize = 16;
246 h.AxesGrid.XLabelStyle.FontSize = 12;
247 h.AxesGrid.YLabelStyle.FontSize = 12;
248
249
250 figure(8)
251 bode(Gvg);
252 hold
253 bode(Gvgc)
254 title('');
255 h = gcr;

```

```

256 h.AxesGrid.Xunits = 'Hz';
257 h.AxesGrid.TitleStyle.FontSize = 16;
258 h.AxesGrid.XLabelStyle.FontSize = 12;
259 h.AxesGrid.YLabelStyle.FontSize = 12;
260
261
262
263 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
264 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
265 %%% Components
266
267 %%% Droop passive
268 display(' Droop passive ')
269 R1 = 100000;
270 C1 = 1/(R1*wzc);
271 R2 = Kv*R1;
272 C2 = 1/(R2*wpv);
273
274 R1
275 R2
276 C1
277 C2
278 display(' ')
279
280 %%% Droop active
281
282 display('Droop active ')
283 C1 = 33e-12;
284 R2 = 1/(wp_da*C1);
285 C2 = 1/(R2*wz_da);
286 R1 = 1/(C2*wi_da);
287
288
289 R1
290 R2
291 C1
292 C2
293 display(' ')
294
295 %%% Voltage mode,   R1 >> R3, C1 >> C3
296
297 display('Voltage mode ')
298 C3 = 33e-12;
299 R2 = 1/(C3*wp2);
300 C1 = 1/(R2*wz1);
301 R1 = 1/(C1*wi);
302 C2 = 1/(R1*wz2);
303 R3 = 1/(C2*wp1);
304
305 C1
306 C2
307 C3
308 R1
309 R2
310 R3
311
312 %%% Droop passive

```

```

313 % R1 =
314 %     100000
315 % R2 =
316 %     1.7143e+005
317 % C1 =
318 %     2.4375e-009
319 % C2 =
320 %     1.4583e-010
321
322
323 % Droop active
324 % wi_da = 70000;
325 % wz_da = w0;
326 % wp_da = 70000;
327 %
328 % R1 =
329 %     3.9113e+004
330 % R2 =
331 %     4.3290e+005
332 % C1 =
333 %     3.3000e-011
334 % C2 =
335 %     3.6524e-010
336
337
338 % %% Droop active
339 % wi_da = 150000;
340 % wz_da = w0;
341 % wp_da = 50000;
342 % R1 =
343 %     2.5554e+004
344 % R2 =
345 %     6.0606e+005
346 % C1 =
347 %     3.3000e-011
348 % C2 =
349 %     2.6089e-010
350
351
352 %%% Voltage mode
353 % wi = 7000;
354 % wz1 = 0.3*w0;
355 % wz2 = 1.4*w0;
356 % wp1 = wc;
357 % wp2 = 10*wp1;
358 %
359 % C1 =
360 %     6.9570e-009
361 % C2 =
362 %     5.5000e-009
363 % C3 =
364 %     3.3000e-011
365 % R1 =
366 %     2.3957e+004
367 % R2 =
368 %     7.5758e+004
369 % R3 =

```



```

370 % 4.5455e+003
371
372
373 % %% Voltage mode
374 % wi = 17000;
375 % wz1 = 0.8*w0;
376 % wz2 = 1.3*w0;
377 % wp1 = wc;
378 % wp2 = 10*wp1;
379 %
380 % C1 =
381 % 2.6089e-009
382 % C2 =
383 % 5.3942e-009
384 % C3 =
385 % 3.3000e-011
386 % R1 =
387 % 2.2547e+004
388 % R2 =
389 % 7.5758e+004
390 % R3 =
391 % 4.6346e+003
392
393
394 %%% Droop active
395 R1 = 50e3;
396 R2 = 593e3;
397 C1 = 5e-12;
398 C2 = 267e-12;
399 wi = 1/(R1*(C1+C2));
400 fz = 1/(2*pi*R2*C2);
401 fp = 1/(2*pi*R2*C1*C2/(C1+C2));
402 display('Droop active - good')
403 wi
404 fz
405 fp

```



# References

- [1] K. Yao, M. Xu and F. Lee, "Design Considerations for VRM Transient Response Based on the Output Impedance," *IEEE Trans. Power Electron.*, vol. 18, no. 6, pp. 1270–1277, November 2003.
- [2] K. Yao, K. Lee, M. Xu and F. Lee, "Optimal Design of the Active Droop Control Method for the Transient Response," in *IEEE Applied Power Elec. Conf. and Expo.*, vol. 2, 2003, pp. 718–723.



## Part III

# Classical Control: Laboratory



## Chapter 10

# 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 11

## Lab 1

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

#### 11.1 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.

#### 11.2 Circuit #1

The circuit in Figure 11.1 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.

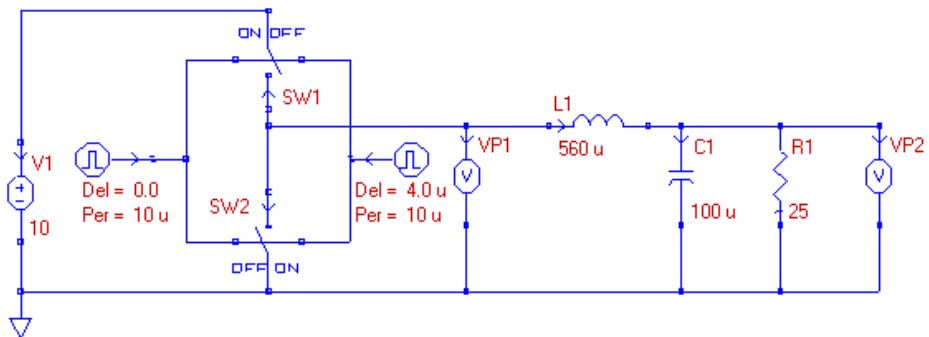


Figure 11.1: A pulse generator followed by a second order (LC) filter, ( $L = 560 \mu H$ ,  $C = 100 \mu F$  and  $R = 25 \Omega$ ) constitutes a dc-to-dc voltage converter. (buck\_clocks.ckt)

### 11.2.1 Tasks

1. Construct the circuit in Figure 11.1 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).

### 11.3 Circuit #2

The circuit in Figure 11.2 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).

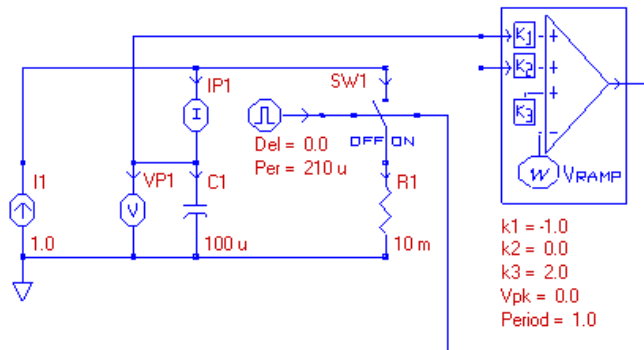


Figure 11.2: Sawtooth generator. A 1 A current source charges a 100  $\mu\text{F}$  capacitor which is periodically quickly discharged through a 10 m $\Omega$  resistor. (sawtooth\_3.ckt)

### 11.3.1 Tasks

1. Construct the circuit in Figure 11.2 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* = 210e-6.
2. Use the following simulation parameters: *Final Time* = 1e-3, *Step Size* = 1e-6, *Start time* = 0, *End time* = 1e-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?

## 11.4 Circuit #3

We will modify the circuit in Figure 11.1 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  $40\text{ kHz}$ . 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.

### 11.4.1 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 12

## Lab 2

### System Identification of a 2<sup>nd</sup> Order System through Step Response

#### 12.1 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*.

#### 12.2 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 12.1.

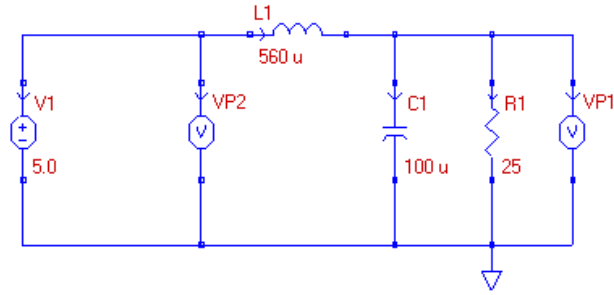


Figure 12.1: PECS schematic of the LCR network used in this lab. The values of the components are  $L = 560 \mu H$ ,  $C = 100 \mu F$  and  $R = 25 \Omega$ . (LCR\_step\_0.ckt)

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

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

where  $\omega_n$  is the undamped natural frequency and  $\zeta$  is the damping factor where  $\zeta < 1$ .

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

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

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

A typical step response for an underdamped second order system is shown in Figure 12.2.

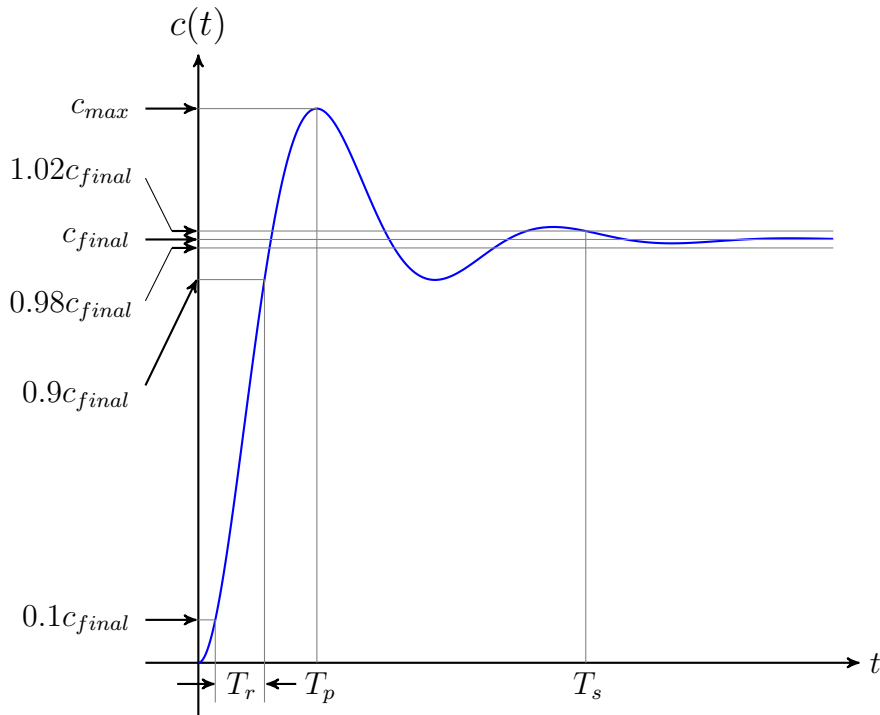


Figure 12.2: Second-order underdamped step response

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 12.2.

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  $\omega_n$  and  $\zeta$  as seen next.

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

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

Making  $\zeta$  the subject of the equation leads to:

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

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

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

or

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

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  $\omega_n$  and  $\zeta$ .

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  $v_0 = 5$ .

## 12.3 Tasks

### 12.3.1 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*  $\rightarrow$  *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.

- 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  $c'_{final}$ .
- 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:

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

- Using the values from Task (5) determine the percentage overshoot, %OS, and subsequently determine the damping factor,  $\zeta$ , using the formulas given above.
- 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:

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

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

### 12.3.2 Matlab simulation – results check

We will next check the results obtained above.

- 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
$\zeta$			
$\omega_n$			
<b>K</b>			

- 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:

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

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", 7<sup>th</sup> edition. Pages 173 – 176.





# Chapter 13

## Lab 3

### System Identification of the dc-to-dc Buck converter

#### 13.1 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.

#### 13.2 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 13.1. 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.

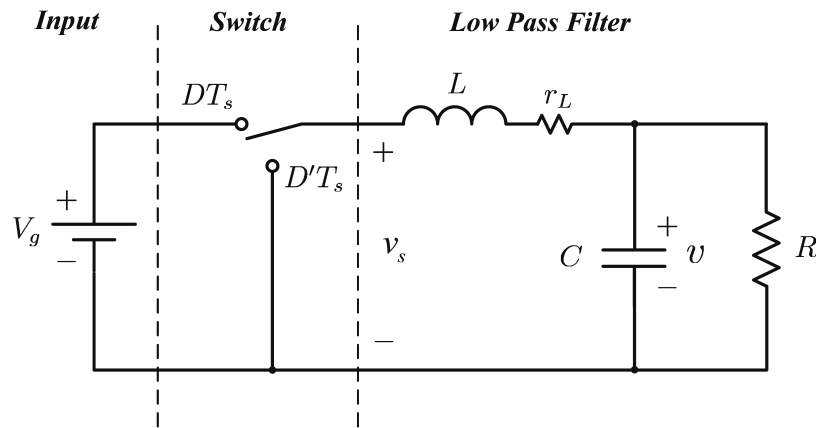


Figure 13.1: Dc-to-dc Buck converter.

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 13.2. 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 13.3. Through the use of the PECS simulator we will examine the steady state operation of this system.

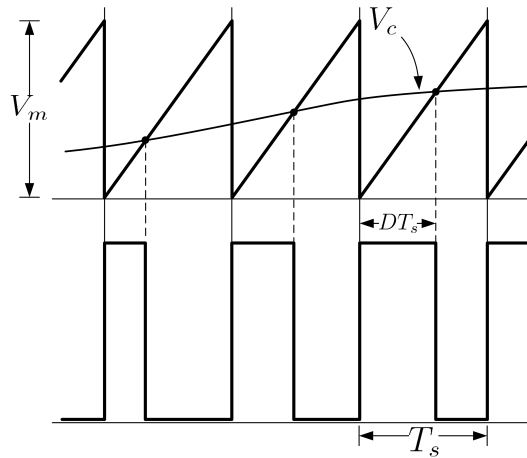
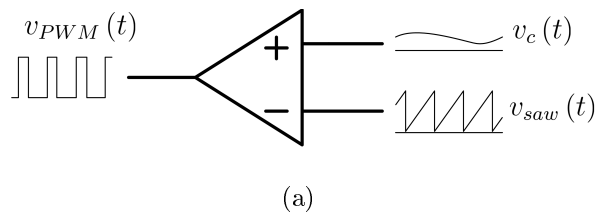


Figure 13.2: (a) PWM comparator, (b) comparison of the control signal,  $v_c$ , with the sawtooth waveform  $v_{saw}$ , results in a variable pulse width rectangular waveform used to drive the switch in the buck converter.

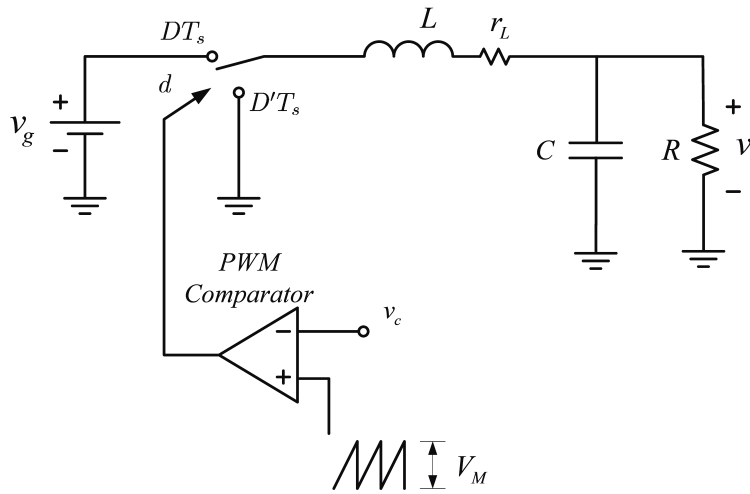


Figure 13.3: Open loop controlled buck converter. The duty ratio  $d$  of the converter is set by the control voltage,  $v_c$ , via the PWM comparator.

### 13.3 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 13.4. In this lab we'll do a partial verification of these transfer functions comparing them with previously derived results.

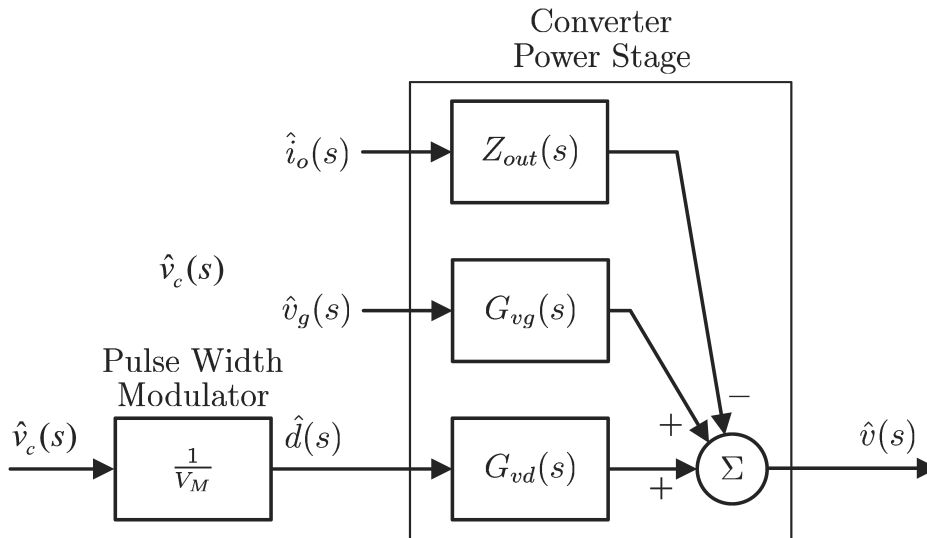


Figure 13.4: Block diagram model of the Buck converter along with the pulse width modulator.

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,  $\xi$ , 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 13.5a, and during the remainder of the period (the  $D'T_s$  interval) is seen in the Figure 13.5b. 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.

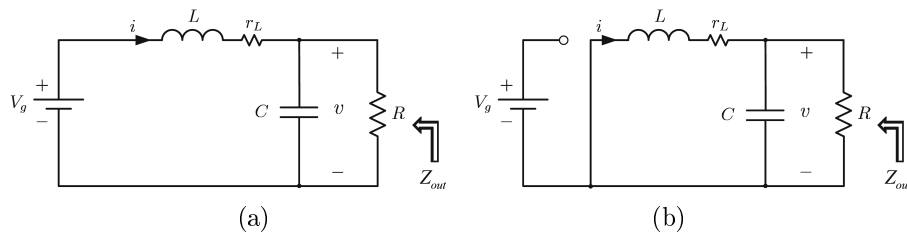


Figure 13.5: Buck converter configuration during (a) the first subinterval,  $DT_s$ , and (b) the second subinterval,  $D'T_s$ .

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 13.6 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 ( $\hat{\phantom{x}}$ ) indicates a small-signal variation. A similar expression can be written for the output voltage:  $v = V + \hat{v}$

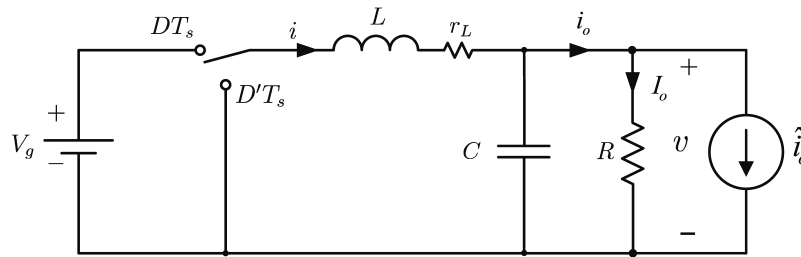


Figure 13.6: The effect of output current variations,  $\hat{i}_o$ , causing output voltage variations,  $\hat{v}$ , is quantified by transfer function  $-Z_{out}$ .

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.

## 13.4 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 13.7 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.

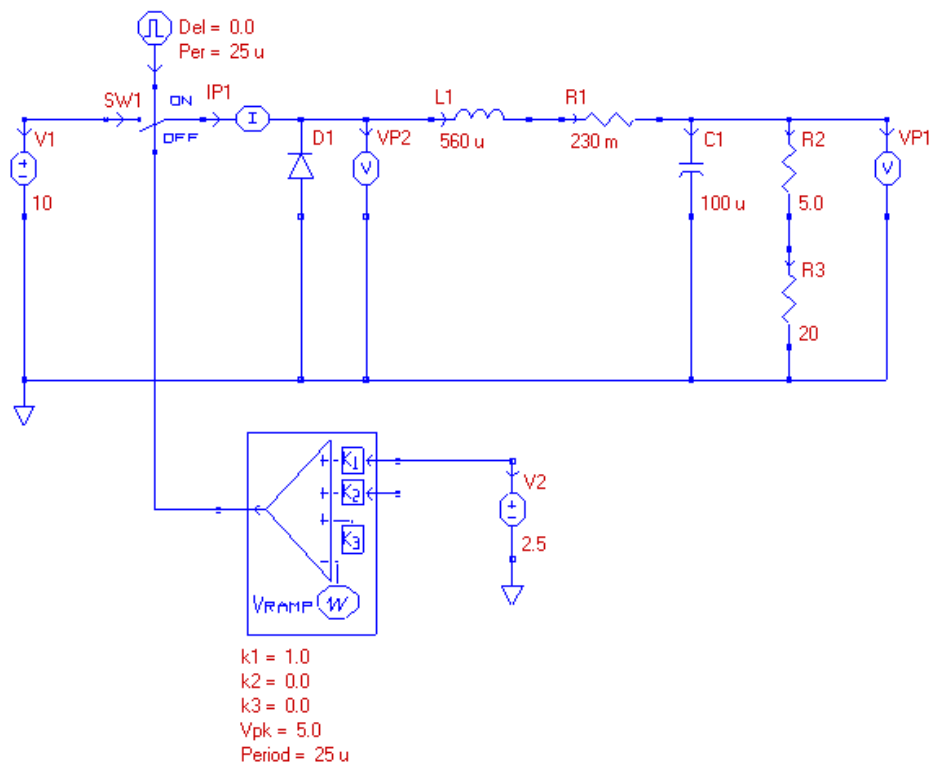


Figure 13.7: PECS schematic of the Buck converter with pulse width modulator. (Lab3\_1.ckt)

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  $\Delta v$  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 13.8. 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 13.4) 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.

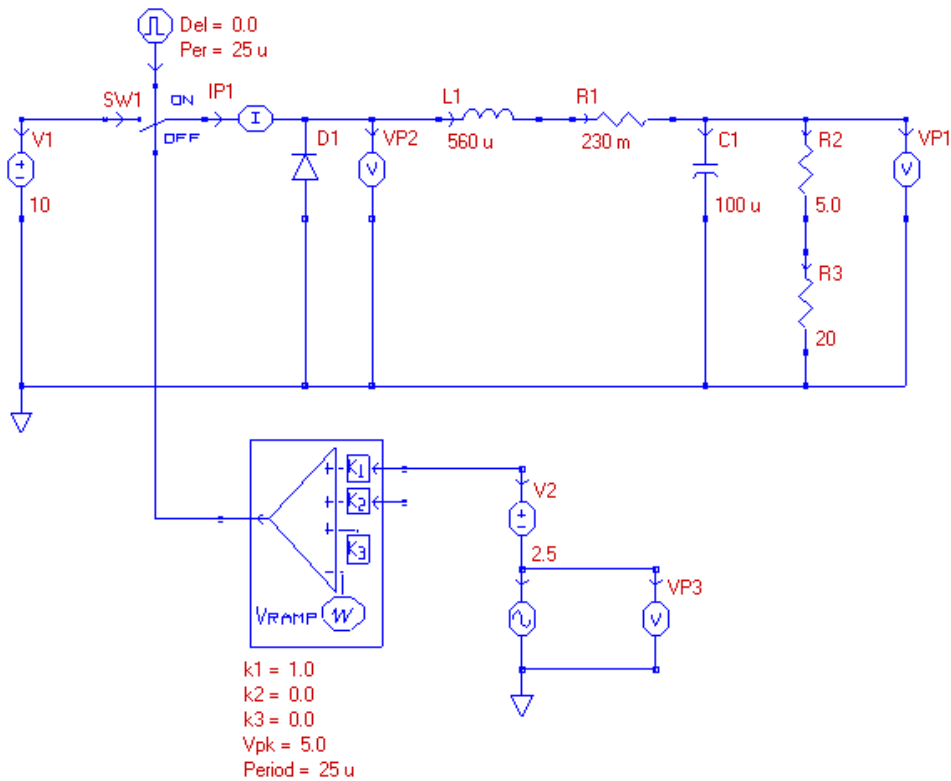


Figure 13.8: Buck converter with pulse width modulator driven by a sinusoidal voltage source. The added sinusoidal source has parameters: *Peak Amplitude* = 0.2 volts, *Frequency* = 100 Hz. and *Phase (degs)* = 0. (Lab3\_2.ckt)

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 13.9. 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,  $\Delta v$ , and steady state error, *SSE*.

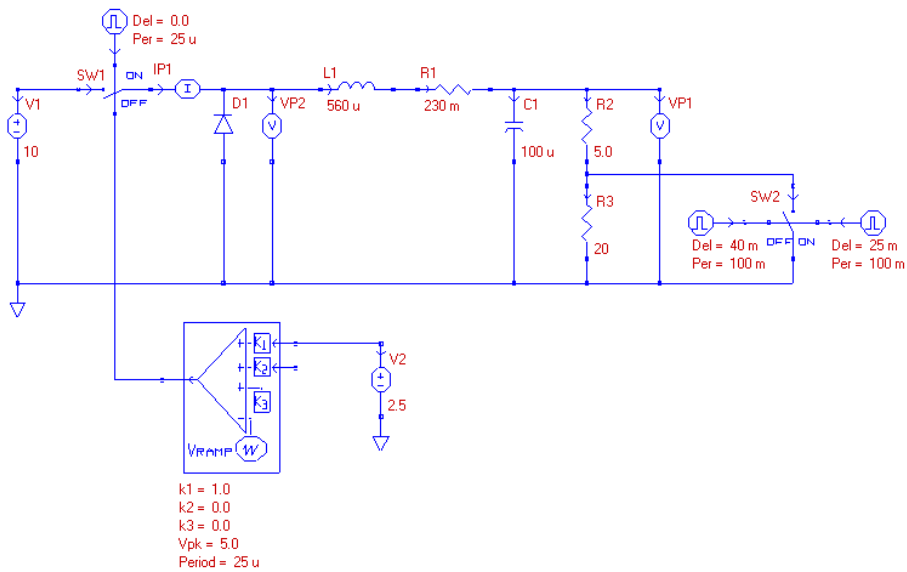


Figure 13.9: Buck converter with step load changes (Lab3\_3.ckt)

## 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  $-180^\circ$  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,  $\Delta v$ , 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,  $\Delta v$ , 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))

```

## 13.5 Note

### Computing maximum peak-to-peak output voltage deviation, $\Delta v$ , and steady state error, $SSE$ :

With reference to Figure 13.10 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  $\Delta v$  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.

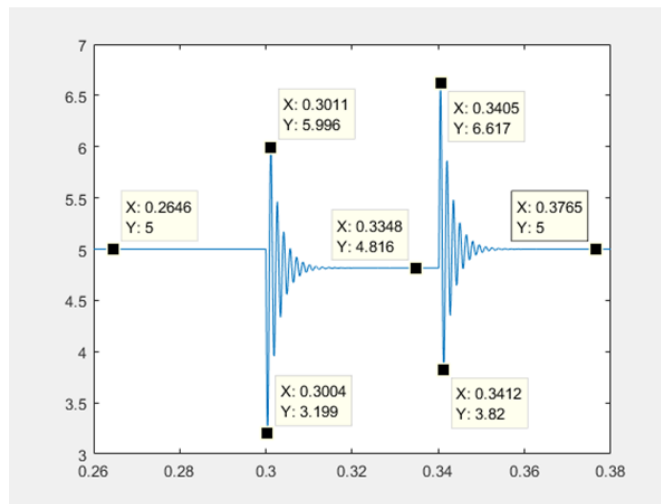


Figure 13.10: The plot shows the output voltage response to an input step change. This response is used to determine the maximum peak-to-peak output voltage deviation,  $\Delta v$ , and steady state error,  $SSE$ .

## 13.6 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: $\Delta v$		
$v_g$ step: $SSE$		
$i_{out}$ step: $\Delta v$		
$i_{out}$ step: $SSE$		

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

1. Comparison of the results for  $\Delta v$  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,  $v_g$ .

## 13.7 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,  $\hat{v}_e$ , 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 14

## Lab 4

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

#### 14.1 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.

#### 14.2 Background

The open loop dc-to-dc buck converter circuit to be constructed is shown in Figure 14.1 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

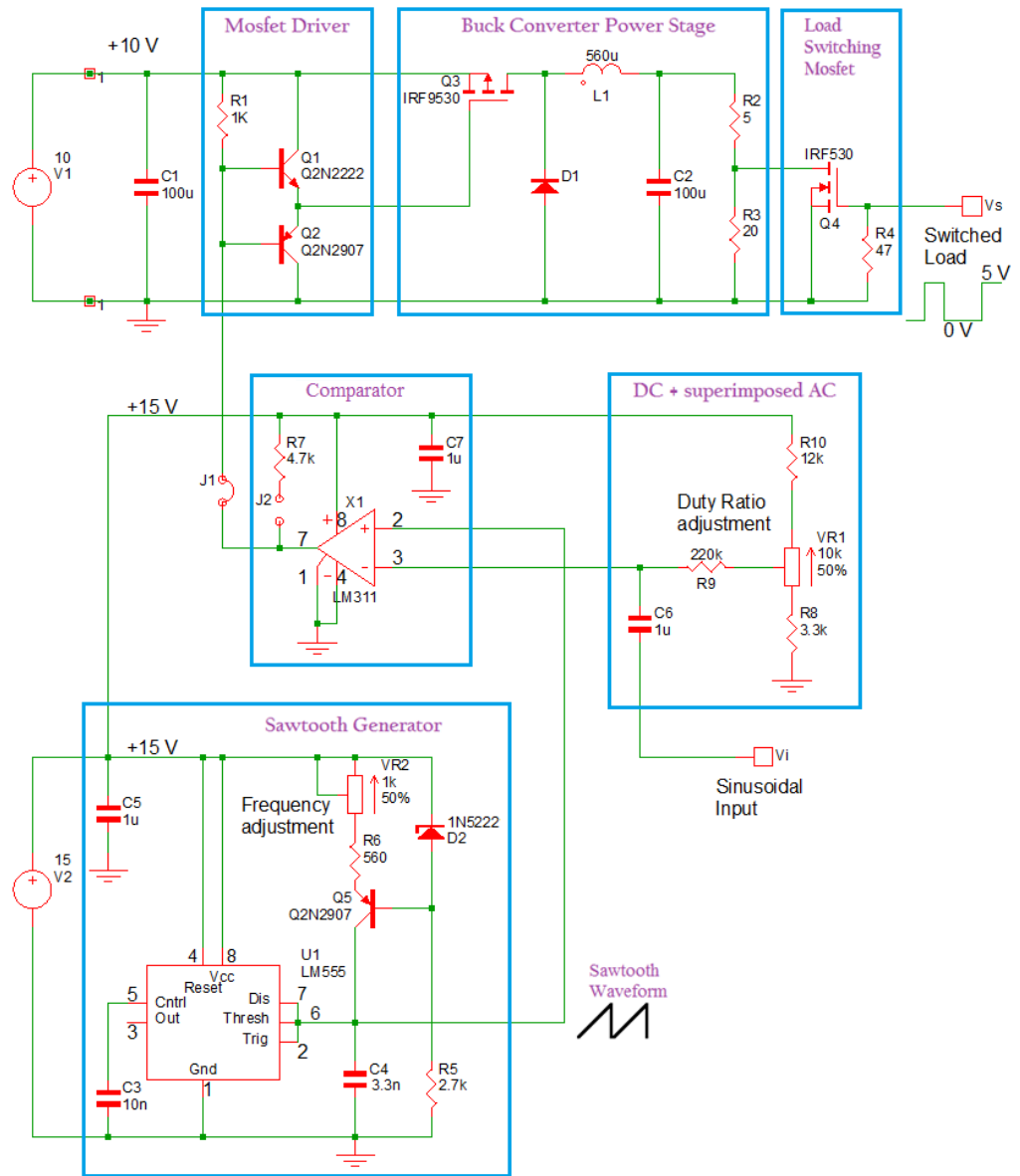


Figure 14.1: Schematic of the open-loop dc-to-dc buck converter.

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 14.1) is used to charge a capacitor (C4 in Figure 14.1) 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 14.1) which adjusts the rate at which the threshold discharge voltage is reached thus varying the frequency of the resulting sawtooth.

### 14.3 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  $40\text{ kHz}$ .
  - (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 screenshot 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  $\Omega$  and 20  $\Omega$  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  $180^\circ$  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,  $\Delta v$ , and the steady state error,  $SSE$ . If necessary, please see the note in Lab. 3 concerning how to determine these quantities.

## 14.4 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
<b>Sawtooth (Task 1b)</b>			

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
$\Delta v$			
$SSE$			

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

## 14.5 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  $\mu F$  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  $\mu H$  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  $\Omega$ . Its n-channel complement, the IRF530 mosfet, used to switch the load, has a maximum  $R_{DS(on)}$  of 0.16  $\Omega$ .

**Load resistors:** The values and power ratings of the load resistances are 5  $\Omega$ , 5  $W$  and 20  $\Omega$ , 2  $W$ .

# Chapter 15

## Lab 5

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

#### 15.1 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.

#### 15.2 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 15.1. 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

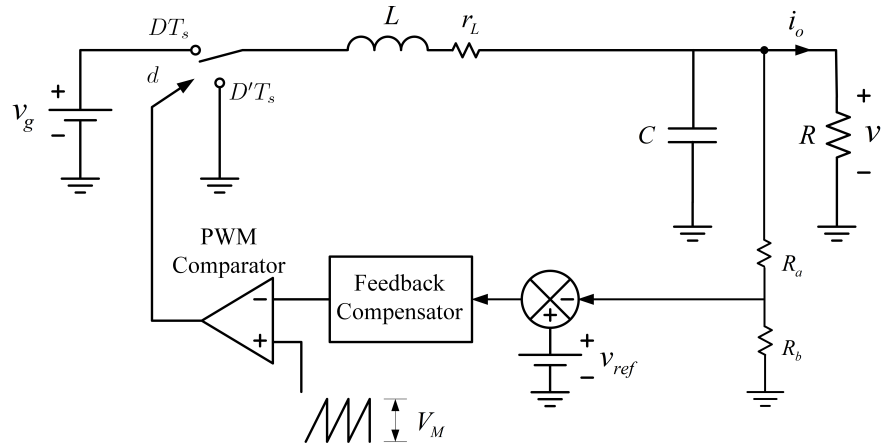


Figure 15.1: Closed loop dc-to-dc buck converter voltage regulator

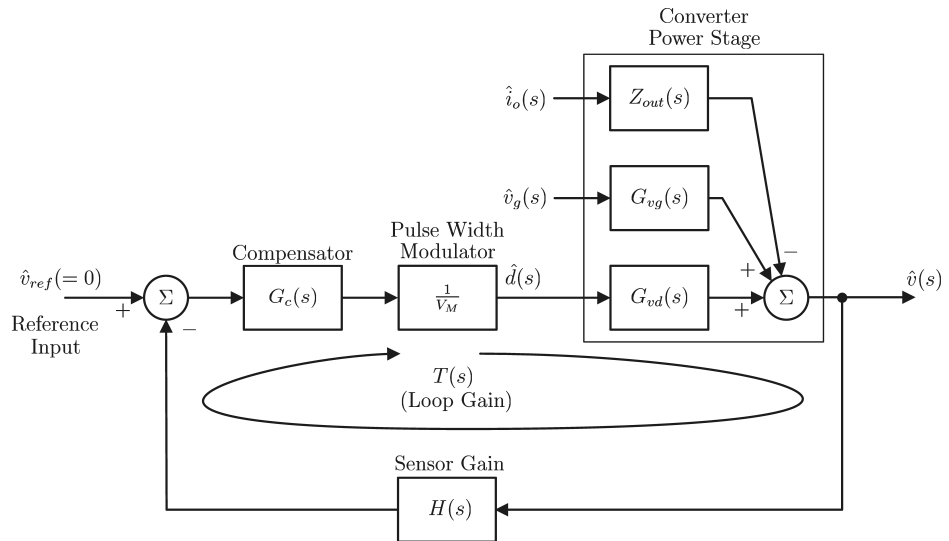


Figure 15.2: Closed loop dc-to-dc buck converter voltage regulator block diagram

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

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 ‘ $\hat{\phantom{x}}$ ’ 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 15.2. 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 15.2.

The open loop transfer functions for the converter and modulator are summarized in Table 15.1.

Table 15.1: 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 15.4. 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 15.3. 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 15.2), in closing the loop around the system. The resulting compensated loop gain transfer function of the system will be examined.

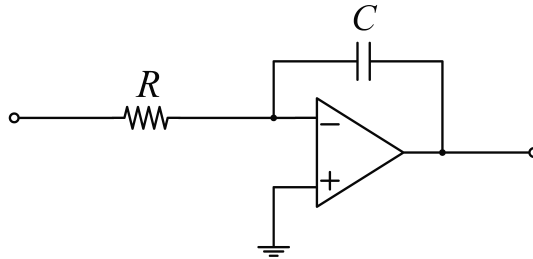


Figure 15.3: Integral compensator implementation,  $G_c(s) = -\frac{K_i}{s}$ , where  $K_i = \frac{1}{RC}$

## 15.3 Tasks

### 15.3.1 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. Uncompensated loop gain, 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  $K_i = \frac{1}{RC}$ . (Ignore the negative sign in Bode plot construction.) Don't forget to annotate the sketch.
  - iii. Construct the (composite) asymptotic Bode plot 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

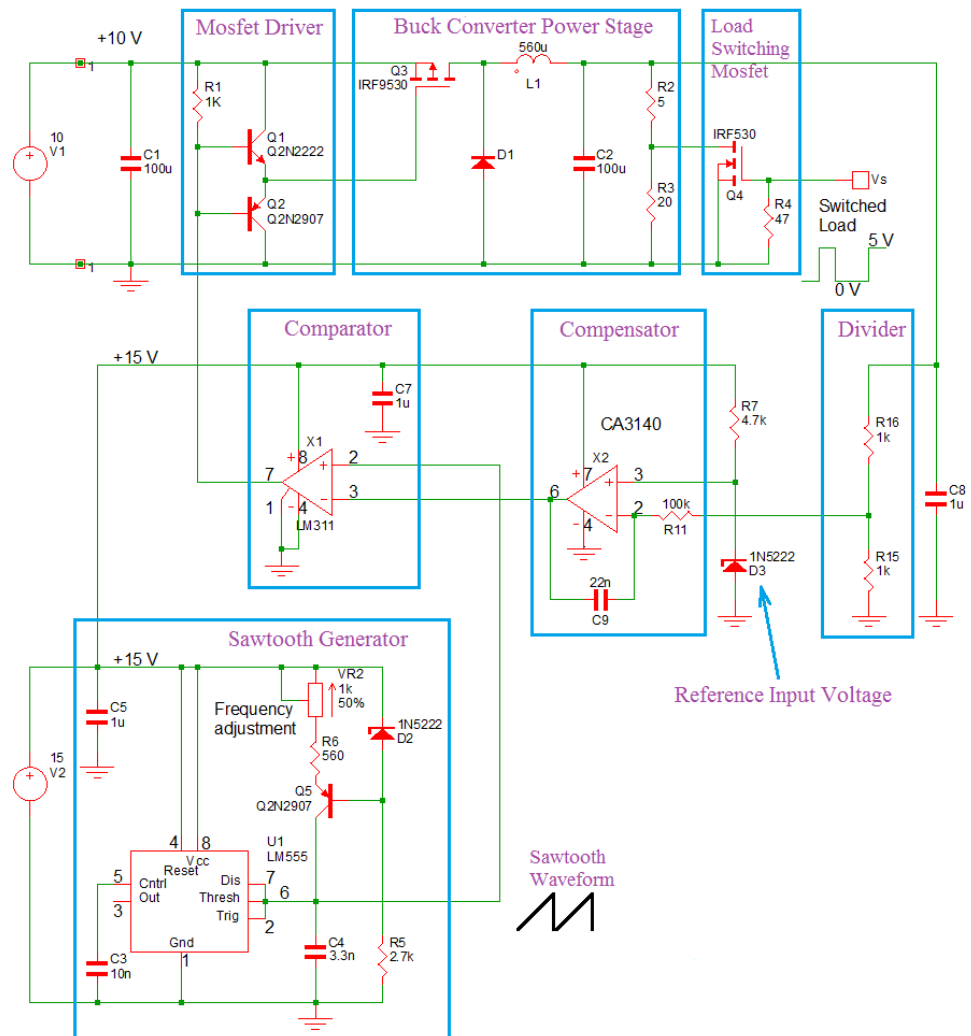


Figure 15.4: Schematic of the closed-loop voltage regulator system.



plot determine the phase margin and unity gain crossover frequency, and the gain margin (in dB) and the associated  $-180^\circ$  phase crossover frequency.

- (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  $-180^\circ$  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,  $\Delta v$ , 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,  $\Delta v$ , 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,  $\Delta v$ , 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,  $\Delta v$ , and steady state error, *SSE*.

### 15.3.2 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 15.4. 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,  $\Delta v$ , and steady state error,  $SSE$ .

### 15.3.3 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 $\phi_{PM}$ (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 $\Delta v$			Unavailable
$v_g$ : step $SSE$			Unavailable
$i_{out}$ : step $\Delta v$			
$i_{out}$ : 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  $\omega_z$  and  $\omega_p$

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.)

## 15.4 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.

### 15.4.1 Proportional Control

8. The proportional controller is shown in Figure 15.5. 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.

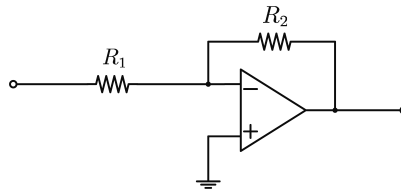


Figure 15.5: Proportional compensator implementation,  $G_c(s) = -K_p$ , where  $K_p = \frac{R_2}{R_1}$

### 15.4.2 Lead Control

9. The lead controller is shown in Figure 15.6.

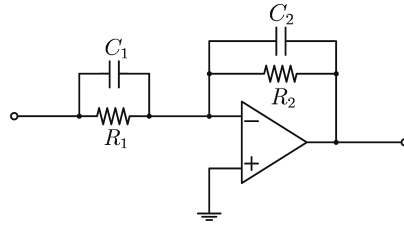


Figure 15.6: Lead compensation implementation,  $G_c(s) = -K \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{\omega_p}}$ , for  $\omega_z < \omega_p$ .

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.

## 15.5 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 16

## Lab 6

### **Closed Loop Feedback System - Design: Design of an effective feedback compen- sator 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.

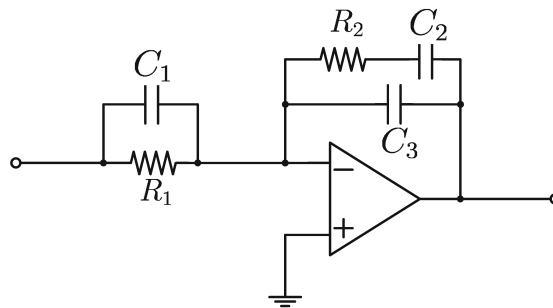


Figure 16.1: ‘Dominant pole plus lead’ compensator to be designed in this lab.

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  $\geq 45^\circ$  and gain margin  $\geq 10$  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 & Rytönen 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  $-180^\circ$  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,  $\Delta v$ , 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,  $\Delta v$ , 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,  $\Delta v$ , 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  $\Omega$  to 5  $\Omega$  back to 25  $\Omega$ . Obtain the step response plot and determine the maximum peak-to-peak output voltage deviation,  $\Delta v$ , 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,  $\Delta v$ , 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 $\phi_{PM}$ (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:

- $\phi_{PM}$ : phase margin obtained from Matlab *margin* command
- $f_c$ : unity gain crossover frequency obtained from Matlab *margin* command
- $\Delta v$ : 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)
<b>Compensator Transfer Function</b> $G_c(s)$	No compensator		
$\phi_{PM}$ (degrees) <b>MATLAB</b>			
$f_c$ (kHz) <b>MATLAB</b>			
$i_{out}$ step: $\Delta v$ (mV) <b>LAB</b>			
$i_{out}$ step: $\Delta v$ (mV) <b>PECS</b>			
$i_{out}$ step: $\Delta v$ (mV) <b>MATLAB</b>			
$i_{out}$ step: SSE (mV) <b>LAB</b>			
$i_{out}$ step: SSE (mV) <b>PECS</b>			
$i_{out}$ step: SSE (mV) <b>MATLAB</b>			
$v_g$ step: $\Delta v$ (mV) <b>PECS</b>			
$v_g$ step: $\Delta v$ (mV) <b>MATLAB</b>			
$v_g$ step: SSE (mV) <b>PECS</b>			
$v_g$ step: SSE (mV) <b>MATLAB</b>			

\*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 $\mu$ F capacitor electrolytic
C2	100 $\mu$ F capacitor Kemet A758EK107M1AAAE016
C3	10 nF capacitor
C4	3.3 nF capacitor
C5	1 $\mu$ F capacitor
C6	1 $\mu$ F capacitor
C7	1 $\mu$ F capacitor
C8	1 $\mu$ 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 $\mu$ H inductor Bourns Inc. 2200HT-561-V-RC
R1	1 $k\Omega$ $\frac{1}{16}$ W Resistor
R2	5 $\Omega$ 5 W Power Resistor
R3	20 $\Omega$ 2 W Power Resistor
R4	47 $\Omega$ $\frac{1}{16}$ W Resistor
R5	2.7 $k\Omega$ $\frac{1}{16}$ W Resistor
R6	560 $\Omega$ $\frac{1}{16}$ 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

# Lab Grading Sheets

## Lab 1 Grading Sheet

### Circuit #1:

1. Your PECS schematic \_\_\_\_\_/1
2. Task 4:
  - (i) VP2 plot \_\_\_\_\_/1
  - (ii) Steady state output voltage value \_\_\_\_\_/1
3. Tasks 6 and 7:
  - (i) Zoomed plot with VP2 and VP1 \_\_\_\_\_/1
  - (ii) Peak-to-peak ripple of VP2 \_\_\_\_\_/1
4. Task 8: VP1:
  - (i) peak amplitude \_\_\_\_\_/1
  - (ii) period \_\_\_\_\_/1
  - (iii) pulse width \_\_\_\_\_/1
  - (iv) duty ratio \_\_\_\_\_/1
5. Task 9: Explanation of steady state value \_\_\_\_\_/1

### Circuit #2:

1. Task 1: Your PECS schematic \_\_\_\_\_/1
2. Task 3: Plot of VP1 and IP1 \_\_\_\_\_/1
3. Task 4: VP1 (voltage across the capacitor):
  - (i) peak amplitude \_\_\_\_\_/1
  - (ii) period \_\_\_\_\_/1
  - (iii) Factors determining amplitude and \_\_\_\_\_/1
  - (iv) why? \_\_\_\_\_/1

### Circuit #3:

1. Task 1: Your PECS schematic \_\_\_\_\_/3
2. Task 2
  - (i) VP2 plot \_\_\_\_\_/1
  - (ii) Steady state output voltage value \_\_\_\_\_/1
  - (iii) Zoomed plot with VP2 and VP1 \_\_\_\_\_/1
  - (iv) Peak-to-peak ripple of VP2 \_\_\_\_\_/1
  - (v) VP1:
    - (a) peak amplitude \_\_\_\_\_/1
    - (b) period \_\_\_\_\_/1
    - (c) pulse width \_\_\_\_\_/1
    - (d) duty ratio \_\_\_\_\_/1
  - (vi) Explanation of steady state value \_\_\_\_\_/1

3. Task 3: Completed table \_\_\_\_\_/5

4. Task 4:

(i) The difference in ripple values between Circuit #3 and Circuit #1 \_\_\_\_\_/1

(ii) Why would you expect this? \_\_\_\_\_/1

Report: \_\_\_\_\_/5

**Total:** \_\_\_\_\_/40

## Lab 2 Grading Sheet

1. Task 1:	Your PECS schematic_____	/1
2. Task 4:		
	(i) Plot_____	/1
	(ii) $c'_{max}$ _____	/1
	(iii) $c'_{final}$ _____	/1
3. Task 5:		
	(i) $c_{max}$ _____	/1
	(ii) $c_{final}$ _____	/1
4. Task 6:		
	(i) %OS_____	/2
	(ii) $\zeta$ _____	/2
5. Task 7:		
	(i) $T'_s$ _____	/1
	(ii) $T_s$ _____	/1
6. Task 8:	$K$ _____	/2
7. Task 9:		
	(i) LCR transfer function derivation_____	/2
	(ii) $K$ as a function of elements_____	/1
	(iii) $a1$ as a function of elements_____	/1
	(iv) $a2$ as a function of elements_____	/1
	(v) completed table_____	/6
8. Task 10:	Your complete Matlab code_____	/2
9. Task 11:		
	(i) Matlab plot 1_____	/1
	(ii) Matlab plot 2_____	/1
10. Task 12:		
	(i) Output of Matlab <i>stepinfo</i> function_____	/1
	(ii) Completed table_____	/2
11. Task 13:		
	(i) Derivation of transfer function with $rL$ included_____	/2
	(ii) Transfer function with simplification applied_____	/1
	(iii) Appreciable change in DC gain?_____	/1
	(iv) Appreciable change in undamped natural frequency?_____	/1
	(v) Appreciable change in damping factor?_____	/1
Report:	_____	/7
<b>Total:</b>	_____	<b>/45</b>

## Lab 3 Grading Sheet

1. Task 1:
  - (i) Your PECS schematic \_\_\_\_\_/1
  - (ii) Start-up transient plot \_\_\_\_\_/1
  - (iii) Steady state output voltage value \_\_\_\_\_/1
2. Task 2:
  - (i) Plot of output voltage, diode voltage and switch current in proper format  
/3
  - (ii) Average output voltage value \_\_\_\_\_/1
  - (iii) Peak-to-peak output voltage ripple value \_\_\_\_\_/1
  - (iv) Peak-to-peak diode voltage value \_\_\_\_\_/1
  - (v) Peak inductor current \_\_\_\_\_/1
  - (vi) Switching frequency \_\_\_\_\_/1
  - (vii) Duty ratio \_\_\_\_\_/1
  - (viii) Explanation of how duty ratio is set \_\_\_\_\_/2
3. Task 3:
  - (i) Simulation plot \_\_\_\_\_/1
  - (ii)  $K_{vg}$  \_\_\_\_\_/2
  - (iii)  $\Delta v$  \_\_\_\_\_/1
  - (iv) SSE \_\_\_\_\_/1
4. Task 4:
  - (i) Your PECS schematic \_\_\_\_\_/1
  - (ii) Filter corner frequency in Hz. \_\_\_\_\_/1
  - (iii) Plot of VP3 and VP1 in proper format \_\_\_\_\_/1
  - (iv) Peak-to-peak of VP3 and of VP1 \_\_\_\_\_/2
  - (v) In-phase or out of phase? \_\_\_\_\_/1
  - (vi)  $K_{vd}$  \_\_\_\_\_/2
5. Task 5:
  - (i) Your PECS schematic \_\_\_\_\_/1
  - (ii) Simulation plot of VP1 \_\_\_\_\_/1
  - (iii)  $\Delta v$  \_\_\_\_\_/1
  - (iv) SSE \_\_\_\_\_/1
6. Task 6a:
  - (i) Loop gain transfer function \_\_\_\_\_/2
  - (ii) Your complete MATLAB code \_\_\_\_\_/1
  - (iii) Matlab plot \_\_\_\_\_/1

(iv) Unity gain and phase crossover frequencies, phase and gain margins values	/2
7. Task 6b:	
(i) $G_{vd}$ transfer function	/2
(ii) Your Matlab code	/1
(iii) Your Matlab plot	/1
8. Task 6c:	
(i) $-Z_{out}$ transfer function	/2
(ii) Your Matlab code	/1
(iii) Your Matlab plot	/1
9. Task 6d:	
(i) Your complete Matlab code	/1
(ii) Matlab plot	/1
(iii) $\Delta v$	/1
(iv) SSE	/1
10. Task 6e:	
(i) $Z_{out}$ transfer function	/1
(ii) Your Matlab code	/1
(iii) Your Matlab plot	/1
(iv) $\Delta v$	/1
(v) SSE	/1
11. Results (summary):	
(i) Transfer function DC gains' table	/3
(ii) Margins and associated frequencies table	/2
(iii) Step response characteristics table	/2
(iv) Comparison of results for $\Delta v$ and $i_{out}$ changes	/2
(v) Simple symbolic expression for SSE for a step in $\Delta v$	/2
Report:	/5
<b>Total:</b>	<b>/70</b>

## Lab 4 Grading Sheet

1. One photo of constructed circuit (power stage on perf board and the rest on solderless board?) \_\_\_\_\_/7
2. Task 1b:
  - (i) Screenshot of sawtooth (40 kHz?) \_\_\_\_\_/1
  - (ii) Minimum values of sawtooth \_\_\_\_\_/1
  - (iii) Maximum values of sawtooth \_\_\_\_\_/1
3. Task 2b:
  - (i) Minimum achievable duty ratio calculation \_\_\_\_\_/2
  - (ii) Maximum achievable duty ratio \_\_\_\_\_/2
  - (iii) Minimum achieved duty ratio value obtained in the lab \_\_\_\_\_/1
  - (iv) Maximum achieved duty ratio value obtained in the lab \_\_\_\_\_/1
4. Task 2c:
  - (i) Screenshot of output of LM311 (15 V pk-pk? and 50% duty ratio?) \_\_\_\_\_/1
5. Task 3b:
  - (i) Screenshot of output of mosfet driver (10 V pk-pk?) \_\_\_\_\_/1
6. Task 4b:
  - (i) Screenshot of diode voltage (10 V pk-pk? Sharp transitions?) \_\_\_\_\_/1
7. Task 4c:
  - (i) Measured value of average output voltage \_\_\_\_\_/1
  - (ii) Expected value of average output voltage (why?) \_\_\_\_\_/1
8. Task 4d:
  - (i) Screenshot of input and output sinusoidal \_\_\_\_\_/1
  - (ii) Measured pk-pk value of output sinusoid \_\_\_\_\_/1
  - (iii) Measured pk-pk value of input \_\_\_\_\_/1
  - (iv) Gain = ratio of output/input calculated \_\_\_\_\_/1
  - (v) What does this gain represent? \_\_\_\_\_/1
  - (vi) What are the components of this gain? \_\_\_\_\_/2
  - (vii) In-phase or out-of-phase? \_\_\_\_\_/1
  - (viii) Will the compensator need to provide phase inversion so as to provide negative feedback? \_\_\_\_\_/1
9. Task 5b:
  - (i) Screenshot of output voltage response to stepped load change \_\_\_\_\_/1
  - (ii)  $\Delta v$ , maximum pk-pk output voltage deviation \_\_\_\_\_/1
  - (iii) *SSE*, steady state error \_\_\_\_\_/1

10. Results (summary):

- (i) Sawtooth table \_\_\_\_\_/1
- (ii) Duty ratio table \_\_\_\_\_/1
- (iii) Sinusoid table \_\_\_\_\_/1
- (iv)  $\Delta v$  and SSE table \_\_\_\_\_/5
- (v) Observations concerning step response \_\_\_\_\_/4

Report: \_\_\_\_\_/5

**Total:** \_\_\_\_\_/50



## Lab 5 Grading Sheet

1. Task 1:
  - (i) Sketch of magnitude and phase response of uncompensated loop gain \_\_\_\_\_/1
  - (ii) Sketch properly annotated \_\_\_\_\_/1
2. Task 1a.ii:
  - (i) Sketch of magnitude and phase response of compensator \_\_\_\_\_/1
  - (ii) Sketch properly annotated \_\_\_\_\_/1
3. Task 1a.iii:
  - (i) Sketch of magnitude and phase response of compensated loop gain (i.e. above two combined) \_\_\_\_\_/1
  - (ii) Sketch properly annotated \_\_\_\_\_/2
  - (iii) From sketch determine, unity gain crossover frequency \_\_\_\_\_/1
  - (iv) From sketch determine, phase margin \_\_\_\_\_/1
  - (v) From sketch determine, crossover frequency \_\_\_\_\_/1
  - (vi) From sketch determine, gain margin \_\_\_\_\_/1
4. Task 1b:
  - (i) Expression for compensated loop gain,  $T$  \_\_\_\_\_/1
  - (ii) Expression for denominator polynomial of  $1/(1+T)$  or  $T/(1+T)$  \_\_\_\_\_/1
  - (iii) Stable range of  $K_i$  \_\_\_\_\_/1
  - (iv) Actual value of  $K_i$  used \_\_\_\_\_/1
5. Task 2a:
  - (i) Matlab *margin* plot \_\_\_\_\_/1
  - (ii) unity gain crossover frequency, phase margin, -180 crossover frequency, gain margin \_\_\_\_\_/1
6. Task 2b:
  - (i) Expression for  $G_{vg\_CL}$  \_\_\_\_\_/1
  - (ii) Matlab code to produce *bodemag* plots of  $G_{vg\_CL}$  and (open loop)  $G_{vg}$  /1
  - (iii) Matlab *bodemag* plots of  $G_{vg\_CL}$  and (open loop)  $G_{vg}$  \_\_\_\_\_/1
  - (iv) (iii) with requested frequency range \_\_\_\_\_/1
7. Task 2c:
  - (i) Expression for  $-Z_{out\_CL}$  \_\_\_\_\_/1
  - (ii) Matlab code to produce *bodemag* plots of  $-Z_{out\_CL}$  and (open loop)  $-Z_{out}$  /1
  - (iii) Matlab *bodemag* plots of  $-Z_{out\_CL}$  and (open loop)  $-Z_{out}$  \_\_\_\_\_/1
  - (iv) (iii) with requested frequency range \_\_\_\_\_/1
8. Task 3a:

(i)	Full Matlab code for simulation of step response of $G_{vg\_CL}$	/1
(ii)	Step response obtained from the Matlab code	/1
(iii)	$\Delta v$	/1
(iv)	SSE	/1
9.	Task 3b:	
(i)	Full Matlab code for simulation of step response of $-Z_{out\_CL}$	/1
(ii)	Step response obtained from the Matlab code	/1
(iii)	$\Delta v$	/1
(iv)	SSE	/1
10.	Task 4a:	
(i)	PECS schematic for simulation of step response of $G_{vg\_CL}$	/1
(ii)	Step response obtained from PECS	/1
(iii)	$\Delta v$	/1
(iv)	SSE	/1
11.	Task 4b:	
(i)	PECS schematic for simulation of step response of $-Z_{out\_CL}$	/1
(ii)	Step response obtained from PECS	/1
(iii)	$\Delta v$	/1
(iv)	SSE	/1
12.	Task 5b:	
(i)	Lab screen shot of step response of $-Z_{out\_CL}$	/1
(ii)	$\Delta v$	/1
(iii)	SSE	/1
13.	Task 6a:	
(i)	Loop stability margins and frequencies table	/1
14.	Task 6b:	
(i)	$\Delta v$ and SSE summary table	/1
(ii)	Observations on results	/2
15.	Task 7:	
(i)	Step response difference with proportional and lead compensator with the integral compensator	/2
16.	Task 8: Optional compensator – proportional compensator: if submitted grade with identical grading sheet	/2
17.	Task 9: Optional compensator – lead compensator: if submitted grade with identical grading sheet	/2
Report:		/5
<b>Total:</b>		/55

## Lab 6 Grading Sheet

1. In Lab demonstration of switched load feedback voltage regulator \_\_\_\_\_/10
2. Task 1:
  - (i) Full documentation of your design procedure (do not copy plots from anywhere) \_\_\_\_\_/20
  - (ii) Compensator parameters from simplified equations derived from your asymptotic plots \_\_\_\_\_/5
  - (iii) model of the PWM modulator use a pk-pk ramp amplitude value measured in Lab.4 \_\_\_\_\_/1
3. Task 2a:
  - (i) Determine the loop gain transfer function \_\_\_\_\_/1
  - (ii) Matlab *margin* plot in Hz \_\_\_\_\_/1
  - (iii) margins and associated frequencies \_\_\_\_\_/1
4. Task 2b:
  - (i) Expression for  $G_{vg\_CL}$  \_\_\_\_\_/1
  - (ii) Matlab code to produce *bodemag* plots of  $G_{vg\_CL}$  and (open loop)  $G_{vg}$  /1
  - (iii) Matlab *bodemag* plots of  $G_{vg\_CL}$  and (open loop)  $G_{vg}$  \_\_\_\_\_/1
  - (iv) (iii) in frequency range 1 Hz to 10 kHz \_\_\_\_\_/1
5. Task 2c:
  - (i) Expression for  $-Z_{out\_CL}$  \_\_\_\_\_/1
  - (ii) Matlab code to produce *bodemag* plots of  $-Z_{out\_CL}$  and (open loop)  $-Z_{out}$  /1
  - (iii) Matlab *bodemag* plots of  $-Z_{out\_CL}$  and (open loop)  $-Z_{out}$  \_\_\_\_\_/1
  - (iv) (iii) in frequency range 1 Hz to 10 kHz \_\_\_\_\_/1
6. Task 3a:
  - (i) Full Matlab code for simulation of step response of  $G_{vg\_CL}$  \_\_\_\_\_/1
  - (ii) Step response obtained from the Matlab code \_\_\_\_\_/1
  - (iii)  $\Delta v$  \_\_\_\_\_/1
  - (iv) SSE \_\_\_\_\_/1
7. Task 3b:
  - (i) Full Matlab code for simulation of step response of  $-Z_{out\_CL}$  \_\_\_\_\_/1
  - (ii) Step response obtained from the Matlab code \_\_\_\_\_/1
  - (iii)  $\Delta v$  \_\_\_\_\_/1
  - (iv) SSE \_\_\_\_\_/1
8. Task 4a:
  - (i) PECS schematic for simulation of step response of  $G_{vg\_CL}$  \_\_\_\_\_/1

(ii) Step response obtained from PECS	_____	/1
(iii) $\Delta v$	_____	/1
(iv) SSE	_____	/1
9. Task 4b:		
(i) PECS schematic for simulation of step response of $-Z_{out\_CL}$	_____	/1
(ii) Step response obtained from PECS	_____	/1
(iii) $\Delta v$	_____	/1
(iv) SSE	_____	/1
10. Task 5b:		
(i) Lab screen shot of step response of $-Z_{out\_CL}$	_____	/2
(ii) $\Delta v$	_____	/1
(iii) SSE	_____	/1
11. Task 6a:		
(i) Loop stability margins and frequencies table	_____	/4
12. Task 6b:		
(i) Large results summary table	_____	/15
13. Task 6c:		
(i) Observations on results	_____	/5
Report:	_____	/9
<b>Total:</b>	_____	/100

Part IV

Modern Control



# Chapter 17

## Introduction

There are three main problems that can be examined in the study of systems in the controls context: system dynamics, system identification or modeling, and system control. They develop from the three aspects that are present in the block diagram of a basic system: input, system, and output [1]. Typically, two of the three aspects are known, and the third must be determined from the other two.

In system control, the system is known, and the input to the system that produces a desired output must be determined. Part II focuses on the fundamental control problem of regulation for disturbance rejection as it pertains to DC-DC converters, and uses the Ćuk DC-DC converter as the platform for applying the various design steps, leading up to a minimal-order compensator design, demonstrated in Chapter 23. It is assumed that the reader of this part of the book is familiar with the classical control system design techniques presented earlier as the modern control design methods build on a classical foundation. Note that for the regulation problem, the desired output value is fixed, whereas in the servo problem, the desired output is to track a changing setpoint.

The Ćuk DC-DC converter was chosen as an example system for two main purposes. This nonlinear switching circuit can act to raise (boost) or lower (buck) the voltage from input to output, making it a generic DC-DC converter (compared to converters that can only boost or buck the input voltage, but not do both). Also, the circuit contains four energy storage devices leading to a fourth-order system, which creates sufficient complexity in an output feedback compensator to require compensator order reduction. This allows an original idea regarding order reduction to be presented. Boost or buck converters are typically implemented with only two energy storage components, and the resulting simplicity in compensators designed using modern control techniques either does not require model reduction or renders one of the techniques presented in this part of the text practically useless.

Analysis of the Ćuk converter circuit begins in Chapter 18. The control system analysis and design procedures use MATLAB from The Mathworks, Inc., and code is presented at the end of each chapter for the relevant chap-

ter material. The nonlinear Ćuk switching circuit is modeled as a small-signal continuous linear time invariant (LTI) system using state space averaging. The LTI model to be used during the design process is validated by transient comparison with a nonlinear circuit simulation to justify the assumption that the small-signal model would be adequate. The open-loop performance characteristics are tested, and a set of performance criteria for the closed-loop controlled system are specified.

Chapter 19 covers pole placement using state feedback. This chapter uses desired poles given by a filter prototype that is optimal with respect to an integral performance index and discusses how to select a weighting parameter that determines the closed-loop pole locations.

Integral augmentation of the state feedback architecture is described in Chapter 20. This allows the closed-loop system to completely eliminate steady state error, which could not be accomplished by state feedback alone.

Chapter 21 discusses state estimation using full- and reduced-order observers to allow for the use of output feedback, as state information is not always available to the designer.

Chapter 22 shows the application of optimal control and estimation using linear quadratic methods. These techniques allow the designer to determine optimal controller and optimal estimator gains. Loop transfer recovery is discussed as a means to recover desirable frequency-domain stability margins that are lost when designing an optimal output feedback compensator.

In order to design compensators that can be constructed from a minimum number of components, Chapter 23 covers order reduction methods. First, a reduced-order optimal compensator with recovered loop gain is designed. This step is followed by applying balanced realization and truncation techniques to eliminate states with little effect on performance, resulting in additional compensator order reduction.

Chapter 24 describes how to implement two of the final compensator designs that were created and shows the difference in analog controller circuit complexity that can arise from only one additional order in the compensator.

Chapter 25 presents the minimal compensator circuit test results from a power electronics simulator that prove the performance of the final controller design exceeded the original design specifications.

Finally, Chapter 26 wraps up Part II of the text with a recap of the methods presented and their results.



# References

- [1] B. A. Ogunnaike and W. H. Ray, *Process Dynamics, Modeling, and Control*. New York, NY: Oxford University Press, 1994.



## Chapter 18

# System Analysis

Prior to designing a controller for a system, the control system designer must understand the system's characteristics. For example, is the system open-loop stable? Are there dominant poles? Are there poles that may be neglected during design? Is the system controllable using the selected inputs? Can an estimator be constructed based on the measured outputs? These types of questions should be answered both intuitively and mathematically prior to embarking on an attempt to design a controller for the system.

As stated in Chapter 17, the Ćuk DC-DC converter is used here as the example system for demonstrating the compensator design processes described in Part II. The starting point is the construction of a mathematical model of the system in MATLAB. The model is a mathematical description of some or all of the behavior of the real-world system that is adequate for performing controller design. State space averaging yields a linear small-signal model for the nonlinear switching system [1]. Additionally, a nonlinear circuit model was created in the Power Electronics Circuit Simulator (PECS) software package in order to validate the performance of the assumed linear model. PECS uses a schematic-based circuit editor and features its own plotting tool, PECSPLOT.

A note on notation: the zeros, poles, and gains of systems discussed in Part IV are in the Evans form, i.e., the coefficient of the highest power of  $s$  in each factored term is unity and the stated gain is not the DC gain of the system. This is in contrast to Bode form, where the constant in each factored term is unity and the DC gain is explicitly stated. Evans form was chosen for convenience, as it is the form used by *zpk* systems in MATLAB.

### 18.1 The Ćuk Converter

The Ćuk converter is a step-down/step-up converter based on a switching boost-buck topology. Essentially, the converter is composed of two sections, an input stage and an output stage. The schematic of the Ćuk converter is presented in Figure 18.1, with component values given in Subsection 18.2.3. (The lowercase

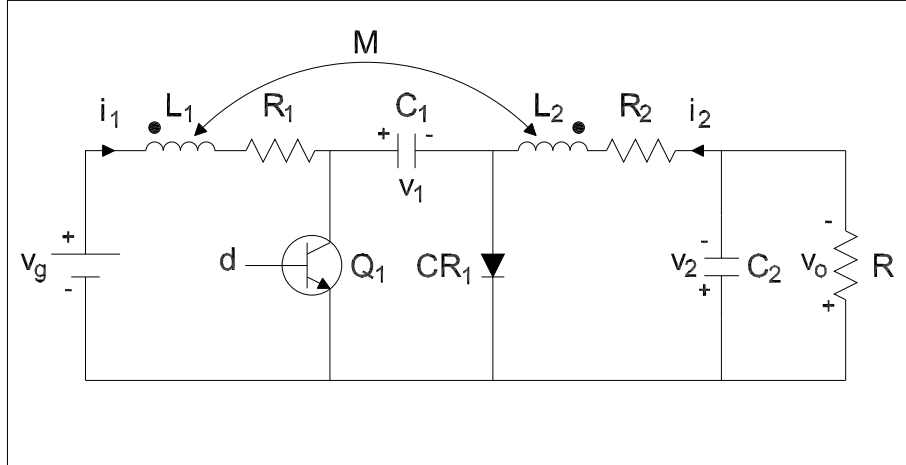


Figure 18.1: Ćuk converter with inductor equivalent series resistances.

variables indicate small-signal deviations from nominal operating point variables as obtained by linearization.) The input voltage  $v_g$  is fed into the circuit via inductor  $L_1$ . When transistor  $Q_1$  is on, current  $i_1$  builds the magnetic field of the inductor in the input stage. The diode  $CR_1$  is reverse biased, and energy dissipates from the storage elements in the output stage. When  $Q_1$  turns off, inductor  $L_1$  tries to maintain the current flowing through it by reversing polarity and sourcing current as its magnetic field collapses. It thus provides energy to the output stage of the circuit via capacitor  $C_1$ . Both currents  $i_1$  and  $i_2$  must sum to zero in the steady state, since the assumption is that voltage  $v_1$  is essentially constant (given that the voltage across a capacitor cannot change instantaneously and the switching speed of the circuit is high). This provides for the following charge conservation relation:

$$i_1 t_{on} + i_2 t_{off} = 0 \quad (18.1)$$

The inductor currents are the input and output currents, therefore, if the principle of conservation of energy is applied:

$$\frac{v_o}{v_g} = \frac{D_s}{1 - D_s} \quad (18.2)$$

where  $D_s$  is the duty cycle of the switch,  $D_s \triangleq \frac{t_{on}}{t_{on} + t_{off}}$ . Equation 18.2 shows that by controlling the duty cycle of the switch (by small-signal deviation  $d$ ), the output voltage  $v_o$  can be controlled and can be higher or lower than the input voltage  $v_g$ . By using a controller to vary the duty cycle during operation, the circuit can also be made to reject disturbances, as will be shown.

## 18.2 The Ćuk Converter Model

The Ćuk converter selected was designed with just enough mutual inductance to avoid a nonminimum phase converter structure, a requirement for a controller design method used later in the text.

### 18.2.1 Analysis of Inductors with Mutual Coupling and Equivalent Series Resistances

The equations for the voltage across two inductors  $L_1$  and  $L_2$  joined by mutual coupling are:

$$\begin{aligned}v_{L_1} &= L_1 \frac{di_1}{dt} + M \frac{di_2}{dt} \\v_{L_2} &= M \frac{di_1}{dt} + L_2 \frac{di_2}{dt}\end{aligned}$$

The solutions of the two simultaneous equations are:

$$\begin{aligned}\frac{di_1}{dt} &= \frac{L_2}{L_1 L_2 - M^2} v_{L_1} + \frac{-M}{L_1 L_2 - M^2} v_{L_2} \\ \frac{di_2}{dt} &= \frac{-M}{L_1 L_2 - M^2} v_{L_1} + \frac{L_1}{L_1 L_2 - M^2} v_{L_2}\end{aligned}$$

For the circuit with equivalent series resistances included in the inductor models when  $Q_1$  conducts:

$$\begin{aligned}v_{L_1} &= v_g - i_1 R_1 \\v_{L_2} &= v_1 - v_2 - i_2 R_2\end{aligned}$$

therefore:

$$\begin{aligned}\frac{di_1}{dt} &= \frac{M}{\sigma^2} v_2 + \frac{-M}{\sigma^2} v_1 + \frac{MR_2}{\sigma^2} i_2 + \frac{-L_2 R_1}{\sigma^2} i_1 + \frac{L_2}{\sigma^2} v_g \\ \frac{di_2}{dt} &= \frac{-L_1}{\sigma^2} v_2 + \frac{L_1}{\sigma^2} v_1 + \frac{-L_1 R_2}{\sigma^2} i_2 + \frac{MR_1}{\sigma^2} i_1 + \frac{-M}{\sigma^2} v_g\end{aligned}$$

with  $\sigma^2 = L_1 L_2 - M^2$ .

When  $Q_1$  does not conduct:

$$\begin{aligned} v_{L_1} &= v_g - i_1 R_1 - v_1 \\ v_{L_2} &= -v_2 - i_2 R_2 \end{aligned}$$

therefore:

$$\begin{aligned} \frac{di_1}{dt} &= \frac{M}{\sigma^2} v_2 + \frac{-L_2}{\sigma^2} v_1 + \frac{MR_2}{\sigma^2} i_2 + \frac{-L_2 R_1}{\sigma^2} i_1 + \frac{L_2}{\sigma^2} v_g \\ \frac{di_2}{dt} &= \frac{-L_1}{\sigma^2} v_2 + \frac{M}{\sigma^2} v_1 + \frac{-L_1 R_2}{\sigma^2} i_2 + \frac{MR_1}{\sigma^2} i_1 + \frac{-M}{\sigma^2} v_g \end{aligned}$$

with  $\sigma^2 = L_1 L_2 - M^2$ .

## 18.2.2 The State Space Averaged Model

State space averaging is a well-known method used in modeling switching converters. For a system with a single switching component with a nominal duty cycle, a model may be developed by determining the state and measurement equations for each of the two switch states, then calculating a weighted average of the two sets of equations using the nominal values of the time spent in each state as the weights. To develop the state space averaged model, the equations for the rate of inductor current change derived in Subsection 18.2.1 are used along with the equations for the rate of capacitor voltage change that may be derived from the Čuk converter circuit. For the purposes of modeling, the state vector is given by:

$$x = [ v_2 \quad v_1 \quad i_2 \quad i_1 ]'$$

When  $Q_1$  conducts, the following state space matrices result:

$$\begin{aligned} A_1 &= \begin{bmatrix} -\frac{1}{RC_2} & 0 & \frac{1}{C_2} & 0 \\ 0 & 0 & \frac{-1}{C_1} & 0 \\ -\frac{L_1}{\sigma^2} & \frac{L_1}{\sigma^2} & -\frac{L_1 R_2}{\sigma^2} & \frac{MR_1}{\sigma^2} \\ \frac{M}{\sigma^2} & -\frac{M}{\sigma^2} & \frac{MR_2}{\sigma^2} & -\frac{L_2 R_1}{\sigma^2} \end{bmatrix} \\ B_1 &= \begin{bmatrix} 0 \\ 0 \\ -\frac{M}{\sigma^2} \\ \frac{L_2}{\sigma^2} \end{bmatrix} \\ C_1 &= [ 1 \quad 0 \quad 0 \quad 0 ] \\ D_1 &= [ 0 ] \end{aligned}$$

When  $Q_1$  is not conducting, the circuit model is represented by the following state space matrices:

$$\begin{aligned}
 A_2 &= \begin{bmatrix} -\frac{1}{RC_2} & 0 & \frac{1}{C_2} & 0 \\ 0 & 0 & 0 & \frac{-1}{C_1} \\ -\frac{L_1}{\sigma^2} & \frac{M}{\sigma^2} & -\frac{L_1 R_2}{\sigma^2} & \frac{MR_1}{\sigma^2} \\ \frac{M}{\sigma^2} & -\frac{L_2}{\sigma^2} & \frac{MR_2}{\sigma^2} & -\frac{L_2 R_1}{\sigma^2} \end{bmatrix} \\
 B_2 &= \begin{bmatrix} 0 \\ 0 \\ -\frac{M}{\sigma^2} \\ \frac{L_2}{\sigma^2} \end{bmatrix} \\
 C_2 &= [1 \ 0 \ 0 \ 0] \\
 D_2 &= [0]
 \end{aligned}$$

By representing the duty cycle of the switch as  $D_s$ , the following results may be obtained for a state space averaged model of the Ćuk converter:

$$\begin{aligned}
 D'_s &\triangleq 1 - D_s \\
 \frac{V_o}{V_g} &= \frac{D_s}{1 - D_s} = \frac{D_s}{D'_s}
 \end{aligned}$$

$$\begin{aligned}
 A &= D_s A_1 + D'_s A_2 \\
 B &= D_s B_1 + D'_s B_2 \\
 C &= D_s C_1 + D'_s C_2 \\
 D &= D_s D_1 + D'_s D_2 \\
 X &= -A^{-1} B V_g \\
 B_d &= (A_1 - A_2) X + (B_1 - B_2) V_g \\
 D_d &= (C_1 - C_2) X + (D_1 - D_2) V_g
 \end{aligned}$$

$$\begin{aligned}
 \dot{\tilde{x}} &= A\tilde{x} + B\tilde{v}_g + B_d\tilde{d} \\
 \tilde{v}_o &= C\tilde{x} + D\tilde{v}_g + D_d\tilde{d}
 \end{aligned}$$

where the tilde ( $\tilde{\cdot}$ ) indicates a small signal deviation from nominal, and:

$$\begin{aligned}
 x &= X + \tilde{x} \\
 v_g &= V_g + \tilde{v}_g \\
 d &= D_s + \tilde{d} \\
 v_o &= V_o + \tilde{v}_o
 \end{aligned}$$

$$\begin{aligned}
A &= \begin{bmatrix} -\frac{1}{RC_2} & 0 & \frac{1}{C_2} & 0 \\ 0 & 0 & -\frac{D_s}{C_1} & \frac{1-D_s}{C_1} \\ -\frac{L_1}{L_1 L_2 - M^2} & \frac{D_s L_1 + M - D_s M}{L_1 L_2 - M^2} & 0 & 0 \\ \frac{M}{L_1 L_2 - M^2} & \frac{-D_s M - L_2 + L_2 D_s}{L_1 L_2 - M^2} & 0 & 0 \end{bmatrix} \\
B &= \begin{bmatrix} 0 \\ 0 \\ -\frac{M}{L_1 L_2 - M^2} \\ \frac{L_2}{L_1 L_2 - M^2} \end{bmatrix} \\
C &= [1 \ 0 \ 0 \ 0] \\
D &= [0] \\
B_d &= \begin{bmatrix} 0 \\ -\frac{D_s V_g}{R(1-D_s)^2 C_1} \\ \frac{V_g (L_1 - M)}{(1-D_s)(L_1 L_2 - M^2)} \\ \frac{V_g (-M + L_2)}{(1-D_s)(L_1 L_2 - M^2)} \end{bmatrix} \\
D_d &= [0]
\end{aligned}$$

The equilibrium state vector is:

$$X = \begin{bmatrix} V_2 \\ V_1 \\ I_2 \\ I_1 \end{bmatrix} = \begin{bmatrix} \frac{D_s V_g}{1-D_s} \\ \frac{V_g}{1-D_s} \\ \frac{D_s V_g}{R(1-D_s)} \\ \frac{D_s^2 V_g}{R(1-D_s)^2} \end{bmatrix}$$



### 18.2.3 Component Values

Here are the Ćuk converter circuit component values used in the remainder of Part IV:

$$\begin{aligned}L_1 &= 0.5 \text{ mH} \\R_1 &= 0.01 \Omega \\L_2 &= 7.5 \text{ mH} \\R_2 &= 0.01 \Omega \\M &= -1.5 \text{ mH} \\C_1 &= 2.0 \mu\text{F} \\C_2 &= 20 \mu\text{F} \\R &= 30 \Omega \\V_g &= 12 \text{ V} \\d &= 0.667 \\f &= 100 \text{ kHz}\end{aligned}$$

## 18.3 Ćuk Converter Open Loop Performance

Before a controller was designed, the performance of the open-loop Ćuk model was examined. A state space block diagram for the open-loop model is shown in Figure 18.2. The state space equations were determined to be:

$$\begin{aligned}\dot{x} &= Ax + Bv_g + B_d d & (18.3) \\v_o &= Cx \\x &= [v_2 \quad v_1 \quad i_2 \quad i_1]'\end{aligned}$$

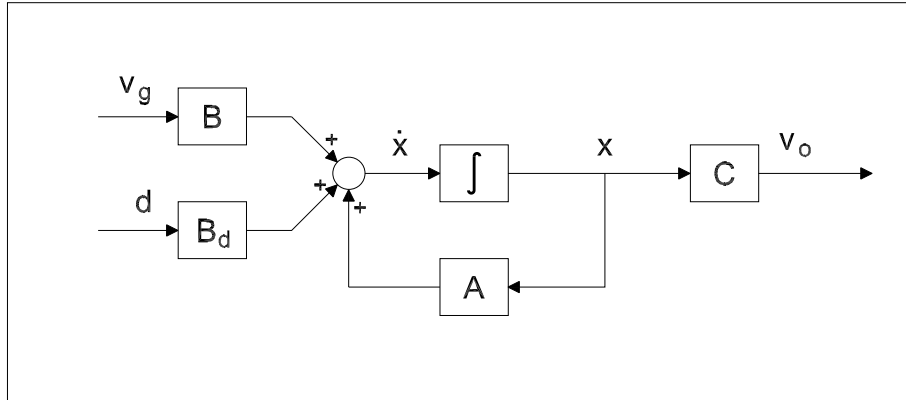


Figure 18.2: State space model of the Ćuk converter.

The state space matrices for the open-loop model from the disturbance input  $v_g$  to the output  $v_o$  are the state space averaged matrices  $\{A, B, C, D\}$ . The state space matrices for the open-loop system from the control input  $d$  to the output  $v_o$  are the state space averaged matrices  $\{A, B_d, C, D\}$ . Thus, the model of the Ćuk converter has two inputs (a control input  $d$  and a disturbance input  $v_g$ ) and one output ( $v_o$ ).

The MATLAB model open-loop response to a unit step disturbance in  $v_g$  is shown in Figure 18.3. By inspection of the plotted response, it was determined that the system reached lightly damped oscillations around a steady state DC value in approximately 20 ms. The steady state value was 26 V, a value predicted from the gain equation for the Ćuk converter:

$$v_o = \frac{D_s}{1 - D_s} v_g \quad (18.4)$$

With nominal duty cycle  $D_s = 0.667$ , a 1 V step input in  $v_g$  produces a 2 V step in the output voltage  $v_o$ . This shows that the open-loop system does not reject disturbances on the input voltage  $v_g$ . Also, note that the output of the circuit is a lightly damped sinusoid, with an approximate frequency of 1.83 kHz (11.5 krad/s).

The PECS circuit is shown in Figure 18.4. The simulator was set up to check the performance of the nonlinear converter in response to a unit step up in  $v_g$ .

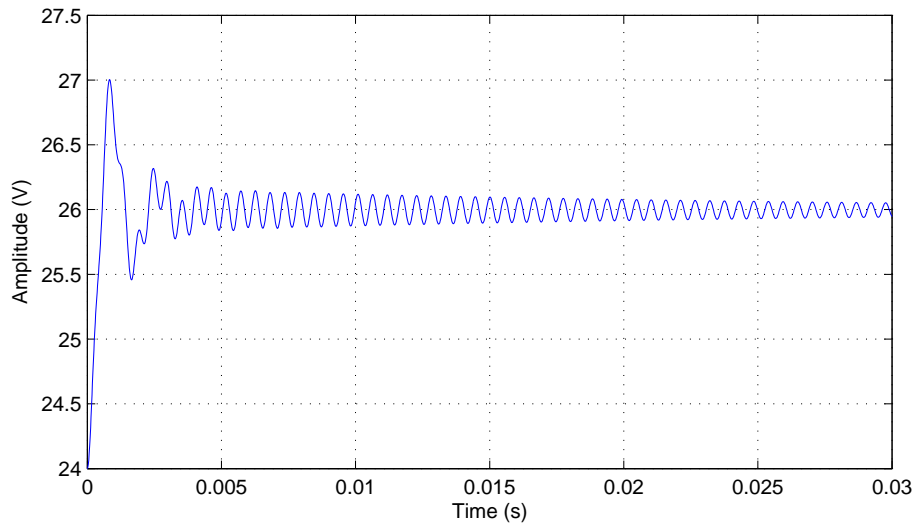


Figure 18.3: Ćuk converter output voltage response to a unit step disturbance in  $v_g$ .

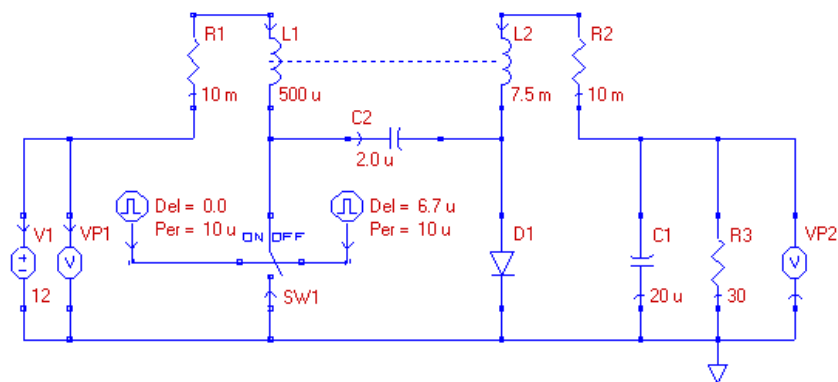


Figure 18.4: The Ćuk converter simulated in PECS.

The PECS plot of these transients is shown in Figure 18.5. Comparison of the MATLAB and PECS plots reveals that the linear model used in MATLAB is an acceptable model of the plant to use for control system design.

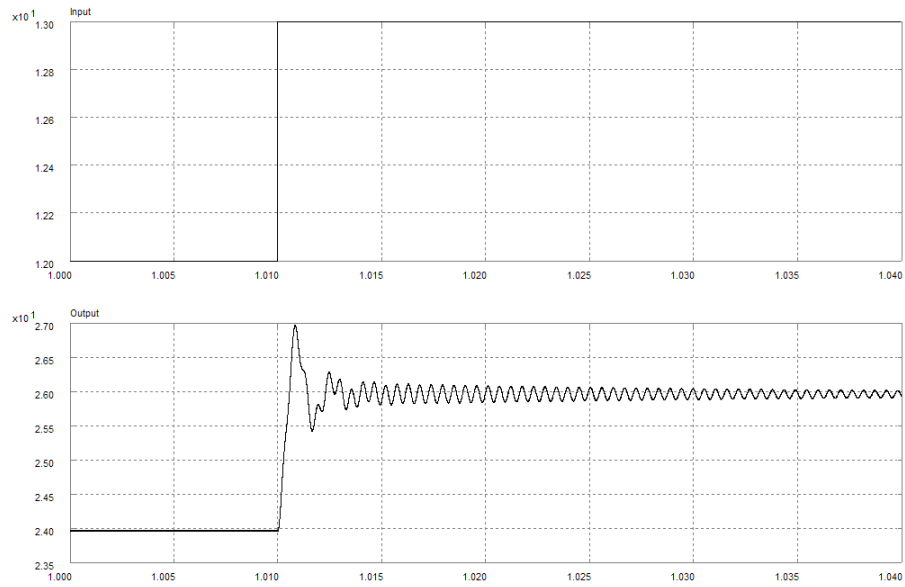


Figure 18.5: Unit step response of the Ćuk converter in PECS.

The pole-zero plot of  $T_{v_{od}}$  is shown in Figure 18.6. All poles and zeros are in the LHP, therefore the Ćuk converter is a stable minimum-phase system. The locations for the zeros and poles are:

$$z = [ -1490 \pm j9000 ]$$

$$p = [ -879 \pm j3641, -40 \pm j11500 ]$$

where the zeroes and poles have units of rad/s. The 1.83 kHz ringing in the output transient caused by the unit step disturbance is due to the frequency associated with the dominant pole pair at  $-40 \pm j11500$  rad/s.

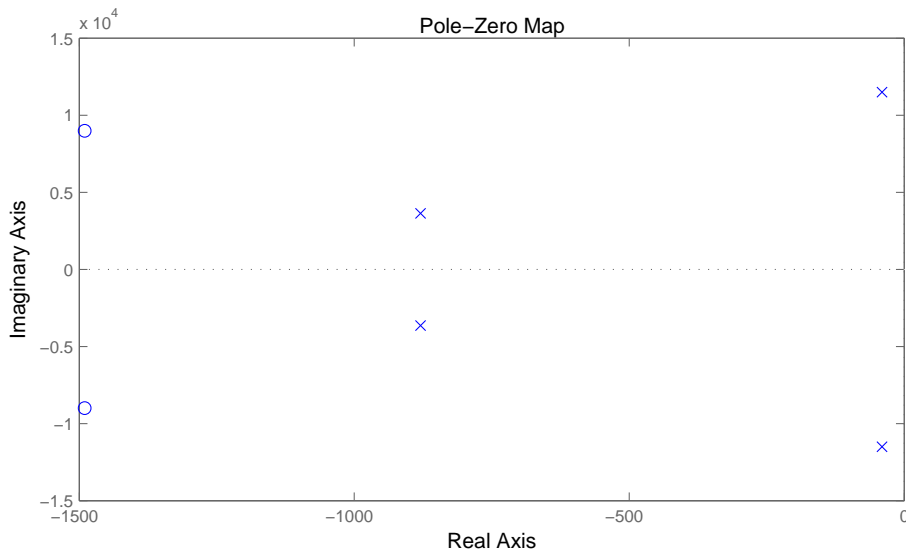


Figure 18.6: Map of the pole and zero locations of  $T_{v_o v_g}$ .

## 18.4 Controllability and Stabilizability

The idea of controllability refers to the ability of the input control  $u$  to affect the system dynamics. Controllability is defined as the ability to move the state of a system from an initial value  $x_0$  to any arbitrary state  $x_f$  within a finite time period  $t$  using the input signal  $u$ . (Note that controllability says nothing about the magnitude of the input signal  $u$ , i.e., the control effort, nor the time  $t$  required to accomplish this transition.) Essentially, it is a test to determine if the closed-loop system poles may be arbitrarily placed in the complex plane.

The controllability matrix  $M_c$  is constructed from  $(A, B)$  in the following manner:

$$M_c = [ B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B ] \quad (18.5)$$

where  $n$  is the order of the system. If  $M_c$  is a full rank matrix, the system is fully controllable. The rank deficiency of  $M_c$  tells the designer how many modes are uncontrollable. There is no rank deficiency in  $M_c$  for the Ćuk converter model, therefore the system is fully controllable.

## 18.5 Observability and Detectability

Observability refers to the ability to determine any initial state  $x_0$  using only a finite record of the output  $y$  between an initial time and a final time. The

observability matrix  $M_o$  is constructed from  $(A, C)$  in the following manner:

$$M_o = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (18.6)$$

where  $n$  is the order of the system. If  $M_o$  is a full rank matrix, the system is fully observable. The rank deficiency of  $M_o$  tells the designer how many modes are unobservable. The Ćuk converter model is fully observable.

## 18.6 Controlling the Ćuk Converter

The model used in controller design is a small-signal model, since, like many other methods of linearization, the state-space averaging method only holds for small deviations from the nominal operating point. Most of the equations and figures that follow refer to deviations from the nominal operating point of the system unless otherwise stated or identifiable from context.

The control system designer must always begin with a set of design specifications when starting a project. The specifications are a set of goals for the behavior of the controlled system, and may need to change during the design process if not achievable or as new information becomes available. Specifications generally consider both transient behavior (e.g., rise time, settling time, percent overshoot) and stability margins (e.g., relative stability, gain margin, phase margin).

### 18.6.1 Time Domain Specifications

Time domain constraints are given by the system performance specifications. The transient response of a regulated system is typically limited in terms of both maximum amplitude deviation from the nominal output and settling time in response to a transient. The goal for the Ćuk converter controller design example is to control the output voltage to within 1% of nominal (i.e., 23.76 to 24.24 V) in response to unit step voltage disturbances in the input. This matches the 1% regulation of standard industrial power supplies sold by a major control system equipment manufacturer. Also, the controller should be able to maintain the nominal output voltage within tolerances as the input varies over a range of 9 to 14 V, though this shall be considered a steady-state, not transient, operating requirement. As a final specification, steady-state error in the output voltage shall be eliminated within 20 milliseconds of the start of a transient.

### 18.6.2 Frequency Domain Specifications

The frequency response of the transfer function  $T_{v_o,d}$  should be high at low frequencies for proper regulation and low at high frequencies for adequate noise

rejection. The example system base switching frequency is 100 kHz ( $6.28 \times 10^5$  rad/s). As the small signal model breaks down above half of the switching frequency, the loop gain at any frequency above 50 kHz ( $3.14 \times 10^5$  rad/s) should be less than 0 dB. Indeed, there should be a design margin left between this frequency and the gain crossover frequency. A gain margin of at least 20 dB and a phase margin of at least  $50^\circ$  will be sought to ensure stability.

### 18.6.3 Control Effort Constraints

The Ćuk converter nominal duty cycle is related to the steady-state gain of the converter  $G$  by Equation 18.4. Neglecting circuit losses, Equation 18.4 may be rearranged to calculate the duty cycle as a function of the output operating point and input voltages,  $v_o$  and  $v_g$ :

$$D_s = \frac{v_o}{v_o + v_g} \quad (18.7)$$

Therefore, the nominal duty cycle at the operating point of 24 V for an input of 12 V is determined to be 0.667. However, the purpose of controller design is to ensure the output voltage remains within 1% of 24 V despite disturbances in the input voltage. Since a deviation model is used, the difference between the nominal operating duty cycle and the duty cycle required to keep the output at exactly 24 V may be approximated, and this is shown in Figure 18.7.

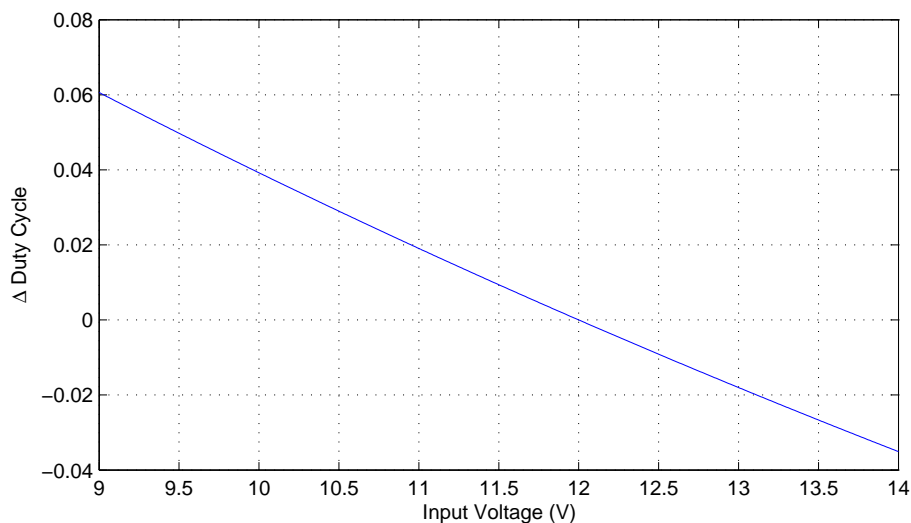


Figure 18.7: The small-signal duty cycle required over the full input voltage range to maintain nominal output voltage.

It is this change in duty cycle that the controller must provide, as the deviation in duty cycle is the small-signal control input of the Ćuk converter. Thus,

it can be predicted from Equation 18.7 or Figure 18.7 that the the controller must change the duty cycle by -0.018 to maintain the output at 24 V for a step disturbance input on  $v_g$  from the nominal 12 V to 13 V. This value of -0.018 will be used to verify the correct steady-state control effort in controller design. (Note that Equation 18.7 does not account for any voltage losses within the circuit, so the duty cycle will actually be slightly higher from the calculated value when any resistances are included in the circuit.) The duty cycle is limited to  $0 \leq D_s \leq 1$ , therefore if the control effort plus the nominal value of 0.667 exceeds these limiting values, the compensator design is not acceptable. This leads to hard limits on the small-signal control effort of  $[-0.667, 0.333]$ , though the inclusion of a design margin to these limits may be desirable. It is up to the individual designer to choose constraints on the control effort, however it is generally best to allow the use of as much of the control effort range as possible.



## 18.7 MATLAB Code

```
1 % SSA Model of Minimum-Phase Cuk Converter
2
3 % Define physical inputs
4 clear
5 close all
6
7 Vg = 12;
8 Vo = 24;
9 r = 28;
10 c1 = 2*10^-6;
11 c2 = 2*10^-5;
12 d = Vo/(Vg+Vo);
13 l1 = 5*10^-4;
14 r1 = 0.01;
15 l2 = 7.5*10^-3;
16 r2 = 0.01;
17 m = -1.5*10^-3;
18
19 % Calculate state variables for each switch position
20 s2 = l1*l2-m^2;
21 A1 = [-1/(r*c2) 0 1/c2 0;
22 0 0 -1/c1 0;
23 -l1/s2 l1/s2 -l1*r2/s2 m*r1/s2;
24 m/s2 -m/s2 m*r2/s2 -l2*r1/s2];
25 B1= [0; 0; -m/s2; l2/s2];
26 B2 = B1;
27 A2 = [-1/(r*c2) 0 1/c2 0;
28 0 0 0 1/c1;
29 -l1/s2 m/s2 -l1*r2/s2 m*r1/s2;
30 m/s2 -l2/s2 m*r2/s2 -l2*r1/s2];
31 C1 = [1 0 0 0];
32 C2 = C1;
33 D = 0;
34
35 % Calculate combined state variables given duty cycle d
36 d1 = 1-d;
37 A = d*A1+d1*A2;
38 B = d*B1+d1*B2;
39 C = d*C1+d1*C2;
40
41 % Calculate B matrix for control input - Bd
42 X = -A^-1*B*Vg;
43 Bd = (A1-A2)*X;
44
45 % Define the state space model from vg to vo
46 SYS = ss(A,B,C,D);
47
48 % Define the state space model from d to vo
49 SYSd = ss(A,Bd,C,D);
50
51 % Unit step response from disturbance input vg to vo
52 figure;
53 [y t] = step(SYS,0.03);
```

```

54 plot(t, Vo+y);
55 xlabel('Time (s)');
56 ylabel('Amplitude (V)');
57 grid on
58
59 % Pole-zero map from d to vo
60 figure;
61 pzmap(SYSd);
62 h = gcr;
63 h.AxesGrid.TitleStyle.FontSize = 10;
64 h.AxesGrid.XLabelStyle.FontSize = 10;
65 h.AxesGrid.YLabelStyle.FontSize = 10;
66 grid on
67
68 % Open-loop Bode Plot from d to vo
69 figure;
70 margin(SYSd);
71 h = gcr;
72 h.AxesGrid.Xunits = 'Hz';
73 h.AxesGrid.TitleStyle.FontSize = 10;
74 h.AxesGrid.XLabelStyle.FontSize = 10;
75 h.AxesGrid.YLabelStyle.FontSize = 10;
76
77 clear A1 A2 B1 B2 C1 C2% X s2 r c1 c2 l1 l2 m d1

```

```

1 % Duty Cycle vs. Vg for Cuk Converter
2
3 % Define the plant using an external file
4 cuk_parameters;
5
6 % Define the range of input voltage to sweep
7 vg = 9:0.01:14;
8
9 d_new = Vo./(vg + Vo);
10 Δ_d = d_new-d;
11
12 % Plot change in duty cycle as vg is swept to keep Vo at 24 V
13 figure;
14 plot(vg, Δ_d);
15 xlabel('Input Voltage (V)');
16 ylabel('\Delta Duty Cycle');
17 axis([9 14 -0.04 0.08]);
18 grid on
19
20 % Plot duty cycle as vg is swept to keep Vo at 24 V
21 figure;
22 plot(vg, d_new);
23 xlabel('Input Voltage (V)');
24 ylabel('Duty Cycle');
25 axis([9 14 0.62 0.74]);
26 grid on

```

# References

- [1] G. C. Verghese, "Dynamic modeling and control in power electronics," in *The Control Handbook*, W. S. Levine, Ed. Boca Raton, FL: CRC Press LLC, 1996, ch. 78.1, pp. 1413–1424.



## Chapter 19

# Pole Placement

For a system that is completely controllable and where all the states are accessible, feedback of all of the states through a gain matrix can be used to place the poles at any desired location in the complex plane. The control law used for state feedback is:

$$u = -Kx \quad (19.1)$$

which uses the matrix  $K$  to place the poles of the system at desired locations [1]. This type of compensator is said to employ full state feedback (FSFB). A FSFB regulator is shown in Figure 19.1.

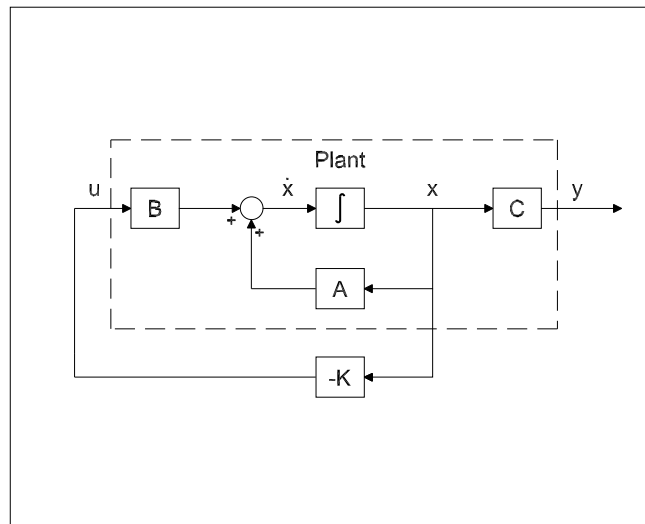


Figure 19.1: State feedback regulation.

## 19.1 Pole Placement via Ackermann's Formula

Ackermann's formula may be used with single-input, single-output (SISO) systems like the Ćuk converter. Ackermann's formula is:

$$K = [ 0 \ 0 \ \dots \ 1 ] M_c^{-1} (A^n + \alpha_1 A^{n-1} + \dots + \alpha_{n-1} A + \alpha_n I) \quad (19.2)$$

This method of determining  $K$  may be used with the system in any representation. It is this method of pole placement that is used in the designs of the state feedback controllers that follow.

## 19.2 Ćuk Converter with State Feedback Compensator

One problem with pole placement is how to go about selecting desirable pole locations. Two main methods of design are commonly followed [2]:

- Select pole locations such that a dominant complex pole pair exists. This technique is generally used when designing tracking systems, for which the transient time domain requirements (e.g., rise time, overshoot, settling time, etc.) are able to be recast into desired dominant pole locations.
- Select pole locations that have been determined to give a prototype time-domain response, e.g., filter pole locations.

The latter method is used in this chapter for pole placement with full state feedback control.

Graham and Lathrop [3] discuss assigning the system poles of higher-order systems to prototype locations that minimizes a performance index (or cost function) known as the integral of the time-weighted absolute error (ITAE) to an input signal:

$$J_{ITAE} = \int_0^{\infty} t |e(t)| dt \quad (19.3)$$

By placing poles in an ITAE filter pattern to minimize  $J_{ITAE}$ , the designer achieves a response that is optimized with respect to deviation from setpoint (provided by the absolute error) and settling time (errors that occur later in the time history contribute more to the  $J_{ITAE}$  cost). Since the goal of the control system designer is to regulate the Ćuk converter output voltage with respect to input voltage disturbances,  $J_{ITAE}$  provides a scalar figure of merit by which to judge controller performance. For regulator problems, the desired output is rejection of disturbance deviations from the nominal operating point. The error between the desired output and the plant output is defined as  $e(t) = r(t) - y(t)$ . Since  $r(t) = 0$  for all time  $t$  in a regulator problem, the error  $e(t)$  is simply  $-y(t)$ .

The frequency-normalized pole locations for minimum ITAE response are given in Table 19.1 up through order five (so that a full-order state feedback

controller with an integrator may be applied to the Ćuk converter), where  $\omega$  is a multiplier that multiplies each pole location in the array of poles listed to convert from frequency-normalized pole locations to desired pole locations<sup>1</sup>. The frequency multiplier  $\omega$  moves the poles outward along radii extending from the origin of the complex plane since it multiplies each real and complex value in a given pole by the same amount, increasing the natural frequency of each pole. The control system designer must determine the value of  $\omega$  that places the poles in such a way as to achieve the desired time domain response. A computer program (e.g., MATLAB from The Mathworks with the Control Systems Toolbox) can be used to iteratively design and test state feedback controllers over a range of values for the scalar multiplier frequency  $\omega$  using a simple search algorithm. where all pole values are in rad/s. By using the steady state error

Table 19.1: Frequency-Normalized Pole Locations for ITAE Response

Order	Factored Pole Locations with Multiplier
1	$\omega[-1]$
2	$\omega[-0.7071 \pm j0.7071]$
3	$\omega[-0.7081, -0.521 \pm j1.068]$
4	$\omega[-0.424 \pm j1.263, -0.626 \pm j0.4141]$
5	$\omega[-0.8955, -0.3764 \pm j1.292, -0.5758 \pm j0.5339]$

as a measurement metric for each iteration during the design,  $m$  can be chosen that produces a steady state error within the performance specification of 1%. Figure 19.2 shows a plot of steady-state error vs.  $\omega$ , and this figure was used to select a value of  $\omega$  that corresponded to 0.24 V (1% voltage regulation). Initially, a wide range of frequencies was selected with a large increment, then the range and the increment were made smaller in order to narrow in on the first frequency with less than 0.24 V of steady state error, which occurs at  $\omega = 10115$ . This prevents moving the poles farther into the left half of the complex plane than necessary to achieve the desired performance. Moving poles too far results in increases in control effort that may see the control effort hit a saturation limit in actual systems, e.g., if the duty cycle were to try to go below 0 or above 1, it could not. The unit step disturbance response of the system with a full state feedback controller designed in this manner is shown in Figure 19.3. Note that there is no more than 0.24 V of maximum error to the 1 V step disturbance in input voltage, indicating that the disturbance is rejected to within the performance specifications. The amplitude and settling time of the transient meet design specifications, so this controller has very desirable time-domain response characteristics. The loop gain of the regulated system is shown in Figure 19.4, where the loop is broken at the large X shown on the control input  $d$  in Figure 19.5.

<sup>1</sup>Frequency-normalization in this case refers to the fact that the minimum natural frequency given by the pole locations for any of the  $n$ th-order set of poles listed (without the frequency multiplier  $\omega$ ) is equal to 1 rad/s

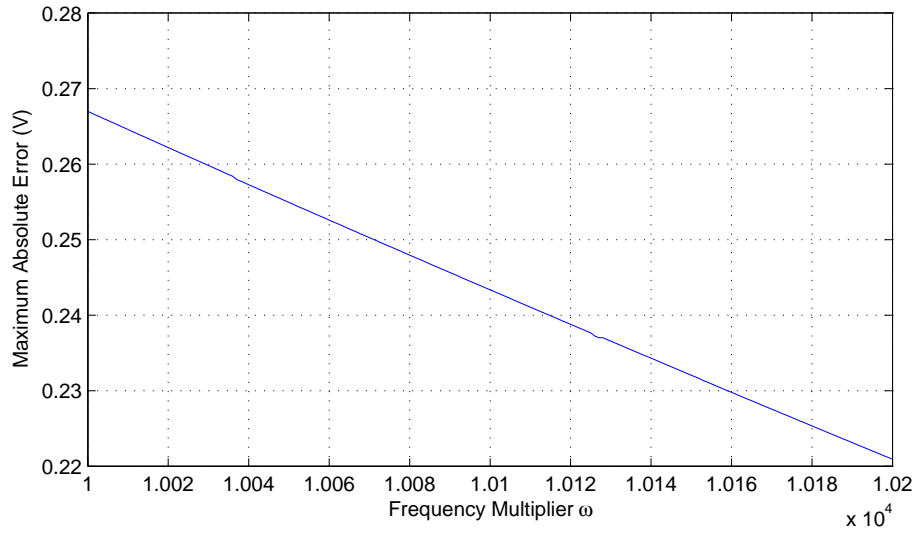


Figure 19.2: Maximum absolute error of unit step disturbance as frequency multiplier is swept.

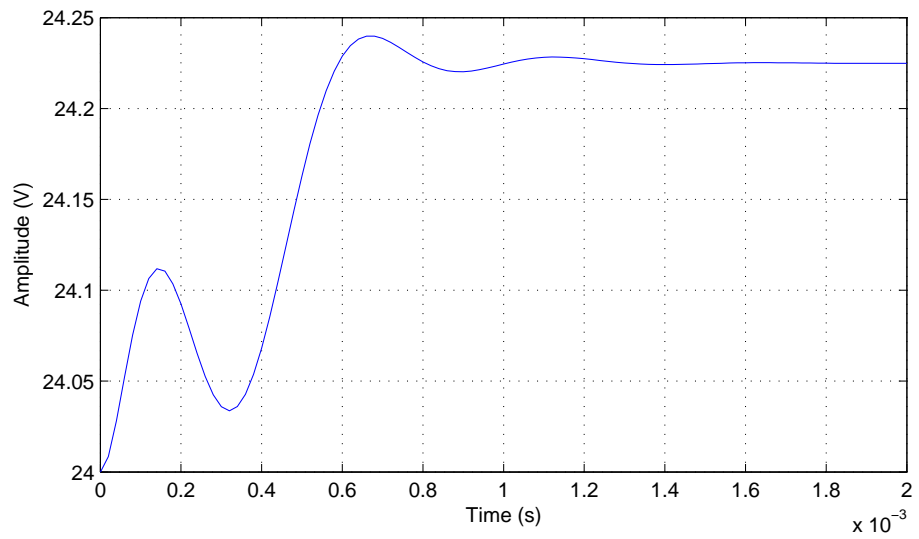


Figure 19.3: Response to unit step disturbance of input voltage.



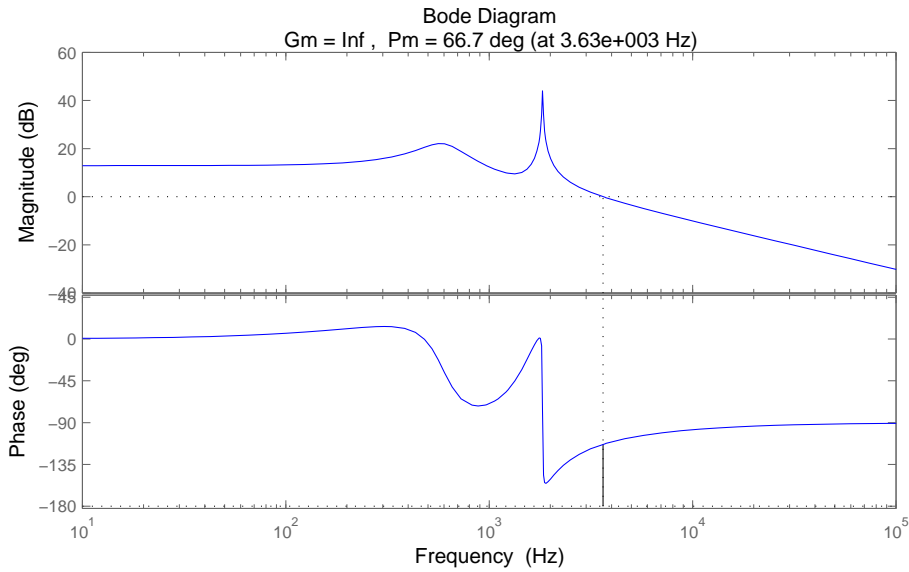


Figure 19.4: Loop gain with the full state feedback controller.

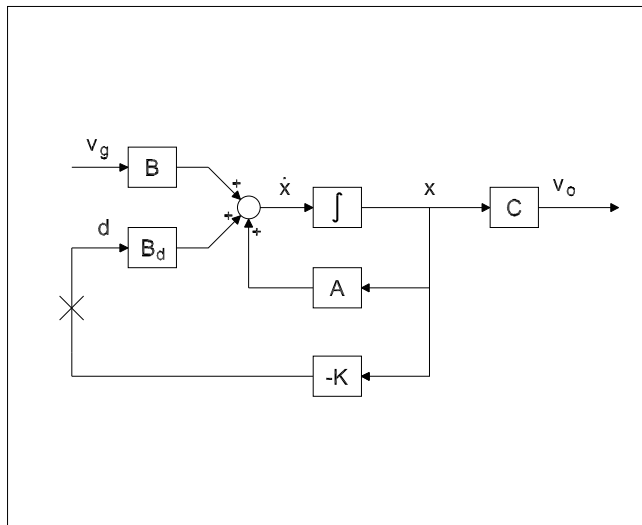


Figure 19.5: The full state feedback controller applied to the Ćuk converter.

MATLAB calculations give the gain margin as  $\infty$  and a phase margin of  $66.7^\circ$ , which are very desirable frequency-domain response characteristics.

Finally, the control effort of the design should be examined. Once again, the control input to the Ćuk converter is the change in duty cycle  $d$  used to turn

on and off  $Q1$ . The control effort is plotted in Figure 19.6, and it can be seen that the effort is not approaching the limits assigned to  $d$ .

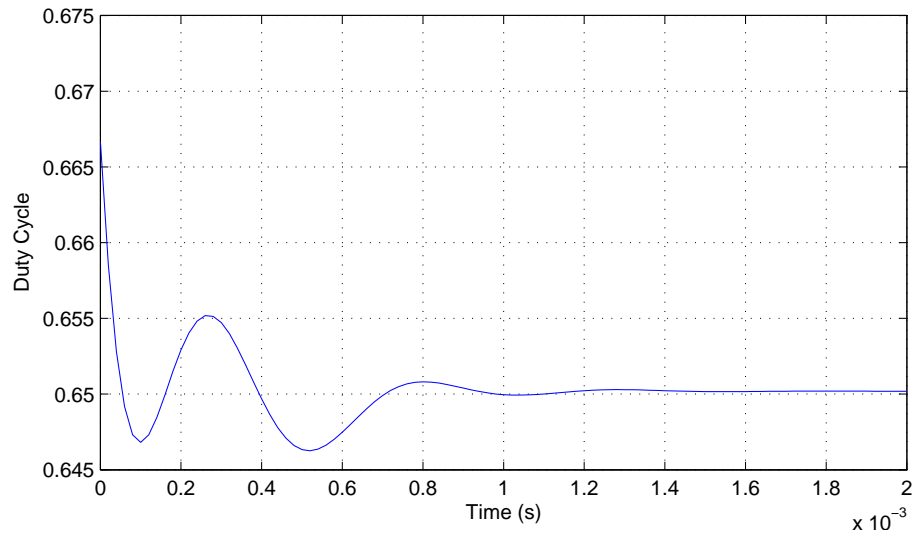


Figure 19.6: Control effort with the full state feedback controller.

It can also be seen that the steady-state deviation control effort is approximately -0.0163, which corresponds roughly to calculations using Equation 18.7.

## 19.3 MATLAB Code

```
1 % Full State Feedback Compensator
2
3 % Define the plant using an external file
4 cuk_parameters;
5
6 % Define ITAE normalized pole positions
7 ITAE_4 = [-0.4240+1.2360i, -0.4240-1.2360i, ...
8 -0.6260+0.4141i, -0.6260-0.4141i];
9
10 % Sweep scaling multiplier and find first multiplier that gives ...
    max abs error
11 % ≤ 0.24 V
12 w0 = [10000:1:10200];
13 w0_length = length(w0);
14 maxerr = zeros(size(w0));
15
16 wb = waitbar(0, 'Sweeping ITAE Filter Scaling Multiplier...');
17
18 for index = 1:length(w0);
19     P = w0(index) * ITAE_4;
20     k = place(A, Bd, P);
21     Abar = [A-Bd*k];
22     SYS1 = ss(Abar, B, C, D);
23     [y t] = step(SYS1, 0.003);
24     maxerr(index) = max(abs(y));
25     waitbar(index/w0_length, wb)
26 end
27 close(wb);
28
29 % Plot the maximum absolute error as a function of the frequency ...
    multiplier
30 figure;
31 h = plot(w0, maxerr);
32 xlabel('Frequency Multiplier \omega');
33 ylabel('Maximum Absolute Error (V)');
34 grid on
35
36 % Determine vector of scaling values that results in max abs ...
    error ≤ 0.24 V
37 z = find(maxerr ≤ 0.24);
38 % Select first element in that vector to use as scaling function
39 w = w0(z(1,1));
40 sprintf('ITAE pole scaling multiplier selected, FSFB: %4.0f', w)
41
42 % Use first scaling frequency with steady state error ≤ 0.24 to ...
    scale the
43 % ITAE filter pole positions
44 P = w * ITAE_4;
45 k = place(A, Bd, P);
46 Abar = [A-Bd*k];
47 SYS_FSF = ss(Abar, B, C, D);
48
49 % Plot closed-loop system response to a unit step disturbance input
```

```

50 [y t x] = step(SYS_FSFb,0.002);
51 figure;
52 plot(t, Vo+y);
53 xlabel('Time (s)');
54 ylabel('Amplitude (V)');
55 grid on
56
57 % System Loop Gain Plot
58 figure;
59 margin(A, Bd, k, 0);
60 h = gcr;
61 h.AxesGrid.TitleStyle.FontSize = 10;
62 h.AxesGrid.XLabelStyle.FontSize = 10;
63 h.AxesGrid.YLabelStyle.FontSize = 10;
64
65 % Control Effort Plot
66 figure;
67 plot(t, -k*x'+d);
68 xlabel('Time (s)');
69 ylabel('Duty Cycle');
70 grid on

```

# References

- [1] R. T. Stefani, B. Shahian, C. J. Savant, Jr., and G. H. Hostetter, *Design of Feedback Control Systems*, 4th ed. New York, NY: Oxford University Press, 2002.
- [2] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 3rd ed. Boston, MA: Addison-Wesley, 1993.
- [3] D. Graham and R. C. Lathrop, "The synthesis of optimum transient response: Criteria and standard forms," *Trans. AIEE*, vol. 72, pp. 273–288, Nov. 1953.



## Chapter 20

# Integral Action

The previous state feedback design for the Ćuk converter resulted in a maximum error of 0.24 V of due to the 1 V disturbance in input voltage, which is just within design specifications. The steady state error was 0.22 V. Additional gain could reduce this error, though it could never be eliminated, as the Ćuk converter is a type 0 system, which means that there will always be some finite steady-state error to a unit step disturbance or setpoint change, even in a controlled system, no matter how high the gain. However, it is desirable to eliminate steady-state error to the unit step entirely if possible. The only way to do this is to have the controller raise the type number<sup>1</sup>. Full state feedback does not introduce an integrator into the closed loop, therefore does not change the type number.

### 20.1 Adding Integrators

In order to eliminate any steady-state offset that may occur, an integrating controller may be added to the controlled system. Integral control is a method of output feedback, as shown in Figure 20.1.

---

<sup>1</sup>The concept of type number goes hand-in-hand with the internal model principle, which states that a loop must contain a transfer function that has a model of the signal it is trying to reject. For example, if trying to reject a unit step  $\frac{1}{s}$ , at least one transfer function in the loop must contain  $\frac{1}{s}$ .

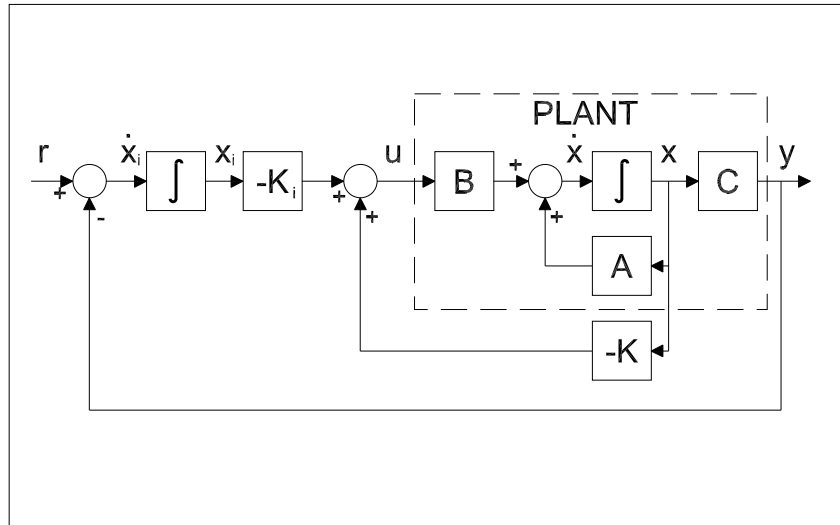


Figure 20.1: Generic system controlled with a FSFB regulator and output integral feedback.

The integrating controller integrates the error between any reference signal and the output  $e(t) = r(t) - y(t)$  and adds it to the state feedback control effort to eliminate steady-state error. The equation for the integrator is  $x_i = \int edt$ , or  $\dot{x}_i = e$ . Since each row in the state space representation is a first-order linear differential equation, and the integrator adds one new differential equation to the system  $\dot{x}_i = -y = -Cx$ , one new state  $x_i$  must be added to the state vector to raise the system from type 0 to type 1. The augmented state vector is  $[x \ x_i]'$  and the new state space quadruple is:

$$\begin{aligned}
 A &= \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \\
 B &= [ B \ 0 ]' \\
 C &= [ C \ 0 ] \\
 D &= 0
 \end{aligned} \tag{20.1}$$

The control law for this augmented system is  $u = -kx - k_i x_i$ . From the above modifications, the desired poles (with an added desired closed-loop pole location to account for the pole associated with the integrator) can be used to determine the state feedback gain, which has the structure  $K = [k \ k_i]$ .



## 20.2 Ćuk Converter with State Feedback and Integral Compensator

The augmented controlled system of the Ćuk converter is shown in Figure 20.2. This control method is described as full state feedback with an integrator (FSFBI).

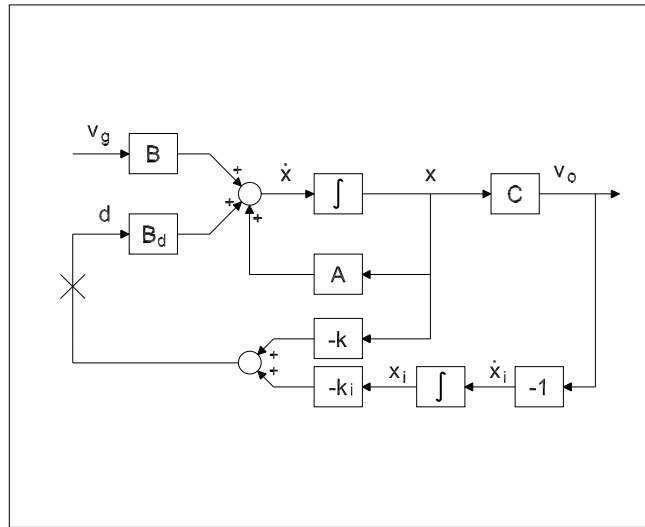


Figure 20.2: System controlled with a FSFBI regulator.

The state quadruple for the system augmented with the new state and controlled with the new control law may be derived from the block diagram.

$$\begin{aligned}\dot{x} &= (A - B_d k)x - B_d k_i x_i + B v_g \\ v_o &= Cx\end{aligned}\quad (20.2)$$

along with the augmented closed-loop state space matrices

$$\begin{aligned}\bar{A} &= \begin{bmatrix} A - B_d k & -B_d k_i \\ -C & 0 \end{bmatrix} \\ \bar{B} &= [B \ 0]' \\ \bar{C} &= [C \ 0] \\ \bar{D} &= [D]\end{aligned}\quad (20.3)$$

The controlled system then becomes:

$$\begin{aligned}\dot{x} &= \bar{A}x + \bar{B}v_g \\ v_o &= \bar{C}x\end{aligned}\quad (20.4)$$

A frequency-sweep search technique similar to that used for the FSFB controller was used to determine the pole placement for the system augmented with an integrator, and the results are shown in Figure 20.3. The frequency multiplier was swept and the first frequency multiplier value with a maximum absolute error determined to occur at  $\omega = 7536$ .

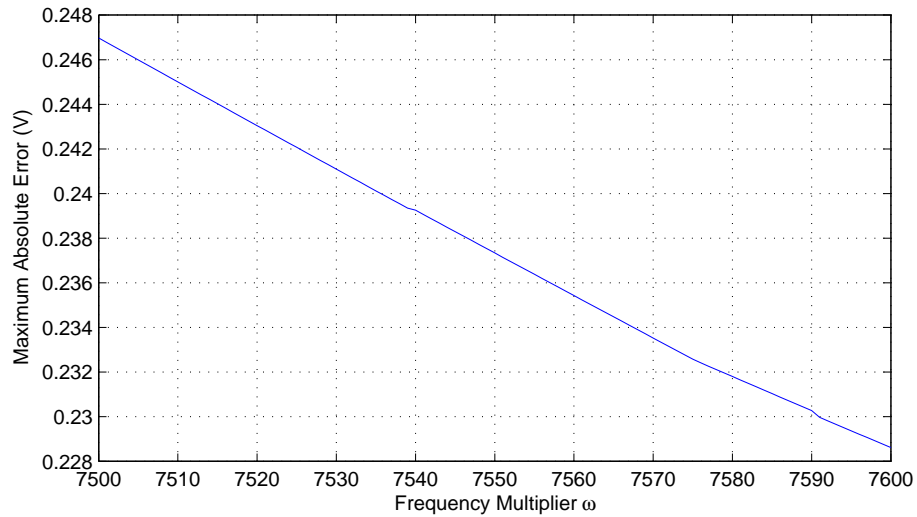


Figure 20.3: ITAE pole frequency multiplier determination.

The step disturbance transient is shown in Figure 20.4.

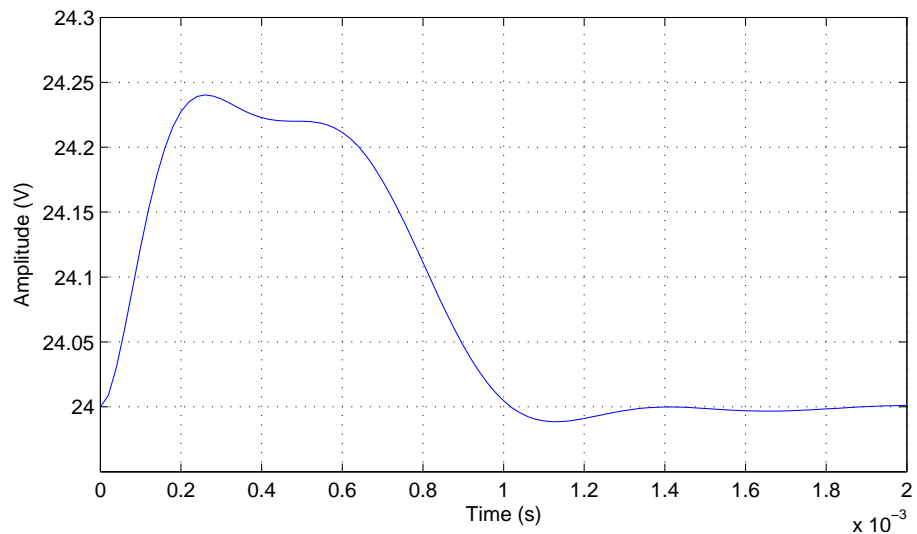


Figure 20.4: Unit step disturbance response of system with FSFBI controller.

The loop gain of the controlled system may be seen in Figure 20.5, where the loop is broken at the large X shown on the control input  $d$  in Figure 20.2. MATLAB calculations give the gain margin as  $\infty$  and a phase margin of  $77.5^\circ$ .

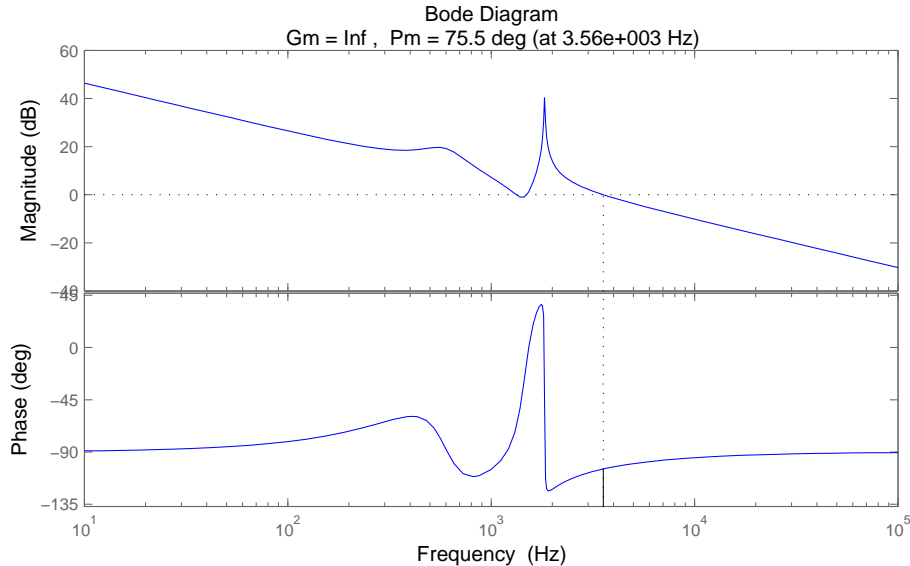


Figure 20.5: Loop gain of Ćuk converter controlled by FSFBI.

Figure 20.6 shows the change in duty cycle effected by the FSFBI controller to control the Ćuk converter. Note that the final value of the control effort is  $-0.018$ . This is the approximate change in duty cycle that is necessary to completely reject the unit step disturbance in  $v_g$  as shown in Figure 18.7.

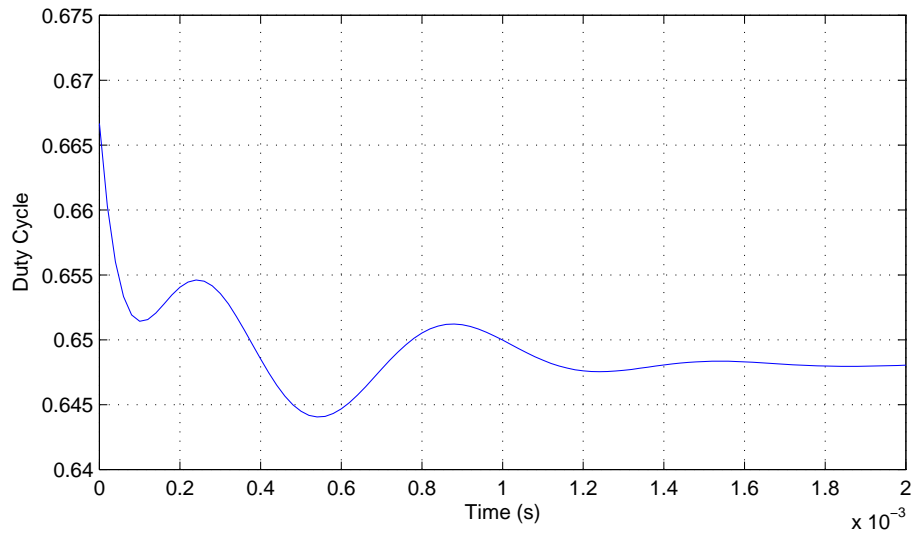


Figure 20.6: Control effort of FSFBI compensator.

From this point forward in this paper, all of the example compensator designs will include an integrator term to completely reject the effects of the step disturbance in  $v_g$ , and though sometimes not explicitly stated in the compensator name, terms related to the integrator will appear in the state equations as well as the block diagrams relating to control of the Ćuk converter.

## 20.3 MATLAB Code

```
1 % Full State Feedback Compensator with Integration
2
3 % Define the plant using an external file
4 cuk_parameters;
5
6 % Define ITAE normalized pole positions
7 ITAE_5 = [-0.8955, -0.3764+1.2920i, -0.3764-1.2920i,...
8 -0.5758+0.5339i, -0.5758-0.5339i];
9
10 % Sweep scaling multiplier and find first multiplier that gives ...
    max error
11 % ≤ 0.24 V
12 w0 = [7500:1:7600];
13 w0_length = length(w0);
14 maxerr = zeros(size(w0));
15
16 wb = waitbar(0, 'Sweeping ITAE Filter Scaling Multiplier...');
17
18 for index = 1:length(w0);
19     P = w0(index) * ITAE_5;
20     A_prime = [A zeros(size(B)); C zeros(size(D))];
21     Bd_prime = [Bd; 0];
22     K = place(A_prime, Bd_prime, P);
23     k = K(1:4);
24     ki = K(end);
25     Abar = [A-Bd*k, -Bd*ki; C, 0];
26     Bbar = [B; 0];
27     Cbar = [C 0];
28     SYS1 = ss(Abar, Bbar, Cbar, D);
29     [y t] = step(SYS1, 0.005);
30     maxerr(index) = max(abs(y));
31     waitbar(index/w0_length, wb)
32 end
33
34 close(wb);
35
36 % Plot the maximum absolute error as a function of the frequency ...
    multiplier
37 figure;
38 h = plot(w0, maxerr);
39 xlabel('Frequency Multiplier \omega');
40 ylabel('Maximum Absolute Error (V)');
41 grid on
42
43 % Determine vector of scaling values that results in max error ≤ ...
    0.24 V
44 z = find(maxerr ≤ 0.24);
45 % Select first element in that vector to use as scaling multiplier
46 w = w0(z(1,1));
47 sprintf('ITAE pole scaling multiplier selected, FSFBI: %4.0f', w)
48
49 % Scale the ITAE filter pole positions
50 P = w * ITAE_5;
```

```

51 A_prime = [A zeros(size(B));-C zeros(size(D))];
52 Bd_prime = [Bd; 0];
53 K = place(A_prime,Bd_prime,P);
54 k = K(1:4);
55 ki = K(end);
56
57 Abar = [A-Bd*k, -Bd*ki; -C, 0];
58 Bbar = [B; 0];
59 Cbar = [C 0];
60 SYS_FSFBI = ss(Abar,Bbar,Cbar,D);
61
62 % Plot closed-loop system response to a unit step disturbance input
63 [y t x] = step(SYS_FSFBI,0.002);
64 figure;
65 plot(t, Vo+y);
66 xlabel('Time (s)');
67 ylabel('Amplitude (V)');
68 grid on
69
70 % System Loop Gain Plot
71 figure;
72 margin(A_prime, Bd_prime, K, 0);
73 h = gcr;
74 h.AxesGrid.TitleStyle.FontSize = 10;
75 h.AxesGrid.XLabelStyle.FontSize = 10;
76 h.AxesGrid.YLabelStyle.FontSize = 10;
77
78 % Control Effort Plot
79 figure;
80 plot(t, -K*x'+d);
81 xlabel('Time (s)');
82 ylabel('Duty Cycle');
83 grid on

```

## Chapter 21

# State Estimation

Since an  $n$ -th order system requires  $n$  states be fed back to the gain matrix  $K$  to allow pole placement anywhere in the complex plane, this requires at least  $n$  measurements of the state variables. This can be prohibitively expensive or complex. In some cases, the internal system states may not even be measurable. In general, only the input and output of a system are available to the control system designer. However, if the system is fully observable, a state estimator (also known as an observer) may be used to provide estimated state values for use in feedback control. The use of an observer requires that the state estimates converge to the actual state values (if starting from different initial states) more rapidly than the system itself responds. The control law used is then:

$$u = -K\hat{x} \quad (21.1)$$

where  $\hat{x}$  indicates that the states fed back into the system are estimates.

In order to quickly force the state estimate to converge to the actual values of the state from arbitrary initial conditions, a correction term must be applied to the estimator dynamics such that the error dynamics approach zero rapidly.

### 21.1 Full-Order State Estimators

A state estimator may be constructed from the same system state space model used for control law gain determination as long as the system is fully observable. The output of the estimator  $C\hat{x}$  can be compared to the output of the system, and any difference between them may be multiplied by a gain vector and fed back to the state estimator dynamics. Therefore:

$$\begin{aligned} e &= y - C\hat{x} \\ &= C(x - \hat{x}) \end{aligned} \quad (21.2)$$

Multiplying this error by a gain vector  $L$ , the desired state error correction term is formed, which can then be added to the dynamics of the estimator to form:

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + Bu - LC(x - \hat{x}) \\ &= (A - LC)\hat{x} + Bu + LCx\end{aligned}\quad (21.3)$$

When  $L$  is chosen such that the eigenvalues of  $A - LC$  lie in the left half of the complex plane, the estimator error  $e \rightarrow 0$  as  $t \rightarrow \infty$ . Since the state estimate must converge to the controlled state faster than the state itself can change, the eigenvalues of  $A - LC$  should be placed farther to the left than the eigenvalues of  $A - BK$ . A good rule of thumb is to make the estimator dynamics at least twice as fast as the controlled system dynamics.

To form an output feedback compensator based on an estimator, the separation principle of controller design holds, which states that the controller gain  $K$  and the observer gain  $L$  can be found independently. The proof of this is in many other references (e.g., [1]), so it shall not be repeated here, but application shall be made of the principle in the design examples to follow.

When paired with the linear state feedback control law, the estimator-based compensator is formed. For the case of state feedback without an integral state added, the compensator is given by:

$$\begin{aligned}\dot{\hat{x}} &= (A - BK - LC)\hat{x} + Ly \\ u &= -K\hat{x}\end{aligned}\quad (21.4)$$

Where the state of the system has been augmented by an integrator state, the compensator is given by:

$$\begin{aligned}\begin{bmatrix} \dot{\hat{x}} \\ \dot{x}_i \end{bmatrix} &= \begin{bmatrix} A - Bk - LC & -Bk_i \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ x_i \end{bmatrix} + \begin{bmatrix} L \\ -1 \end{bmatrix} y \\ u &= \begin{bmatrix} -k & -k_i \end{bmatrix} \begin{bmatrix} \hat{x} \\ x_i \end{bmatrix}\end{aligned}\quad (21.5)$$

## 21.2 Full-Order Estimator-Based Compensator

The block diagram for the Ćuk converter controlled with a full-order state estimator with linear control law and integral action is shown in Figure 21.1, and was used to derive the following closed-loop state equations:

$$\begin{aligned}\dot{x} &= Ax - B_d k \hat{x} - B_d k_i x_i + B v_g \\ \dot{x}_i &= -Cx \\ \dot{\hat{x}} &= LCx - B_d k_i x_i + (A - B_d k - LC)\hat{x} \\ v_o &= Cx\end{aligned}\quad (21.6)$$



and the associated state space matrices:

$$\begin{aligned} \bar{A} &= \begin{bmatrix} A & -B_d k_i & -B_d k \\ -C & 0 & 0 \\ LC & -B_d k_i & A - B_d k - LC \end{bmatrix} \\ \bar{B} &= [B \ 0 \ 0]' \\ \bar{C} &= [C \ 0 \ 0] \\ \bar{D} &= [D] \end{aligned} \quad (21.7)$$

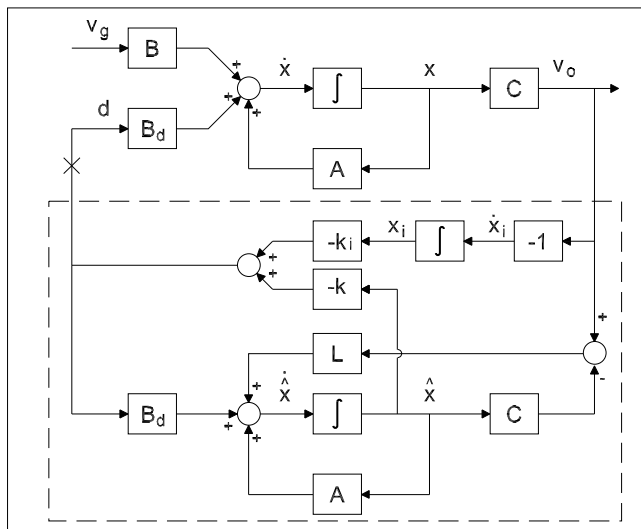


Figure 21.1: An estimated state feedback compensator with integral action.

The step disturbance transient is shown in Figure 21.2. There is more oscillatory behavior in the initial part of the transient response compared to the response under FSFBI control. This is likely caused by the initial estimation of states and their convergence to the actual state values. Comparison of the settling times shows that they are approximately the same, and the only transient differences occur early in the transient.

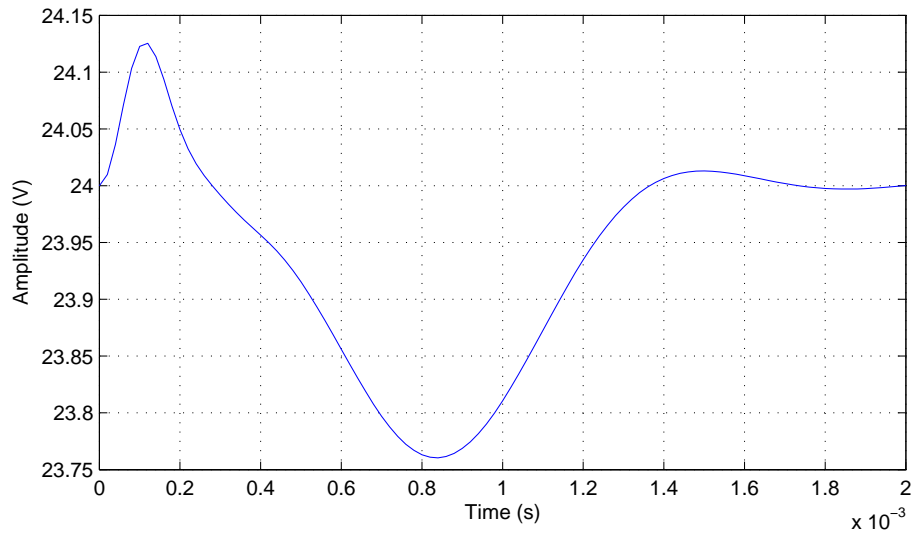


Figure 21.2: Unit step disturbance response of Ćuk converter with ESFBI compensator.

The loop gain of the controlled system may be seen in Figure 21.3, where the loop is broken at the large X shown on the control input  $d$  in Figure 21.1. MATLAB calculations give the gain margin as 5.97 dB and a phase margin of  $26.3^\circ$ . These are significantly lower than the stability margins of previous controllers designed, and these clearly do not meet the design specifications.

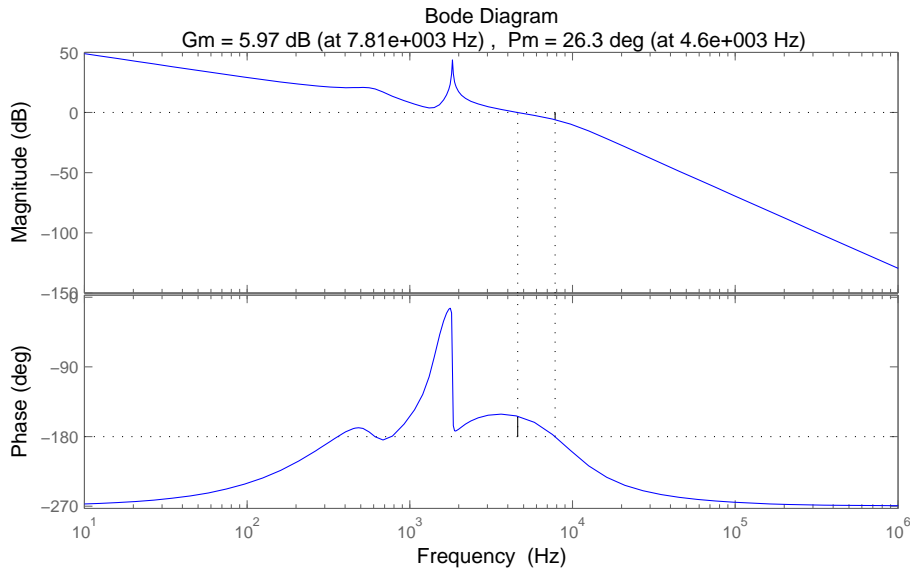


Figure 21.3: Loop gain of Ćuk converter controlled by ESFBI compensator.

Figure 21.4 shows the control effort of the ESFBI controller.

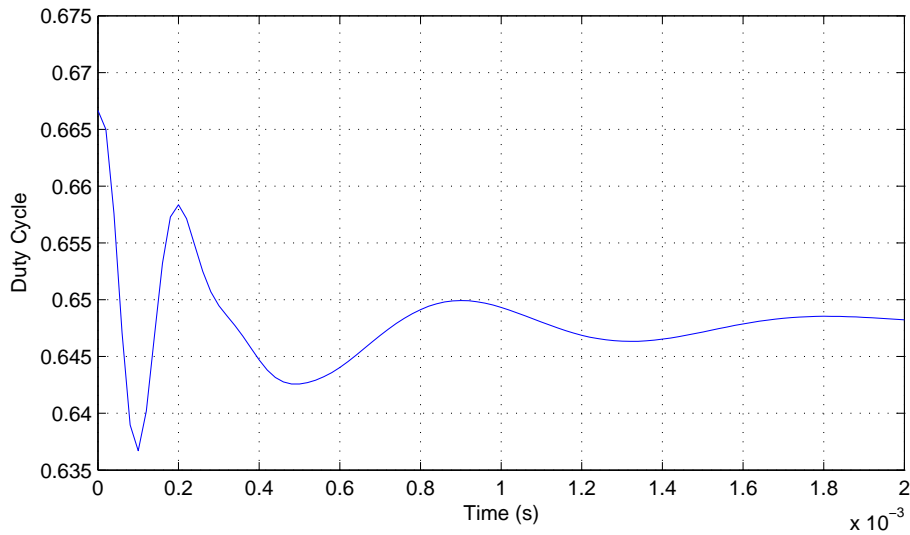


Figure 21.4: Control effort of ESFBI compensator.

Since the Ćuk converter has four states, the observer itself will be a fourth-order system. Along with the integral state, the compensator becomes a fifth-order system. When evaluating desired observer pole locations, keep in mind

that the poles of the observer must be placed to the left of the poles of the compensated plant by a large margin to ensure that the state estimator dynamics are faster than those of the plant. Similarly to the use of fifth-order ITAE pole locations for the plant under state feedback, fourth-order ITAE pole locations may be used to place the poles of the observer. However, examination of the frequency-normalized fifth-order and fourth-order ITAE pole locations from Table 19.1 shows that the fifth-order poles have a minimum pole frequency greater than the maximum pole frequency of the fourth-order poles. Thus, if the designer intends to place the all of the fourth-order ITAE observer poles to the left of the fifth-order ITAE pole locations selected using  $\omega$  by at least a factor of  $n$ , the designer must scale the fourth-order pole locations such that the lowest pole frequency of the fourth-order system must be first moved out to the largest pole frequency of the fifth-order system. This can be done by multiplying the fourth-order frequency-normalized poles by the ratio of the maximum pole frequency of the fifth-order poles to the minimum pole frequency of the fourth-order poles. The resulting fourth-order pole locations can then each be multiplied by  $n$  to move them radially by a factor of  $n$  farther into the left half of the complex plane than the fifth-order normalized pole locations, then they can be multiplied by the value of  $\omega$  used to select the desired fifth-order pole locations. Thus, the desired fourth-order pole locations have been determined.

### 21.3 Reduced-Order State Estimators

If some of the states appear directly in the output as a function of the measurement equation (i.e., are not a linear combination of other states), those states do not need to be estimated. Hence, a reduced-order estimator may be constructed that only estimates the unmeasured states.

If  $C$  is a full rank matrix, a nonsingular linear transformation matrix can be formed by choosing a matrix  $T$  of dimension  $(n - r) \times n$  and forming:

$$P = \begin{bmatrix} C \\ T \end{bmatrix} \quad (21.8)$$

The matrix  $T$  may be any arbitrary matrix that produces a nonsingular  $P$  matrix, as  $P$  must be invertible.

Applying the standard linear transformation  $x = Px$  with Equation (21.8) puts the system into an equivalent representation where  $C$  is of the form:

$$[I_r \quad 0] \quad (21.9)$$

( $I$  is a square identity matrix with dimension  $r \times r$ ,  $r < n$ ). It can be assumed that this representation exists without loss of generality since it is the result of a linear transformation.

The system and feedback gain matrix can now be partitioned into measured

and unmeasured portions:

$$\begin{aligned} \begin{bmatrix} \dot{x}_m \\ \dot{x}_u \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_m \\ x_u \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u & (21.10) \\ y &= x_m \\ K &= [k_m \ k_u] \end{aligned}$$

Since  $x_m$  corresponds to the states that appear in the output of the measurement equation,  $x_u$  represents the remaining unmeasured states. As  $x_m$  is present in the output, these states do not require estimation. This means that a reduced-order state estimator can be constructed that allows the estimation of  $x_u$  in such a manner that all states (either measured or estimated) are available for feedback via a linear control law:

$$u = -k_m x_m - k_u \hat{x}_u \quad (21.11)$$

In order to ensure that the reduced-order observer dynamics converge to the true state values of  $x$ , the error dynamics must converge to zero. A full-order observer uses a correction term to perform this, which is a gain matrix that multiplies the error between plant output and observer output. Unfortunately, since only the  $x_m$  states appear in the plant output, using the plant output contributes no information about the unmeasured states to the estimator and therefore has no dynamic effect on the estimate  $\hat{x}_u$ . However, a variable change can be performed:

$$\hat{x}_u = Ly + z \quad (21.12)$$

where  $z$  is the output of a system of order  $r < n$ :

$$\dot{z} = Ez + Fy + Gu \quad (21.13)$$

with  $E$ ,  $F$ , and  $G$  and the observer gain  $L$  yet to be determined.

Since observer design is concerned with elimination of the error between the actual state and its estimate, the estimation error can be defined as  $e_u = x_u - \hat{x}_u$  and the error dynamics will converge to zero if  $\hat{x}_u \rightarrow x_u$ .

Using the partitioned format of  $x_u$  from Equation 21.10 along with  $\hat{x}_u$  determined from the block diagram in Figure 21.5, it can be shown that:

$$\begin{aligned} \dot{\hat{x}}_u &= L(A_{11}x_m + A_{12}x_u + B_1u) + E(\hat{x}_u - Lx_m) + Fx_m + Gu & (21.14) \\ \dot{x}_u &= A_{21}x_m + A_{22}x_u + B_2u \end{aligned}$$

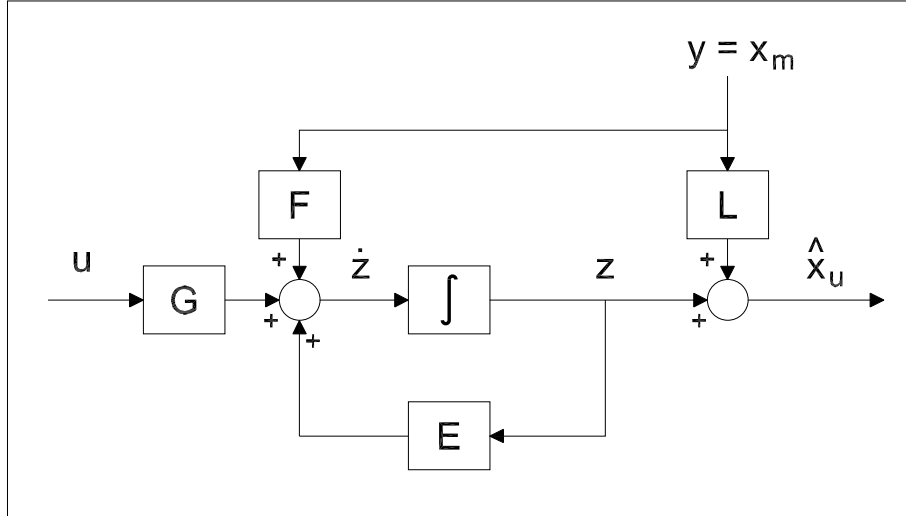


Figure 21.5: Reduced-order estimator construction.

Adding zero to the right side of  $\dot{\hat{x}}_u$  in the form of  $Ex_u - Ex_u$ :

$$\begin{aligned} \dot{\hat{x}}_u = & (LA_{11} - EL + F)x_m + (LA_{12} + E)x_u \dots & (21.15) \\ & +(LB_1 + G)u + E(x_u - \hat{x}_u) \end{aligned}$$

This gives the following equation for the estimator error dynamics when substituted into the time derivative of the unmeasured error  $\dot{e}_u = \dot{x}_u - \dot{\hat{x}}_u$ :

$$\begin{aligned} \dot{e}_u = & (A_{21} - LA_{11} + EL - F)x_m + (A_{22} - LA_{12} - E)x_u \dots & (21.16) \\ & +(B_2 - LB_1 - G)u + Ee_u \end{aligned}$$

In order for the estimator error dynamics to be independent of the state  $x$ , plant output  $y$ , and the input  $u$ , the first three terms on the right side of Equation 21.16 must be zero. This is accomplished by selecting:

$$\begin{aligned} E &= A_{22} - LA_{12} & (21.17) \\ F &= A_{21} - LA_{11} + EL \\ G &= B_2 - LB_1 \end{aligned}$$

This leaves  $\dot{e}_u = Ee_u$ , which converges to zero when  $E$  is an asymptotically stable matrix. Thus, the error dynamics die out, leaving an estimate that equals the state. This means that  $L$  must be selected such that  $A_{22} - LA_{12} \rightarrow 0$  as  $t \rightarrow \infty$ , i.e., the poles of  $E$  must lie in the left half of the complex plant. As with the full-order observer, the reduced-order observer poles should be placed such that the estimator dynamics are much more rapid than the controlled plant dynamics.

When paired with the properly-partitioned linear state feedback control law, the reduced-order estimator-based compensator is formed. For the case of state feedback without an integral state added, the reduced-order compensator is given by:

$$\begin{aligned}\dot{z} &= (E - Gk_u)z + (F - Gk_uL - Gk_m)y \\ u &= -k_uz + (-k_uL - k_m)y\end{aligned}\quad (21.18)$$

Where the state of the system has been augmented by an integrator state, the reduced-order compensator is given by:

$$\begin{aligned}\begin{bmatrix} \dot{z} \\ \dot{x}_i \end{bmatrix} &= \begin{bmatrix} E - Gk_u & -Gk_i \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ x_i \end{bmatrix} + \begin{bmatrix} F - Gk_uL - Gk_m \\ -1 \end{bmatrix} y \\ u &= \begin{bmatrix} -k_u & -k_i \end{bmatrix} \begin{bmatrix} z \\ x_i \end{bmatrix} + (-k_uL - k_m)y\end{aligned}\quad (21.19)$$

## 21.4 Reduced-Order Estimator-Based Compensator

For the Ćuk converter model, matrix  $C$  is of the form  $C = [ I \ 0 ]$  therefore  $v_2$  is a measured state, and the unmeasured states are  $v_1$ ,  $i_2$ , and  $i_1$ . This means that a reduced order observer may be designed that estimates only the three unmeasured states. The block model for a system controlled by a compensator made of a reduced-order estimator and a linear state feedback law is shown in Figure 21.6, and was used to derive the following state equations:

$$\begin{aligned}\dot{x} &= (A - B_dk_uLC - B_dk_mC)x - B_dk_ix_i - B_dk_uz + Bv_g \\ \dot{x}_i &= -Cx \\ \dot{z} &= (FC - Gk_mC - Gk_uLC)x - Gk_ix_i + (D - Gk_u)z \\ v_o &= Cx\end{aligned}\quad (21.20)$$

from which were determined the matrices:

$$\begin{aligned}\bar{A} &= \begin{bmatrix} A - B_dk_mC - B_dk_uLC & -B_dk_i & -B_dk_u \\ & -C & 0 \\ FC - Gk_mC - Gk_uLC & -Gk_i & E - Gk_u \end{bmatrix} \\ \bar{B} &= [ B \ 0 \ 0 ]' \\ \bar{C} &= [ C \ 0 \ 0 ] \\ \bar{D} &= [ D ]\end{aligned}\quad (21.21)$$

where

$$\begin{aligned}E &= A_{22} - LA_{12} \\ F &= EL + A_{21} - LA_{11} \\ G &= B_{d2} - LB_{d1}\end{aligned}\quad (21.22)$$

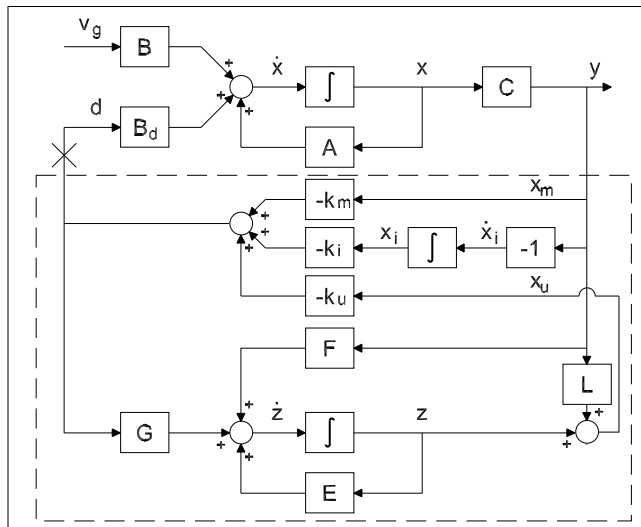


Figure 21.6: A state feedback regulator with integral action based on a reduced-order estimator.

To determine estimator pole placement, the same type of iterative algorithm used for the full-order observer was implemented. The reduced-order observer poles were placed to have faster dynamics than the poles of the controlled system.

The step disturbance transient is shown in Figure 21.7.



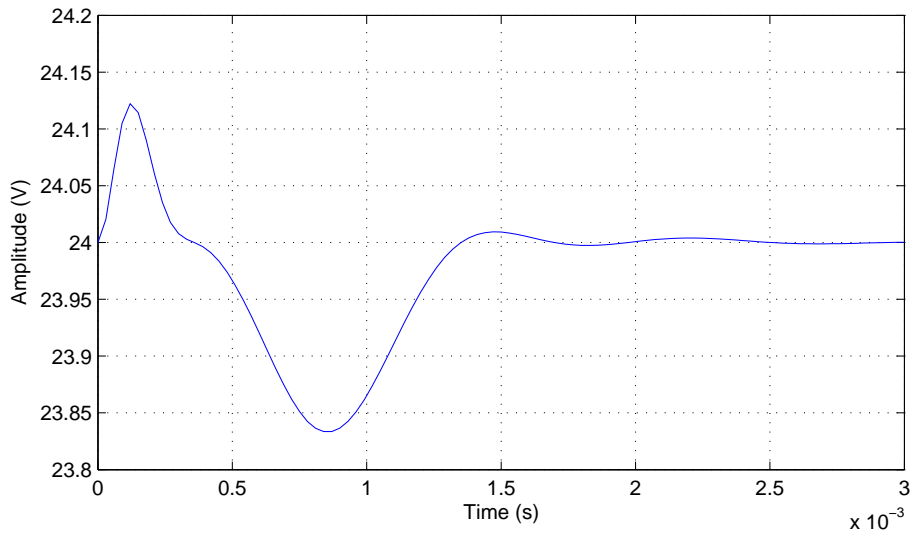


Figure 21.7: Unit step disturbance response of Ćuk converter with ROESFBI compensator.

The loop gain of the controlled system may be seen in Figure 21.8, where the loop is broken at the large X shown on the control input  $d$  in Figure 21.6. MATLAB calculations give the gain margin as  $-24.9$  dB and a phase margin of  $36.5^\circ$ . These clearly do not meet the design specifications, and uncertainty in the system model may make the controlled system unstable.

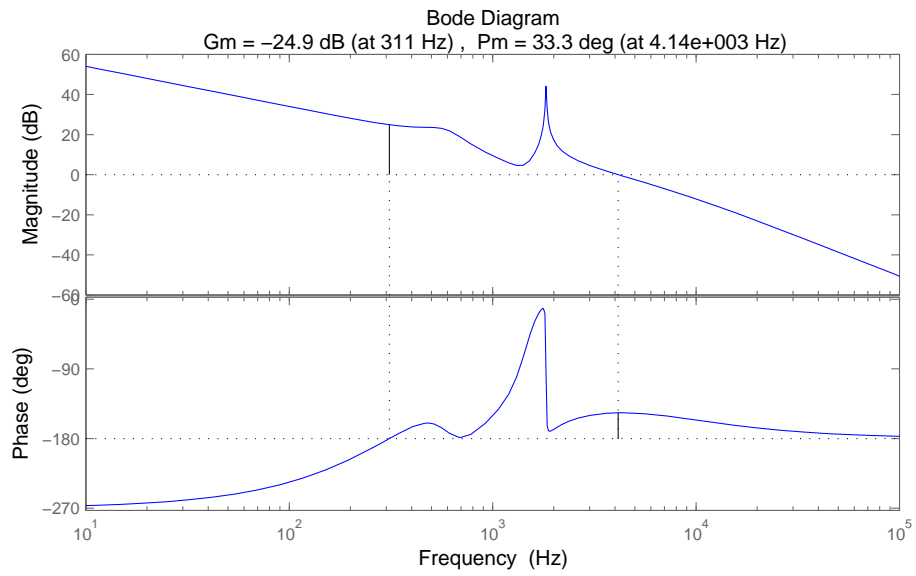


Figure 21.8: Loop gain of Ćuk converter controlled by ROESFBI compensator.

## 21.5 MATLAB Code

```
1 % Estimated State Feedback Compensator with Integration
2
3 % Define the plant using an external file
4 cuk_parameters;
5
6 % Define ITAE normalized pole positions
7 ITAE_5 = [-0.8955, -0.3764+1.2920i, -0.3764-1.2920i,...
8 -0.5758+0.5339i, -0.5758-0.5339i];
9 ITAE_4 = [-0.4240+1.2360i, -0.4240-1.2360i, ...
10 -0.6260+0.4141i, -0.6260-0.4141i];
11
12 % Find the ratio of the maximum natural frequency of a normalized
13 % fifth-order ITAE filter to the minimum natural frequency of a
14 % normalized fourth-order ITAE filter.
15
16 ra54 = max(abs(ITAE_5))/min(abs(ITAE_4));
17
18 % Define the desired ratio of the minimum natural frequency of the
19 % estimator poles to the maximum natural frequency of the ...
    controller poles
20 rra = 3;
21
22 % Sweep scaling multiplier and find first multiplier that gives ...
    max error
23 %  $\leq 0.24$  V
24 w0 = [5400:1:5600];
25 w0_length = length(w0);
26 err_ITAE = zeros(size(w0));
27
28 wb = waitbar(0, 'Sweeping ITAE Filter Scaling Multiplier...');
29 for index = 1:length(w0);
30     % Sweep frequency to scale the ITAE filter pole positions for
31     % the controller
32     Pk = w0(index) * ITAE_5;
33
34     % Set the minimum natural frequency of the ITAE filter pole ...
        positions
35     % for the estimator to a multiple of the maximum natural ...
        frequency of
36     % the controller poles to make the estimator converge faster ...
        than the
37     % controller responds
38
39     Pl = rra * ra54 * w0(index) * ITAE_4;
40     A_prime = [A zeros(size(B)); -C zeros(size(D))];
41     Bd_prime = [Bd; 0];
42     C_prime = [C 0];
43     K = place(A_prime, Bd_prime, Pk);
44     k = K(1:4);
45     ki = K(end);
46     L = place(A', C', Pl)';
47     Abar = [A, -Bd*ki, -Bd*k;
48             -C, 0, zeros(1, length(A))];
```

```

49         L*C, -Bd*ki, A-Bd*k-L*C];
50     Bbar = [B; 0; zeros(4,1)];
51     Cbar = [C 0 zeros(size(C))];
52     SYS_ESFBI = ss(Abar,Bbar,Cbar,D);
53     [y t] = step(SYS_ESFBI,0.003);
54     maxerr(index) = max(abs(y));
55     waitbar(index/w0_length, wb)
56 end
57 close(wb);
58
59 % Plot the maximum absolute error as a function of the frequency ...
    multiplier
60 figure;
61 plot(w0,maxerr);
62 axis tight
63 xlabel('Scalar Multiplier \omega');
64 ylabel('Maximum Error (V)');
65 grid on
66
67 % Determine vector of scaling values that results in max error ≤ ...
    0.24 V
68 z = find(maxerr≤0.24);
69 % Select first element in that vector to use as scaling multiplier
70 w = w0(z(1,1));
71 sprintf('ITAE pole scaling multiplier selected, ESFBI: %4.0f',w)
72
73 % Use frequency of minimum ITAE to scale the ITAE filter pole ...
    positions for
74 % the controller
75 Pk = w * ITAE_5;
76
77 % Set the minimum natural frequency of the ITAE filter pole ...
    positions for
78 % the estimator to a multiplier of the maximum natural frequency ...
    of the
79 % controller poles to make the estimator converge faster than the
80 % controller responds
81
82 P1 = rra * ra54 * w * ITAE_4;
83 A_prime = [A zeros(size(B));-C zeros(size(D))];
84 Bd_prime = [Bd; 0];
85 C_prime = [C 0];
86 K = place(A_prime,Bd_prime,Pk);
87 k = K(1:4);
88 ki = K(end);
89 L = place(A', C', P1)';
90 Abar = [A,    -Bd*ki, -Bd*k;
91         -C,   0, zeros(1,length(A));
92         L*C, -Bd*ki, A-Bd*k-L*C];
93 Bbar = [B; 0; zeros(4,1)];
94 Cbar = [C 0 zeros(size(C))];
95 SYS_ESFBI = ss(Abar,Bbar,Cbar,D);
96
97 % Plot controller poles and observer poles
98 figure;
99 plot(Pk,'bx')
100 hold on

```

```

101 plot(P1, 'r+')
102 hold off
103 legend('Controller poles', 'Observer poles')
104 xlabel('Real')
105 ylabel('Imaginary')
106 grid on
107
108 % Plot closed-loop system response to a unit step disturbance input
109 [y t x] = step(SYS_ESFBI,0.002);
110 figure;
111 h = plot(t, Vo+y);
112 xlabel('Time (s)');
113 ylabel('Amplitude (V)');
114 grid on
115
116 figure;
117 plot(t, x);
118 xlabel('Time (s)');
119 ylabel('States');
120 grid on
121 legend('v_2', 'v_1', 'i_2', 'i_1', 'x_i', 'v_{2e}', 'v_{1e}', ...
122 'i_{2e}', 'i_{1e}', 'Location', 'SouthEast');
123
124 % System Loop Gain Plot
125 A1 = [A, zeros(4,1), zeros(4,4);
126       -C, 0, zeros(1,4);
127       L*C, -Bd*ki, A-Bd*k-L*C];
128 Bd1 = [Bd;0;zeros(4,1)];
129 C1 = [zeros(1,4), ki, k];
130 D1 = 0;
131 sys1 = ss(A1,Bd1,C1,D1);
132 figure(5);
133 margin(sys1)
134 h = gcr;
135 h.AxesGrid.Xunits = 'Hz';
136 h.AxesGrid.TitleStyle.FontSize = 10;
137 h.AxesGrid.XLabelStyle.FontSize = 10;
138 h.AxesGrid.YLabelStyle.FontSize = 10;
139
140 % Control Effort Plot
141 figure;
142 h = plot(t, -[zeros(1,4), ki, k]*x'+d);
143 xlabel('Time (s)');
144 ylabel('Duty Cycle');
145 grid on

```

```

1 % Reduced-Order Estimated State Feedback Compensator with ...
  Integration
2
3 % Define the system using an external file
4 cuk_parameters;
5
6 % Partition the matrices for reduced-order estimator design
7 A11 = A(1,1);
8 A12 = A(1,2:4);
9 A21 = A(2:4,1);

```

```

10 A22 = A(2:4,2:4);
11
12 B1 = B(1);
13 B2 = B(2:4);
14
15 Bd1 = Bd(1);
16 Bd2 = Bd(2:4);
17
18 % Define ITAE normalized pole positions
19 ITAE_5 = [-0.8955, -0.3764+1.2920i, -0.3764-1.2920i,...
20 -0.5758+0.5339i, -0.5758-0.5339i];
21 ITAE_3 = [-0.7081, -0.5210+1.068i, -0.5210-1.068i];
22
23 % Find the ratio of the maximum natural frequency of a normalized
24 % fifth-order ITAE filter to the minimum natural frequency of a
25 % normalized third-order ITAE filter.
26
27 ra53 = max(abs(ITAE_5))/min(abs(ITAE_3));
28
29 % Define the desired ratio of the minimum natural frequency of the
30 % estimator poles to the maximum natural frequency of the ...
    controller poles
31 rra = 2;
32
33 % Sweep scaling multiplier and find first multiplier that gives ...
    max error
34 % ≤ 0.24 V
35 w0 = [5700:1:5900];
36 w0_length = length(w0);
37 maxerr = zeros(size(w0));
38
39 wb = waitbar(0, 'Sweeping ITAE Filter Scaling Multiplier...');
40
41 for index = 1:length(w0);
42     % Scale the controller pole positions using a search
43     Pk = w0(index) * ITAE_5;
44
45     % Set the minimum natural frequency of the ITAE filter pole ...
        positions
46     % for the estimator to a multiple of the maximum natural ...
        frequency of
47     % the controller poles to make the estimator converge faster ...
        than the
48     % controller responds
49
50     Pl = rra * ra53 * w0(index) * ITAE_3;
51     A_prime = [A zeros(size(B)); -C zeros(size(D))];
52     Bd_prime = [Bd; 0];
53     C_prime = [C 0];
54     K = place(A_prime, Bd_prime, Pk);
55     km = K(1);
56     ku = K(2:4);
57     ki = K(end);
58     L = place(A22', A12', Pl)';
59
60     E = A22 - L*A12;
61     F = E*L + A21 - L*A11;

```

```

62     G = Bd2 - L*Bd1;
63
64     Abar = [A-Bd*km*C-Bd*ku*L*C, -Bd*ki, -Bd*ku;
65            -C, zeros(1,4) ;
66            F*C-G*km*C-G*ku*L*C, -G*ki, E-G*ku];
67     Bbar = [B; 0; zeros(3,1)];
68     Cbar = [C 0 zeros(1,3)];
69     SYS1 = ss(Abar,Bbar,Cbar,D);
70     [y t] = step(SYS1,0.003);
71     maxerr(index) = max(abs(y));
72     waitbar(index/w0_length, wb)
73 end
74 close(wb);
75
76 % Plot the maximum absolute error as a function of the frequency ...
77     multiplier
78 figure;
79 plot(w0,maxerr);
80 axis tight
81 xlabel('Scalar Multiplier \omega');
82 ylabel('Maximum Error (V)');
83 grid on
84
85 % Determine vector of scaling values that results in max error ≤ ...
86     0.24 V
87 z = find(maxerr≤0.24);
88 % Select first element in that vector to use as scaling multiplier
89 w = w0(z(1,1));
90 sprintf('ITAE pole scaling multiplier selected, ESFBI: %4.0f',w)
91
92 % Determine controller and observer gains
93 Pk = w * ITAE_5;
94
95 % Set the minimum natural frequency of the ITAE filter pole ...
96     positions for
97 % the estimator to a multiplier of the maximum natural frequency ...
98     of the
99 % controller poles to make the estimator converge faster than the
100 % controller responds
101
102 P1 = rra * ra53 * w * ITAE_3;
103 A_prime = [A zeros(size(B));-C zeros(size(D))];
104 Bd_prime = [Bd; 0];
105 C_prime = [C 0];
106 K = place(A_prime,Bd_prime,Pk);
107 km = K(1);
108 ku = K(2:4);
109 ki = K(end);
110 L = place(A22', A12', P1)';
111
112 E = A22 - L*A12;
113 F = E*L + A21 - L*A11;
114 G = Bd2 - L*Bd1;
115
116 Abar = [A-Bd*km*C-Bd*ku*L*C, -Bd*ki, -Bd*ku;
117        -C, zeros(1,4) ;
118        F*C-G*km*C-G*ku*L*C, -G*ki, E-G*ku];

```

```

115 Bbar = [B; 0; zeros(3,1)];
116 Cbar = [C 0 zeros(1,3)];
117 SYS_ROESFBI = ss(Abar,Bbar,Cbar,D);
118
119 % Plot controller and observer pole locations
120 figure;
121 plot(Pk,'bx')
122 hold on
123 plot(Pl,'r+')
124 hold off
125 legend('Controller poles','Observer poles')
126 xlabel('Real')
127 ylabel('Imaginary')
128
129 % Plot closed-loop system response to a unit step input
130 [y t x] = step(SYS_ROESFBI,0.003);
131 figure;
132 plot(t, Vo+y);
133 xlabel('Time (s)');
134 ylabel('Amplitude (V)');
135 grid on
136
137 % System Loop Gain Plot
138 A1 = [A, zeros(4,1), zeros(4,3);
139       -C, 0, zeros(1,3);
140       F*C-G*ku*L*C-G*km*C, -G*ki, E-G*ku];
141 Bd1 = [Bd; 0; zeros(3,1)];
142 C1 = [km*C+ku*L*C, ki, ku];
143 D = 0;
144 sys1 = ss(A1, Bd1, C1, D);
145 figure;
146 margin(sys1);
147 h = gcr;
148 h.AxesGrid.TitleStyle.FontSize = 10;
149 h.AxesGrid.XLabelStyle.FontSize = 10;
150 h.AxesGrid.YLabelStyle.FontSize = 10;
151
152 % Control Effort Plot
153 figure;
154 h = plot(t, -[km, zeros(1,3), ki, ku]*x'+d);
155 xlabel('Time (s)');
156 ylabel('Duty Cycle');
157 grid on

```



# References

- [1] R. T. Stefani, B. Shahian, C. J. Savant, Jr., and G. H. Hostetter, *Design of Feedback Control Systems*, 4th ed. New York, NY: Oxford University Press, 2002.



## Chapter 22

# Linear Quadratic Optimal Control

Linear quadratic optimal control uses penalties on state transients  $x$  and control effort  $u$  to optimize system performance with respect to a figure of merit determined by a cost function. There is typically a classical trade-off designed into the cost function: one cannot have tight control over state transients with small control effort. In other words, small output transients require large controller gains (and therefore control effort). Additionally, quadratic forms are used to ensure that only the magnitude and not the sign of the transient contributes to the cost determined by the penalty function.

### 22.1 Linear Quadratic Regulators

To optimally control state transients and control effort within performance specifications, a compensator is sought that seeks to provide a control effort  $u$  that minimizes a Lagrangian cost function:

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (22.1)$$

subject to the constraint of the state equation:

$$\dot{x} = Ax + Bu \quad (22.2)$$

This is known as the linear quadratic regulator (LQR) problem. The weight matrix  $Q$  is an  $n \times n$  positive semidefinite matrix (for a system with  $n$  states) that penalizes variation of the state from the desired state. The weight matrix  $R$  is an  $m \times m$  positive definite matrix that penalizes control effort. Solutions for the constrained optimal system can be found in [1], [2], and [3]. The well-published time invariant solution to this problem is:

$$K = R^{-1} B^T P \quad (22.3)$$

where  $P$  is the unique, symmetric, positive definite solution to the steady-state algebraic Riccati equation (ARE):

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \quad (22.4)$$

The minimum value of the cost function is based on the initial state  $x_0$ , and is given by:

$$J_{min} = x_0^T P x_0 \quad (22.5)$$

The LQR designed for a SISO system can be shown to possess very desirable stability properties: it always has a gain margin between  $\{-6 \text{ dB}, \infty\}$  and a phase margin of at least  $60^\circ$ . However, it has a high frequency roll-off rate of only 20 dB per decade so the open loop frequency response shows susceptibility to high frequency noise.

Since the weight matrices  $Q$  and  $R$  are both included in the summation term within the cost function, it is really the relative size of the weights within each quadratic form which are important. Simple inspection of the cost function shows that multiplying both weight matrices by the same real constant (e.g.,  $\kappa$ ) will not affect their ratio. The multiplier  $\kappa$  may be factored out of the integral, thus returning the cost function to its original form. Thus, the problem of minimizing  $\kappa J$  becomes the same as minimizing  $J$ . Therefore, holding one weight matrix constant while varying either the individual elements or a scalar multiplier of the other is an acceptable technique for iterative design. It is good for the designer to maintain an understanding of the effects of manipulating individual weights, however. In general, raising the effective penalty a single state or control input by manipulating its individual weight will tighten the control over the variation in that parameter, however it may do so at the expense of larger variation in the other states or inputs.

## 22.2 Ćuk Converter with LQR Compensator

The fifth-order system formed by augmenting the Ćuk converter with an integral state requires that  $Q$  be a five-by-five matrix. To review, the state vector is  $[v_2 \ v_1 \ i_2 \ i_1 \ x_i]^T$ , where  $x_i$  corresponds to the integral of the reference error  $e$ . Since the states have physical significance in this model, it is easy to see that each voltage, current, or error transient may be individually penalized using the diagonal elements of the  $Q$  matrix. As the objective of controlling the Ćuk converter is to regulate the output  $v_o = v_2$  in the face of disturbances to  $v_g$ , penalizing transients that occur on state  $v_2$  is a logical choice, as is penalizing the state  $x_i$  associated with the reference error integral as the integral of that error should be minimized to provide for good regulation. Hence, the  $Q$  matrix chosen for design of the LQR has two positive entries corresponding to the first ( $Q_{11}$ ) and last ( $Q_{55}$ ) entries along the diagonal to ensure it is positive semidefinite. The Ćuk converter has only a single control input, and for initial design  $R$  was set equal to 1 arbitrarily. If the control effort exceeds the limitations put on it

(refer to Section 18.6), the value of  $R$  may need to be increase to penalize the control effort.

After proceeding through an iterative process where  $Q_{11}$  and  $Q_{55}$  were varied with  $R$  fixed, a finalized design was determined. The step disturbance response of the controlled Ćuk converter is shown in Figure 22.1

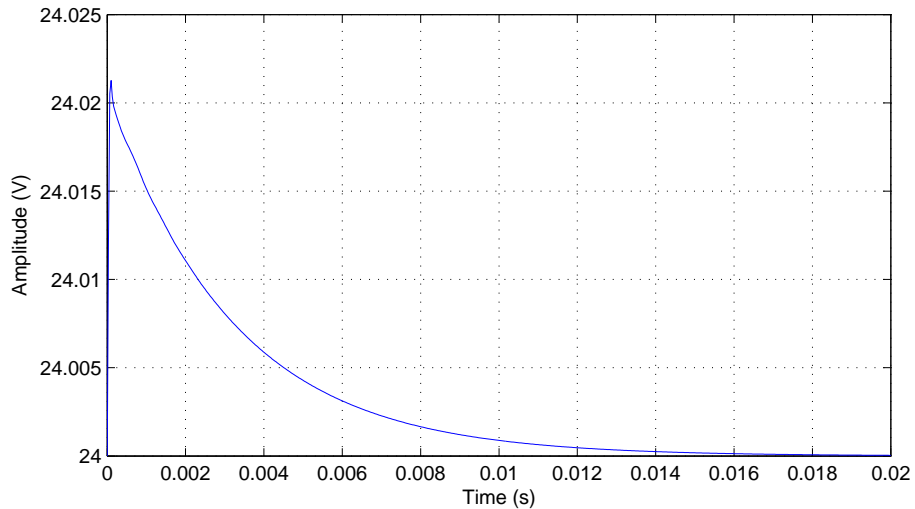


Figure 22.1: Unit step disturbance response of the Ćuk converter controlled by LQR.

The loop gain of the controlled system is shown in Figure 22.2, where the loop is broken at the large X shown on the control input  $d$ .

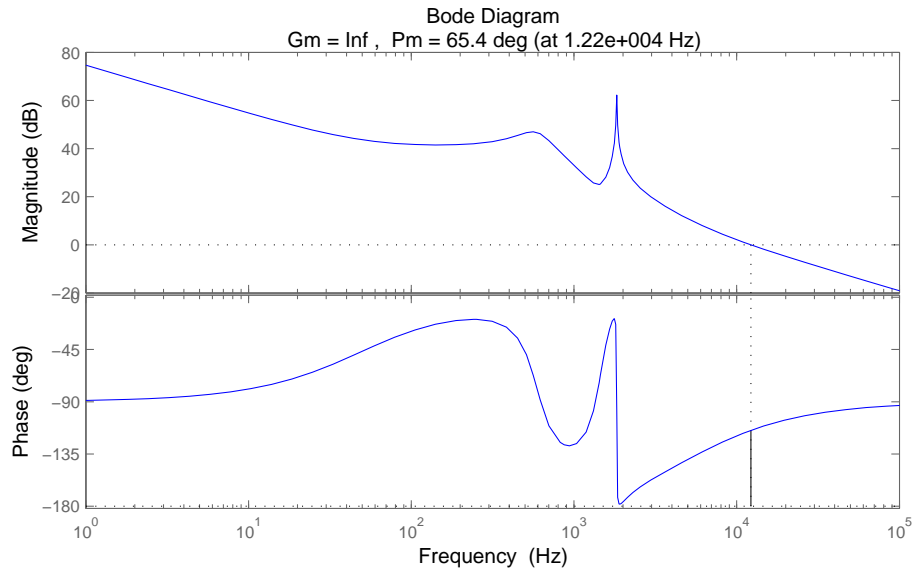


Figure 22.2: Loop gain of Ćuk converter controlled by LQR compensator.

Figure 22.3 shows the control effort produced by the optimal controller.

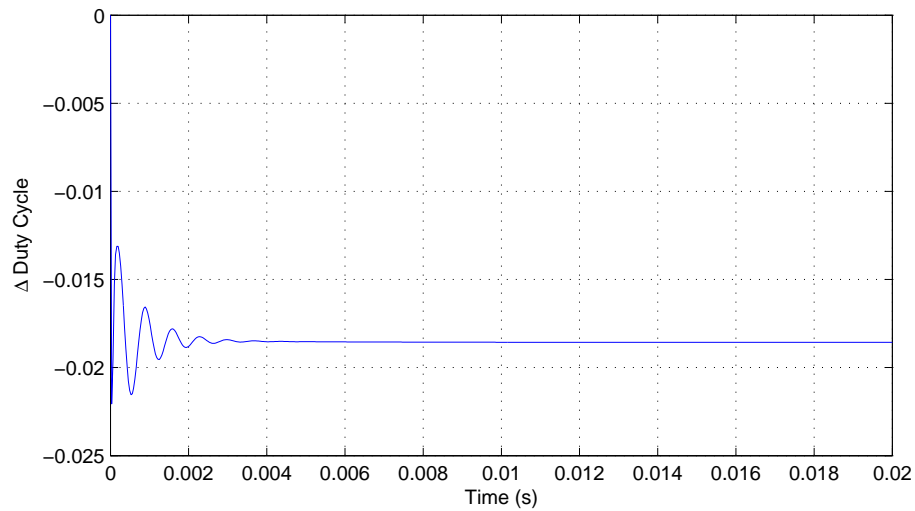


Figure 22.3: LQR control effort.

Note that the time of the unit step disturbance transient with the LQR compensator was significantly longer than the settling times of the previously designed compensators. However, the amplitude deviation was significantly

smaller. This is an acceptable design tradeoff since the performance specifications were still met.

## 22.3 Linear Quadratic Gaussian Regulators

The LQR problem requires that full state feedback be used. This is not always possible, as was discussed in Section 21. The Kalman-Bucy filter is the dual to the LQR problem; it forms an optimal state estimator in the presence of process and measurement noise for a system that is observable. The presence of noise introduces stochastic effects on the state trajectory, therefore, the optimal state estimator must deal with these stochastic effects appropriately by removing them. Since the process of removing noise from a signal is commonly known as filtering, the optimal observer became known as the Kalman filter, or Kalman-Bucy filter, after its creator(s). The Kalman filter is based on essentially the same mathematics as the LQR and, as such, has come to be known as a linear quadratic estimator (LQE). The cost function  $J_0$  that is minimized is the error variance between the state vector and its estimate. LQE calculates the solution to an ARE (which happens to be the error covariance) and uses it to determine  $L$ , the estimator gain vector, along the lines of the same equations given for the LQR gain  $K$ .  $Q_0$ , the covariance matrix of the process noise, and  $R_0$ , the covariance matrix of the sensor noise, have to be selected for optimal determination of the observer gain vector  $L$ .

When combined, the linear quadratic controller and estimator form a structure known as a linear quadratic Gaussian (LQG) compensator. Once the state variables have been estimated, they are fed back through the controller gain  $K$  to close the loop. The separation principle previously mentioned still holds true for LQG design: the LQR and LQE problems can be solved independently via two algebraic Riccati equations.

Adding process disturbances and measurement noise to the state space system description results in:

$$\begin{aligned} \dot{x} &= Ax + Bu + \omega \\ y &= Cx + \nu \end{aligned} \quad (22.6)$$

where  $\omega$  is the disturbance signal (typically modeled by Gaussian white noise of spectral density  $Q_0$ ) and  $\nu$  is additive measurement noise (Gaussian white noise, with spectral density  $R_0$ ). Note that the fact that neither  $\omega$  or  $\nu$  has a coefficient implies a coefficient of  $I$ , meaning each state and output has its own distinct noise [4]. The time invariant solution to the optimal estimator problem is then:

$$L = P_0 C^T R_0^{-1} \quad (22.7)$$

where  $P$  (known as the estimation error variance) is the unique, symmetric, positive definite solution to the steady-state algebraic Riccati equation:

$$AP_0 + P_0 A^T - P_0 C^T R_0^{-1} C P_0 + Q_0 = 0 \quad (22.8)$$

Since the spectral density of the process and measurement noise in a system is typically unknown, designers frequently treat  $Q_0$  and  $R_0$  as additional design parameters that can be manipulated. The bandwidth of the open-loop system frequency response can be controlled using these matrices. This can allow reduction of the susceptibility of the controlled system to noise at higher frequencies.

## 22.4 Ćuk Converter with LQG Compensator

The unit step response of the Ćuk converter controlled by a LQG compensator is shown in Figure 22.4.

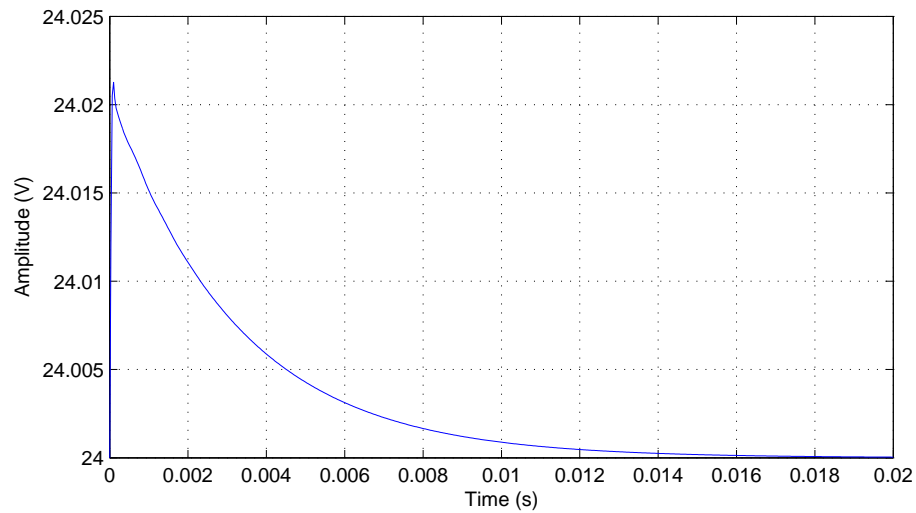


Figure 22.4: Unit step disturbance response of Ćuk converter controlled by LQG compensator.

A comparison of the loop gains of the LQR and LQG controlled systems is shown in Figure 22.5, where the loop is broken at the control input  $d$ . Loss of gain and phase margins has clearly occurred, and the negative gain margin indicates that the system will no longer be closed-loop stable.



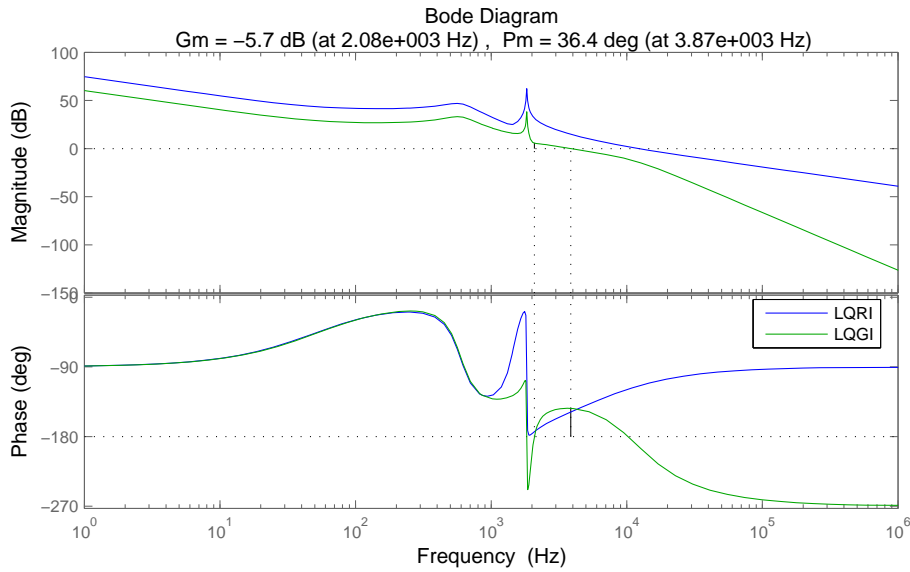


Figure 22.5: Loop gain of Ćuk converter controlled by LQR vs. LQG.

Figure 22.6 shows the control effort produced by the LQG controller.

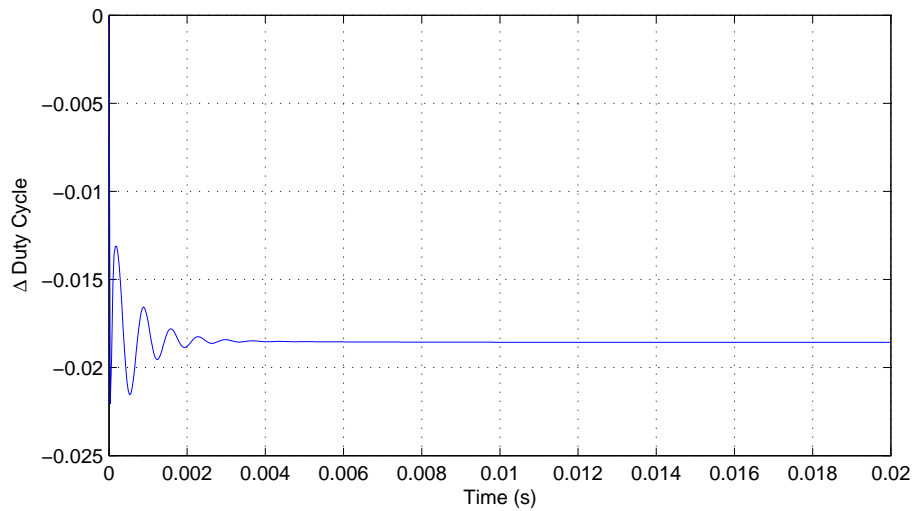


Figure 22.6: LQG control effort.

Minimal discussion is given to this design as it requires the improvements described in the next section.

## 22.5 Control with LQG/LTR Compensators

Newcomers to the world of optimal control might think that the pairing of the optimal controller formed by the solution of the LQR problem and the optimal estimator formed by the construction of a Kalman filter would have optimal properties. This is not necessarily the case. Doyle [5] provided an example to show that the LQG design loses the guaranteed stability margins of the LQR design when feeding back estimated states. Doyle and Stein [6] then showed how this problem could be addressed. In order to recover the good stability margins and sensitivity properties of an LQR design, an iterative procedure known as loop transfer recovery (LTR) may be performed during the LQG design<sup>1</sup>. The Doyle-Stein method of LTR is performed by iteratively increasing the intensity of the noise covariance matrices used in Kalman filter design. LTR causes the open loop frequency domain characteristics associated with the LQR design to be either exactly or asymptotically recovered as an input noise matrix weighting parameter  $q$  is increased. As  $q \rightarrow \infty$ , the properties of the LQR are recovered. Essentially, LTR produces a state estimator whose estimates  $\hat{x}$  are independent of the control input  $u$  (or only weakly dependent on  $u$ ) and dependent only on the input provided by the plant output  $y$ .

Performing the Doyle-Stein method of asymptotic loop transfer recovery on a full-order compensator requires:

- The system must be minimum-phase and strictly proper.
- $R_o = 1$  and  $Q_o = q^2 BB'$

The fact that the plant must be minimum-phase (all poles and zeros in the left half of the complex plane and no pure delays) prevents the compensator designed from being unstable, since the LTR technique moves some of the compensator poles toward the plant zeros where pole-zero cancellation makes them unobservable. (Other LTR methods may be used with nonminimum-phase systems by using subspace [8] or loop-shaping techniques [9].) The remainder of the poles (an excess of poles exists because the plant is strictly proper) move toward infinity in the left-half of the complex plane in a Butterworth filter pattern [10].

The effect of loop transfer recovery is to essentially decouple the observer from the control input  $u$  by raising the observer gain  $L$  so that the state estimate  $\hat{x}$  depends only on the plant output  $y$ . This is illustrated in Figure 22.7. The decoupling is accomplished by the increasing noise intensity on  $u$ , which causes  $L$  to increase such that  $y$  has a larger contribution to the state estimate.

---

<sup>1</sup>A variety of other methods for LTR have been presented in the controls literature, including several which require employment of subspace methods or special coordinate bases [7], [8], but these techniques are beyond the scope of this work.

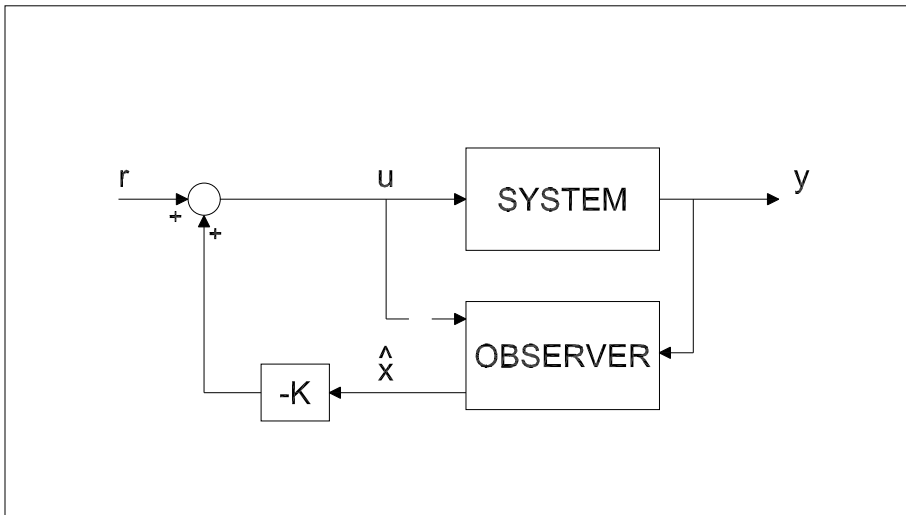


Figure 22.7: Input decoupling effect of loop transfer recovery.

In order to accomplish LTR, the loop is broken at the large  $X$  on the control input shown in Figure 21.1. The loop is explicitly shown in Figure 22.8. The frequency response of the loop from  $d$  to  $d'$  is driven to asymptotically approach the frequency response of the system with LQR control.

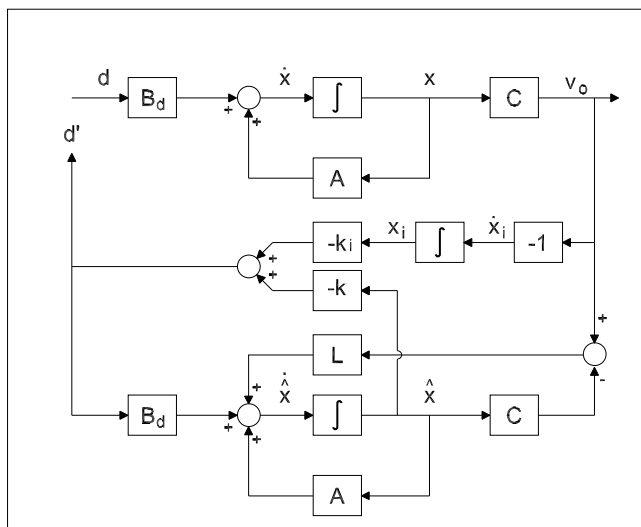


Figure 22.8: The loop to be recovered during loop transfer recovery with a full-order observer.

Figure 22.9 shows the LQGI/LTR iterative design process applied to the Ćuk

converter. Initial LQGI design resulted in a gain margin reduction from infinity to -5.7 dB and a loss of phase margin from  $65.4^\circ$  to  $36.4^\circ$ . The iterative loop transfer recovery process resulted in a controlled system gain margin of 30.2 dB and phase margin of  $61.7^\circ$ . Iteration was stopped at that point since these margins are adequate. Further iterations would result in larger observer gains that may be undesirable during implementation, and recovery of the undesirable high frequency characteristics of the LQR compensator has already begun.

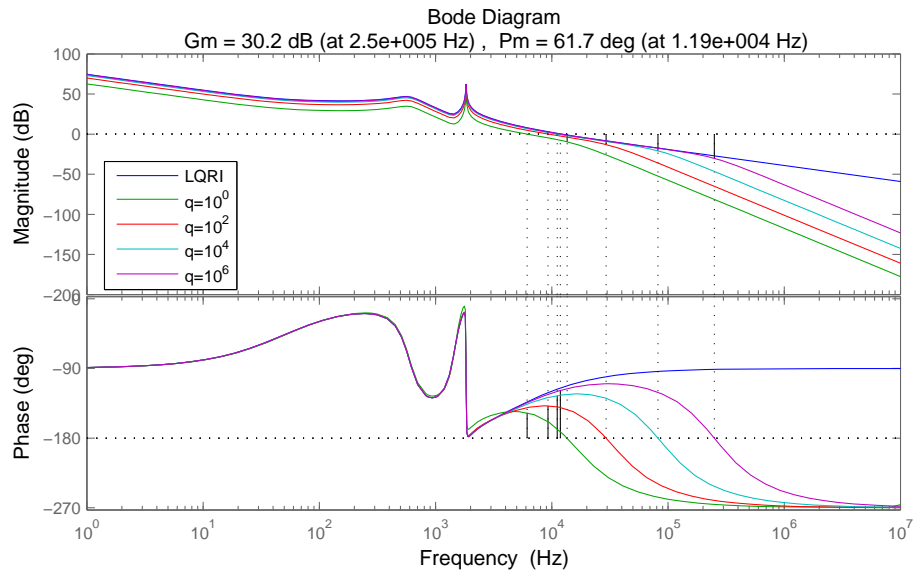


Figure 22.9: Loop transfer recovery of an LQGI compensator.

## 22.6 MATLAB Code

```
1 % Linear Quadratic Regulator Compensator with Integration
2
3 % Define the plant using an external file
4 cuk_parameters;
5
6 % Define the weighting matrices for the integral performance ...
   index, with
7 % weights on the capacitor voltage and the output error
8 Cq = [C 0];
9 Q = Cq'*Cq;
10 Q(1,1) = 1;
11 Q(2,2) = 0;
12 Q(3,3) = 0;
13 Q(4,4) = 0;
14 Q(5,5) = 100000;
15 R = 1;
16
17 A_prime = [A zeros(size(B));-C zeros(size(D))];
18 Bd_prime = [Bd; 0];
19 B_prime = [B;0];
20 C_prime = [C 0];
21
22 % LQR System design
23 K = lqr(A_prime,Bd_prime,Q,R);
24 k = K(1:4);
25 ki = K(end);
26
27 Abar = [A-Bd*k, -Bd*ki; -C, 0];
28 Bbar = [B; 0];
29 Cbar = [C 0];
30
31 SYS_LQR = ss(Abar,Bbar,Cbar,D);
32
33 % LQR Step Response
34 [y t x] = step(SYS_LQR,0.02);
35 figure;
36 plot(t, Vo+y, 'b');
37 xlabel('Time (s)');
38 ylabel('Amplitude (V)');
39 grid on
40
41 % LQR Loop Gain Plot
42 figure;
43 margin(A_prime, Bd_prime, K, 0);
44 h = gcr;
45 h.AxesGrid.TitleStyle.FontSize = 10;
46 h.AxesGrid.XLabelStyle.FontSize = 10;
47 h.AxesGrid.YLabelStyle.FontSize = 10;
48
49 % Control Effort Plot
50 figure;
51 plot(t, -K*x'+d);
52 xlabel('Time (s)');
```

```

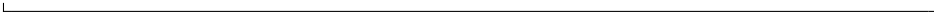
53 ylabel('Duty Cycle');
54 grid on

```

```

1  % Linear Quadratic Gaussian Compensator with Integration
2
3  % Design the LQR compensator
4  lqri;
5
6  % LQG Design
7  Q0 = 1e8*eye(size(A));
8  R0 = 1;
9
10 [L,P,E] = lqe(A,eye(size(A)),C,Q0,R0);
11
12 % Construct the closed-loop system for time-domain response testing
13 Abar = [A,   -Bd*ki, -Bd*k;
14         -C,   0,   zeros(1,4);
15         L*C, -Bd*ki, A-Bd*k-L*C];
16 Bbar = [B; 0; B];
17 Cbar = [C 0 zeros(size(C))];
18 SYS_LQG = ss(Abar,Bbar,Cbar,D);
19
20 % LQG Step Response
21 [y t x] = step(SYS_LQG, 0.02);
22 figure;
23 plot(t, Vo+y, 'b');
24 xlabel('Time (s)');
25 ylabel('Amplitude (V)');
26 grid on
27
28 % LQR vs. LQG System Loop Gain Plot
29 A1 = [A, zeros(4,1), zeros(4,4);
30       -C, 0, zeros(1,4);
31       L*C, -Bd*ki, A-Bd*k-L*C];
32 B1 = [Bd;0;zeros(4,1)];
33 C1 = [zeros(1,4), ki, k];
34 D1 = 0;
35 figure;
36 sys_lqr = ss(A_prime, Bd_prime, K, 0);
37 sys_lqg = ss(A1, B1, C1, D1);
38 margin(sys_lqr)
39 hold on
40 margin(sys_lqg)
41 h = gcr;
42 h.AxesGrid.TitleStyle.FontSize = 10;
43 h.AxesGrid.XLabelStyle.FontSize = 10;
44 h.AxesGrid.YLabelStyle.FontSize = 10;
45 hold off
46 legend('LQRI','LQGI')
47
48 % Control Effort Plot
49 figure;
50 h = plot(t, -[zeros(1,4), ki, k]*x'+d);
51 xlabel('Time (s)');
52 ylabel('Duty Cycle');
53 grid on

```







# References

- [1] B. N. Datta, *Numerical Methods for Linear Control Systems*. San Diego, CA: Elsevier Academic Press, 2004.
- [2] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 10th ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2005.
- [3] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*. Englewood Cliffs, NJ: Prentice Hall, Inc., 1990.
- [4] G. M. Siouris, *An Engineering Approach to Optimal Control and Estimation Theory*. New York, NY: Wiley, 1996.
- [5] J. C. Doyle, “Guaranteed margins for LQG regulators,” *IEEE Trans. Automat. Contr.*, vol. 23, pp. 756–757, Aug. 1978.
- [6] J. C. Doyle and G. Stein, “Robustness with observers,” *IEEE Trans. Automat. Contr.*, vol. 24, pp. 607–611, Aug. 1979.
- [7] B. M. Chen, “Theory of loop transfer recovery for multivariable linear systems,” Ph.D. dissertation, Washington State Univ., Pullman, WA, Dec. 1991.
- [8] A. Saberi and P. Sannuti, “Observer design for loop transfer recovery and for uncertain dynamical systems,” *IEEE Trans. Automat. Contr.*, vol. 35, pp. 878–897, Aug. 1990.
- [9] G. Stein and M. Athans, “The LQG/LTR procedure for multivariable feedback control design,” *IEEE Trans. Automat. Contr.*, vol. 32, pp. 105–114, Feb. 1987.
- [10] B. Friedland, *Control System Design: An Introduction to State-Space Methods*. New York, NY: McGraw-Hill, 1986.



## Chapter 23

# Compensator Order Reduction

With the advent of computer control of systems, high-order system models can be created that allow model-based control methods (such as observer-based compensators) to be easily implemented. These digital controller implementations have many advantages over analog controllers, which may be considered to be outdated. However, analog control can often still be performed at the circuit level with a few discrete components and may be more cost-effective to implement when compared to a microprocessor and its associated support circuitry and programming. Thus, this section examines the idea of controller order reduction for use with analog circuitry. It focuses first on reducing the order of an LQGI/LTR compensator using model reduction techniques, then design of an LQGI/LTR compensator using a reduced-order Kalman filter (ROKF), and finally, application of model reduction techniques to the ROKF-based compensator. This final two-step order reduction technique has not been previously seen in the literature.

### 23.1 Model Reduction of the LQGI/LTR Compensator

Model reduction (MR) may be accomplished by creating a balanced realization (i.e., equal and diagonal controllability and observability Gramian matrices) of the system to be reduced using a linear transformation such as the *balreal* command in MATLAB [1]. Examination of the eigenvalues of the Gramian matrix of the balanced realization allows the designer to identify states that are weakly coupled to both the input and output of the compensator. These states, which are at once both weakly controllable and weakly observable, have small Hankel singular values associated with them. Hankel singular values are determined by taking the square root of the product of the eigenvalues of the controllability

and observability Gramian matrices. In a balanced realization, the Hankel singular values are the diagonal entries of the common controllability/observability Gramian matrix, and the states are ordered from highest to lowest Hankel singular value [2]. Compensator states with small Hankel singular values may be eliminated with little impact to the performance of the compensator using the *modred* command in MATLAB.

In this case, the system to be reduced is the compensator. The *balreal* command identifies only two states that may be eliminated from the LQGI/LTR compensator designed previously without significant loss of accuracy, as may be predicted by the information provided from the pole-zero plot in Figure 23.1. Model reduction resulted in the addition of a high-frequency zero for this particular compensator. This zero was a simulation artifact created by the *modred* command, therefore it was removed by truncation.

Examination of the LQGI/LTR compensator poles, zeros, and gain of the transfer function yields:

$$\begin{aligned} p &= [ -1490 \pm j9000 \quad -1129500 \pm j1129500 \quad 0 ] \\ z &= [ -32410 \quad -319 \quad -1440 \pm j9090 ] \\ k &= 7.195 \times 10^7 \end{aligned} \quad (23.1)$$

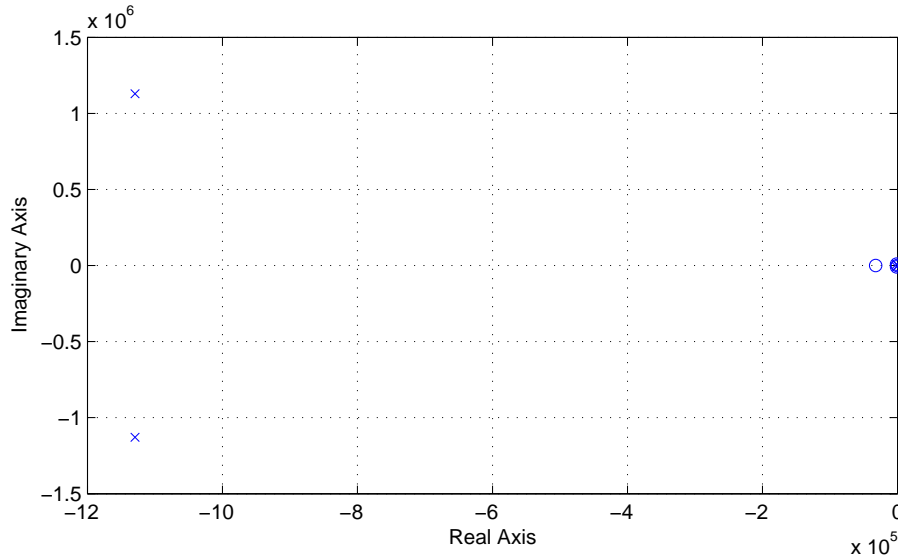


Figure 23.1: Pole-zero plot of the LQGI/LTR compensator.

By comparing the frequencies at which the poles and zeros occur and their complex plane locations, it can be seen that a complex pair of zeros at  $-1490 \pm j9000$  essentially cancels out a complex pair of poles at  $-1440 \pm j9090$ . This is more readily viewed in the close-up view of the pole-zero plot in Figure 23.2. Examination of the compensator after model reduction methods yields:

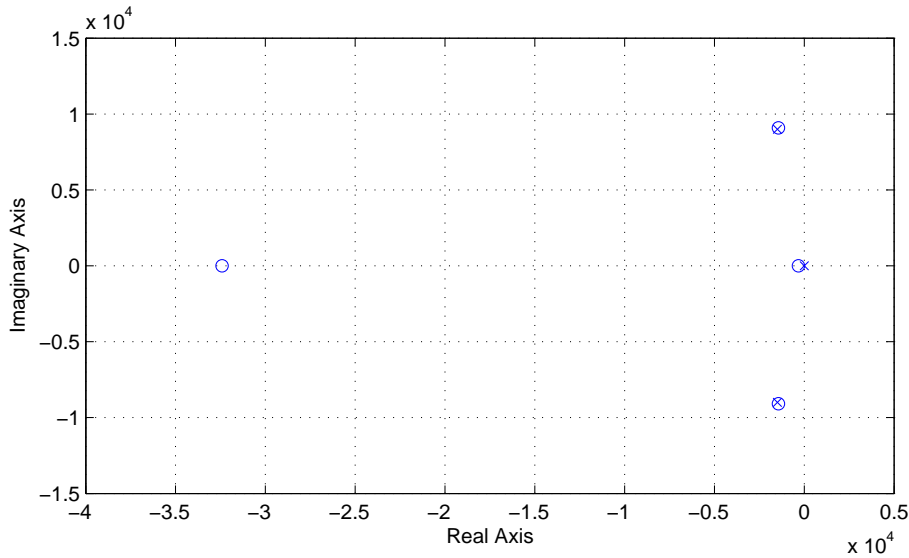


Figure 23.2: Compensator pole-zero plot showing likely pole-zero pair cancellation.

$$\begin{aligned}
 pr &= [ 0 \quad -1126800 \pm j1132300 ] \\
 zr &= [ -319.5 \quad -33035 ] \\
 kr &= 7.18 \times 10^7
 \end{aligned} \tag{23.2}$$

where all poles and zeros are in rad/s.

Once model reduction has been accomplished, validation must occur to verify that significant differences do not exist between the full-order and model-reduced compensators. Comparison of the full-order and model-reduced compensator frequency responses is shown in Figure 23.3. Based on this diagram, it is readily apparent that the model-reduced compensator may be used to replace the full-order compensator without significant difference, thus reducing the controller order. The MRLQGI/LTR compensator can then be achieved with less analog circuitry than the LQGI/LTR yet still perform adequately, and therefore may be selected for implementation.

Care must be taken when performing model reduction to ensure desirable stability margins are not lost. Comparable frequency responses between the full-order and model-reduced compensators tell the designer that margins will probably not change appreciably during model reduction. Verification of margins was performed for system with the MRLQGI/LTR compensator; no change in gain margin resulted but phase margin was reduced to  $61.2^\circ$ , a loss of  $0.5^\circ$ , which is not a significant difference.

The step disturbance response for the model-reduced compensator is shown in Figure 23.4. The step disturbance input is attenuated well, with a maximum

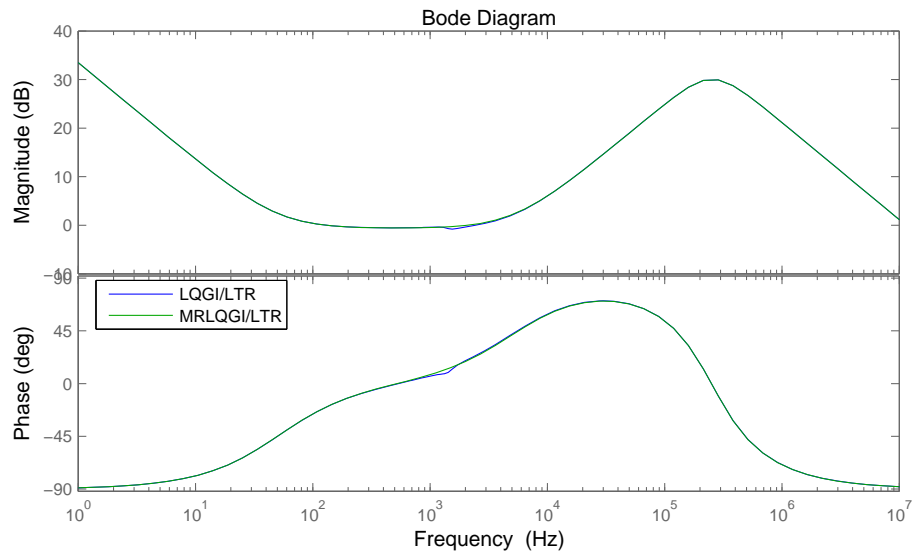


Figure 23.3: Compensator frequency response comparison following model reduction.

deviation of just over 20 mV. This would be a perfectly acceptable controller for the Ćuk converter, however, additional order reduction (though impossible to achieve in this controller incarnation without significantly impacting performance) would be desirable and is explored in the next section.

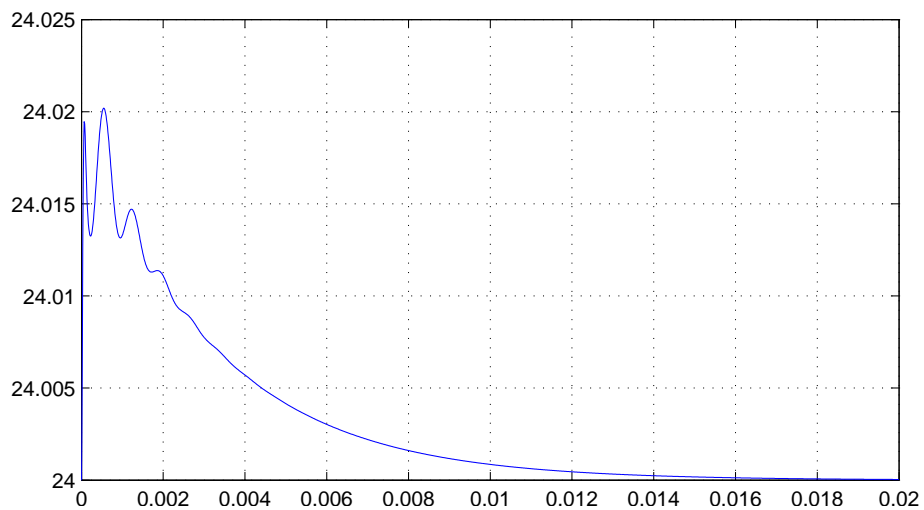


Figure 23.4: Unit step disturbance response of Ćuk converter - MRLQGI/LTR compensator.

## 23.2 A Reduced-Order LQGI/LTR Compensator

The model-reduced compensator designed in Section 23.1 is based on pole-zero cancellation in a balanced state-space realization of the Kalman filter and truncation to remove a high-frequency zero. However, a compensator based on a LQRI paired with a reduced-order Kalman filter (ROKF) can be designed and the LTR technique applied during the design process in order to develop a ROLQGI/LTR compensator. Madiwale and Williams [3] described the mathematics necessary for performing loop transfer recovery with reduced-order linear quadratic compensators and provided proofs that the equations used in this section achieve LTR.

The design of the ROLQGI/LTR compensator begins with construction of a linear quadratic regulator. Once the regulator gain  $K$  has been determined, the system matrices and  $K$  are partitioned to form a reduced-order observer as in Equation 21.10.

Consider a minimum phase system with process noise  $\omega$  (with positive definite Gaussian spectral density matrix  $V$ ) that is distributed into the state dynamics by  $W$  and let the system be free from measurement noise in the measured state vector  $x_m$ :

$$\begin{aligned}\dot{x} &= Ax + Bu + W\omega \\ y &= x_m\end{aligned}\tag{23.3}$$

Process noise characteristics are usually difficult to determine or unknown. Thus, both  $W$  and the corresponding noise spectral density  $V$  are manipulated as design parameters for determining the optimal reduced-order estimator gain  $L$  in a manner similar to the manipulation of  $Q_0$  and  $R_0$  in full-order estimator design methods. Both  $W$  and  $V$  are also partitioned as in Equation 21.10 in order to form the reduced-order Kalman filter. Let:

$$V_{11} = W_1 V_1 W_1' + q^2 B_1 V_2 B_1'\tag{23.4}$$

$$V_{12} = W_1 V_1 W_2' + q^2 B_1 V_2 B_2'\tag{23.5}$$

$$V_{22} = W_2 V_1 W_2' + q^2 B_2 V_2 B_2'\tag{23.6}$$

Then with:

$$\begin{aligned}\bar{A} &= A_{22} - V_{12}' V_{11}^{-1} A_{12} \\ \bar{V} &= V_{22} - V_{12}' V_{11}^{-1} V_{12}\end{aligned}\tag{23.7}$$

where  $V_{11}$  is nonsingular, the following ARE is solved for for  $Q_o$ :

$$\bar{A} Q_o + Q_o \bar{A}' - Q_o A_{12}' V_{11}^{-1} A_{12} Q_o + \bar{V} = 0\tag{23.8}$$

The Kalman filter gain  $L$  can then be determined from:

$$L = (Q_o A_{12}' + V_{12}') V_{11}^{-1}\tag{23.9}$$

Now, as  $q$  is increased in Equations 23.4-23.6, the target feedback loop of the LQR controller is recovered.

Once again, the loop is broken at the large  $X$  on the control input as shown in Figure 21.6. The open loop is shown explicitly in Figure 23.5. The frequency response of the loop from  $d$  to  $d'$  is again driven to asymptotically approach the frequency response of the system with LQR control as shown in Figure 23.6, which is a plot of the iterative LTR process for this reduced-order compensator and the Ćuk converter.

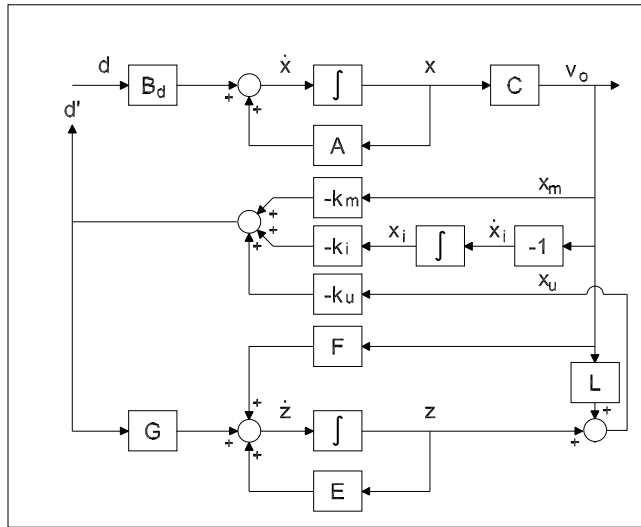


Figure 23.5: Loop transfer recovery with a reduced-order observer.



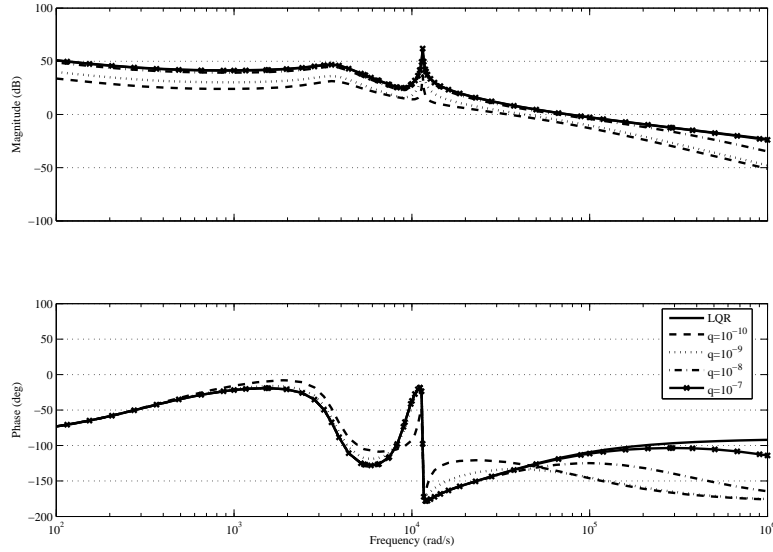


Figure 23.6: Loop transfer recovery of a ROKF-based LQGI compensator.

During simulation, it was found that the original matrices selected to represent the process noise had values that were too large. The initial results showed loop transfer recovery had already occurred, therefore the noise values had to be made smaller to verify that LTR was taking place. For the model of the Čuk converter to be controlled, the following partitions of the  $W$  and  $V$  matrices were used to simulate the fictitious noise:

$$W_1 = \begin{bmatrix} 1 \times 10^{-4} & 0 & 0 \end{bmatrix}, W_2 = \begin{bmatrix} 1 \times 10^{-5} & 0 & 0 \\ 0 & 1 \times 10^{-5} & 0 \\ 0 & 0 & 1 \times 10^{-5} \end{bmatrix}$$

$$V_1 = \begin{bmatrix} 1 \times 10^{-5} & 0 & 0 \\ 0 & 1 \times 10^{-5} & 0 \\ 0 & 0 & 1 \times 10^{-5} \end{bmatrix}, V_2 = 1 \times 10^{-5}$$

The scalar  $q$  was allowed to vary from  $1 \times 10^{-10}$  to  $1 \times 10^{-7}$  during the recovery process.

Examination of the ROLQGI/LTR compensator transfer function yields:

$$\begin{aligned} pr &= \begin{bmatrix} -1490.0 \pm j9000.0 & -2466000.0 & 0.0 \end{bmatrix} \\ zr &= \begin{bmatrix} -32990.0 & -319.2 & -1442.0 \pm j9087.0 \end{bmatrix} \\ kr &= 70.74 \end{aligned} \quad (23.10)$$

where the poles and zeros are given in rad/s. This transfer function shows a pole-zero pair that could possibly be canceled via model reduction methods. The balanced realization approach to model reduction discussed in Section 23.1 was applied. Comparison of the Hankel singular values produced by the *balreal* command revealed that only two states could be eliminated without significant loss of accuracy in the controller model. Elimination of these states resulted in a second-order compensator that had the following characteristics for its transfer function:

$$\begin{aligned} pr &= [ 0.0 \quad -2469000.0 ] \\ zr &= [ -319.4 \quad -33570.0 ] \\ kr &= 70.76 \end{aligned} \tag{23.11}$$

where the poles and zeros are given in rad/s. This model-reduced compensator has a pole at the origin and two zeros on the negative real axis left of the pole, which is a classical PID controller. It also has another pole on the negative real axis beyond the zeros, which corresponds to high frequency low-pass filtering. It may therefore be thought of as a feedback PID compensator with the derivative term acted upon by a first-order low-pass filter. This type of filtered derivative action is typically included in PID controllers to reduce the bandwidth of the controller and associated undesirable amplification of high-frequency noise, as well as to make them implementable (the ideal PID equation is non-causal due to the fact that there is an excess of zeros). This PID controller is very simple to implement with a single inverting operational amplifier configuration. A second inverter buffer stage is needed to eliminate the undesired inversion caused by the first stage.

A frequency response comparison of the two compensators designed in this section is shown in Figure 23.7. There is excellent correlation between the Bode plots, showing that the MRROLQGI/LTR compensator performance will be almost exactly the same as the ROLQGI/LTR compensator.

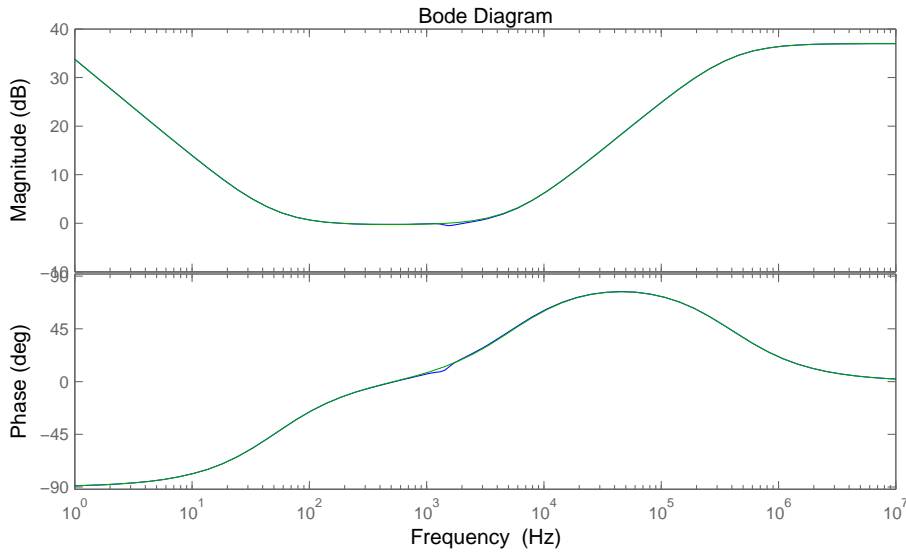


Figure 23.7: Comparison of compensator frequency response - ROLQGI/LTR and MRROLQGI/LTR.

Initial ROLQGI design resulted in no gain margin reduction from infinity but had a loss of phase margin from  $65.4^\circ$  to  $56.3^\circ$ . The iterative loop transfer recovery process resulted in a controlled system phase margin of  $63.7^\circ$ . Further iterations would result in larger observer gains that may be undesirable during implementation. Verification of margins was performed for system with the MRROLQGI/LTR compensator; no change in gain margin resulted but phase margin was reduced to  $63.3^\circ$ , a loss of  $0.4^\circ$ . Again, this is not a significant difference.

The step disturbance response for the model-reduced compensator is shown in Figure 23.8. The step disturbance input is attenuated well, with a maximum deviation of less than 20 mV. This is a perfectly acceptable controller for the Ćuk converter.

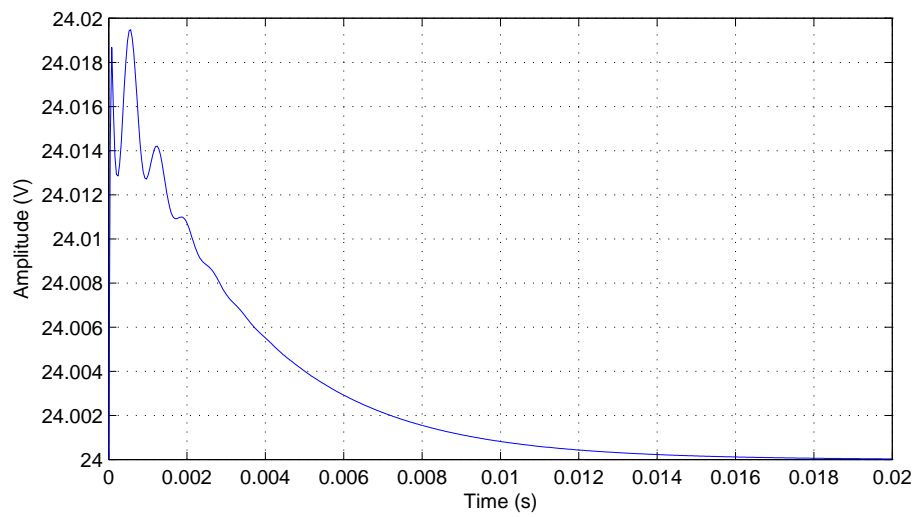


Figure 23.8: Unit step  $v_g$  response of Ćuk converter - MRROLQGI/LTR compensator

## 23.3 MATLAB Code

```
1 % LQGI/LTR and MRLQGI/LTR Compensators
2 clc
3
4 % Design the LQR compensator
5 lqri;
6
7 % Define one of the Kalman filter weight matrices
8 R0 = 1;
9
10 % Plot the magnitude of the open loop LQR
11 h=figure;
12 bode(A_prime, Bd_prime, K, 0);
13 hold on
14
15 % Loop transfer recovery
16 q = [1,1e2,1e4,1e6];
17 for n = 1:length(q)
18     Q0 = q(n)*Bd*Bd';
19     [L,P,E] = lqe(A,eye(size(A)),C,Q0,R0);
20     Aol = [ A, zeros(4,1), zeros(4,4);
21           -C, 0, zeros(1,4);
22           L*C, -Bd*ki, A-Bd*k-L*C];
23     Bol = [Bd; 0; zeros(4,1)];
24     Col = [zeros(1,4), ki, k];
25     sysol = ss(Aol,Bol,Col,D);
26
27     % Plot the frequency response of the open loop LQGI/LTR
28     margin(Aol, Bol, Col, D);
29 end
30 hold off
31 childrenHnd =get(h, 'Children')
32 axes(childrenHnd(3))
33 legend('LQRI','q=10^{0}','q=10^{2}','q=10^{4}','q=10^{6}', ...
34        'Location','SouthWest')
35 h = gcr;
36 h.AxesGrid.TitleStyle.FontSize = 10;
37 h.AxesGrid.XLabelStyle.FontSize = 10;
38 h.AxesGrid.YLabelStyle.FontSize = 10;
39
40 % Compensator
41 Ao = [A-Bd*k-L*C, -Bd*ki;
42       zeros(1,4), 0];
43 Lo = [L; -1];
44 focomp = ss(Ao,Lo,K,0);
45 [zc,pc,kc]=ss2zp(Ao,Lo,K,0);
46 focomp = zpk(zc,pc,kc)
47
48 % Pole-zero plot of full-order compensator
49 [P,Z] = pzmap(focomp);
50 figure;
51 h = plot(real(P),imag(P),'x',real(Z),imag(Z),'bo');
52 ylabel('Imaginary Axis');
53 xlabel('Real Axis');
```

```

54 grid on
55
56 % Close-up of pole-zero plot
57 figure;
58 h = plot(real(P), imag(P), 'x', real(Z), imag(Z), 'bo');
59 ylabel('Imaginary Axis');
60 xlabel('Real Axis');
61 axis([-4e4, 0.5e4, -1.5e4, 1.5e4]);
62 grid on
63
64 % Reduce the order of the compensator
65 [focomp, g] = balreal(focomp);
66 g;
67 elim = (g < 5e-2);
68 mrcomp = modred(focomp, elim);
69 [zrc prc krc] = ss2zp(mrcomp.a, mrcomp.b, mrcomp.c, mrcomp.d);
70 % An added high-frequency zero results from model reduction.
71 % Restore the system gain prior to removing the high frequency zero
72 % using truncation.
73 krc = krc * abs(zrc(1, 1));
74 zrc = zrc(2:3, 1);
75 mrcompzp = zp2ss(zrc, prc, krc)
76 [mrcomp.a, mrcomp.b, mrcomp.c, mrcomp.d] = zp2ss(zrc, prc, krc);
77
78 % Compensator Comparison Bode Plot
79 figure;
80 bode(focomp, mrcomp);
81 h = gcr;
82 h.AxesGrid.TitleStyle.FontSize = 10;
83 h.AxesGrid.XLabelStyle.FontSize = 10;
84 h.AxesGrid.YLabelStyle.FontSize = 10;
85 legend('LQGI/LTR', 'MRLQGI/LTR', 'Location', 'NorthWest');
86
87 % Plot step disturbance response with MRLQGI/LTR compensator
88 figure;
89 sys = ss(A, [B, Bd], C, D);
90 syscl = feedback(sys, mrcomp, [2], [1]);
91 [y, t, x] = step(syscl, 0.02);
92 plot(t, Vo + y(:, 1, 1));
93 grid on

```

```

1 % ROLQGI/LTR and MRROLQGI/LTR Compensators
2
3 % Design the LQR compensator
4 lqri;
5
6 % Partition the system in terms of measured and unmeasured states
7 A11 = A(1, 1);
8 A12 = A(1, 2:4);
9 A21 = A(2:4, 1);
10 A22 = A(2:4, 2:4);
11
12 Bd1 = Bd(1);
13 Bd2 = Bd(2:4);
14
15 % Partition the gain vector determined by linqadreg.m

```

```

16 k = K(1:4);
17 km = K(1);
18 ku = K(2:4);
19 ki = K(end);
20
21 % Form the open-loop LQR system and plot the frequency response
22 olsys = ss(A_prime,Bd_prime,K,0);
23 [gml pml wgcl wpc1] = margin(A_prime,Bd_prime,K,0);
24 gml = 20*log10(gml);
25 h=figure;
26 bode(olsys,{1e1,1e7});
27 hold on
28
29 % Noise distribution matrices
30 F1 = 1e-4*eye(1,3);
31 F2 = 1e-5*diag([1,1,1]);
32
33 % White noise vector intensity
34 V1 = 1e-5*diag([1,1,1]);
35 V2 = 1e-5*1;
36
37 % Loop transfer recovery
38 q = [1e-10,1e-9,1e-8,1e-7];
39 for n = 1:length(q);
40     V11 = F1*V1*F1'+q(n)^2*Bd1*V2*Bd1';
41     V12 = F1*V1*F2'+q(n)^2*Bd1*V2*Bd2';
42     V22 = F2*V1*F2'+q(n)^2*Bd2*V2*Bd2';
43
44     Abar = A22-V12'*inv(V11)*A12;
45     Vbar = V22-V12'*inv(V11)*V12;
46
47     [Q,M,X] = care(Abar',A12',Vbar,V11);
48
49     L = (Q*A12'+V12')*inv(V11);
50
51     E = A22-L*A12;
52     F = E*L+A21-L*A11;
53     G = Bd2-L*Bd1;
54
55     % Build open loop system with plant, observer, and feedback law
56     Aol = [A, zeros(4,1), zeros(4,3);
57           -C, 0, zeros(1,3);
58           F*C-G*km*C-G*ku*L*C, -G*ki, E-G*ku];
59     Bol = [Bd; 0; zeros(3,1)];
60     Col = [km*C+ku*L*C, ki, ku];
61     sysol = ss(Aol,Bol,Col,D);
62
63     % Plot the frequency response of the open loop system
64     margin(sysol);
65 end
66 hold off
67 childrenHnd =get(h, 'Children');
68 axes(childrenHnd(3))
69 legend('LQRI','q=10^{-10}','q=10^{-9}','q=10^{-8}',...
70 'q=10^{-7}','Location','SouthWest')
71 h = gcr;
72 h.AxesGrid.TitleStyle.FontSize = 10;

```

```

73 h.AxesGrid.XLabelStyle.FontSize = 10;
74 h.AxesGrid.YLabelStyle.FontSize = 10;
75
76 % Define the reduced order compensator design
77 Ar = [E-G*ku, -G*ki;
78       zeros(1,3), 0];
79 Br = [F-G*ku*L-G*km; -1];
80 Cr = [ku ki];
81 Dr = ku*L+km;
82 [zr pr kr] = ss2zp(Ar,Br,Cr,Dr);
83 rocomp = zpk(zr,pr,kr);
84
85 % Apply model reduction techniques
86 [rocomp,g] = balreal(rocomp);
87 g
88 elim = (g<5e-2)
89 mrrocomp = modred(rocomp,elim);
90 [zrr prr krr] = ss2zp(mrrocomp.a,mrrocomp.b,mrrocomp.c,mrrocomp.d);
91 mrrocompzp = zpk(zrr,prr,krr)
92
93 % Compensator Comparison Bode Plot
94 figure;
95 bode(rocomp,mrrocomp);
96 h = gcr;
97 h.AxesGrid.TitleStyle.FontSize = 10;
98 h.AxesGrid.XLabelStyle.FontSize = 10;
99 h.AxesGrid.YLabelStyle.FontSize = 10;
100
101 % Plot step disturbance response
102 figure;
103 sys = ss(A, [B,Bd],C,D);
104 syscl = feedback(sys,mrrocomp,[2],[1]);
105 [y,t,x] = step(syscl,0.02);
106 plot(t, 24+y(:,1,1));
107 xlabel('Time (s)');
108 ylabel('Amplitude (V)');
109 grid on
110
111 % Calculate component values for the MRROLQGI/LTR compensator
112 % There are five free parameters to find and four equations.
113 % Arbitrarily assign R3, calculate the remaining component values.
114 R3 = 100000
115 R2 = R3/krr
116 C2 = 1/(R3*abs(zrr(1,1)))
117 R1 = R2*abs(prr(2,1))/abs(zrr(2,1))-R2
118 C1 = 1/(R1*abs(zrr(2,1)))

```



# References

- [1] *MATLAB Control Systems Toolbox 6.0 User's Guide*, The Mathworks, 2004.
- [2] R. S. Sánchez-Peña and M. Sznaier, *Robust Systems: Theory and Applications*. New York, NY: Wiley, 1998.
- [3] A. N. Madiwale and D. E. Williams, "Some extensions of loop transfer recovery," in *Proc. of the American Control Conference (ACC'85)*, Boston, MA, Jun. 1985, pp. 790–795.



## Chapter 24

# Compensator Implementation

At this point, four optimal compensator designs that rely on output feedback and state estimation have been developed. The initial design of the optimal compensator began with a fifth-order controller, with estimated states for each of the four states in the plant and an augmented integral state (LQGI/LTR). Model reduction techniques applied directly resulted in a third-order controller (MRLQGI/LTR), which was a significant improvement in terms of minimizing the circuitry for implementation. The final design began with a fourth-order compensator based on three states from a reduced-order observer augmented by an integral state (ROLQGI/LTR), to which model reduction techniques were applied to form a second-order transfer function (MRROLQGI/LTR). This final regulator had the form of a classical PID controller. The final design had a significant reduction in circuitry yet maintained excellent performance in both the time and frequency domains.

The reduced-order compensators developed in Chapter 23 using model reduction techniques can be implemented with analog circuits using operational amplifiers. The goal is to use the minimum amount of components and circuitry for control (to minimize manufacturing costs) while maintaining adequate controller performance.

### 24.1 MRLQGI/LTR Compensator Construction

The model-reduced compensator designed in Section 23.1 has two zeros on the negative real axis, a LHP pair of complex poles, and a pole at the origin. This compensator can be implemented by a PI compensator cascaded with a Tow-Thomas biquadratic filter as seen in Figure 24.1. The Tow-Thomas filter is a general second-order filter able to implement low-pass, high-pass, band-pass, notch, and all-pass filters with appropriate choices of coefficients of the filter transfer function [1]:

$$T_b(s) = \frac{a_2 s^2 + a_1 s + a_0}{b_2 s^2 + b_1 s + b_0} \quad (24.1)$$

The biquad filter section is used to implement the second-order denominator term with a simple zero by setting  $a_2 = 0$ . The PI section is used to implement the other real zero and the pole at the origin. For both sections, it is desirable to keep the poles and zeros within a certain frequency tolerance of each other to keep the component values reasonable in size.

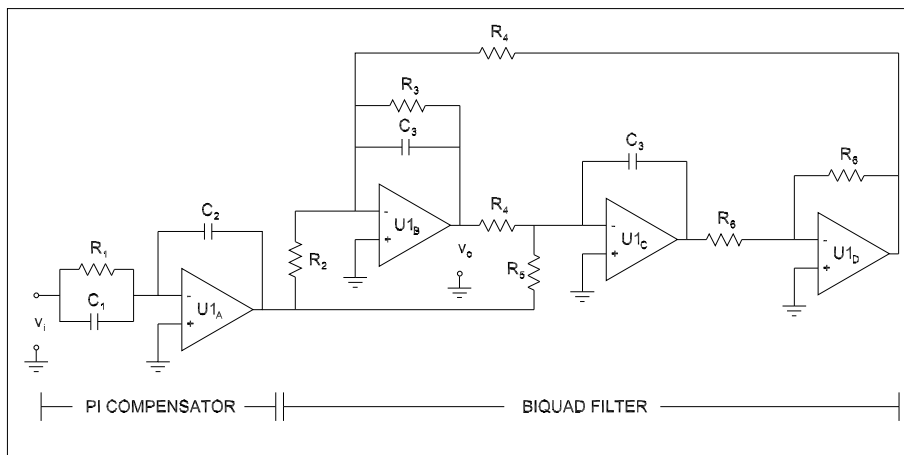


Figure 24.1: The MRLQGI/LTR compensator implemented using operational amplifiers.

The transfer function of the PI section is given by:

$$T_p(s) = -\frac{C_1}{C_2} \left( s + \frac{1}{C_1 R_1} \right) \quad (24.2)$$

The transfer function of the biquad filter section is given by:

$$T_b(s) = -\frac{\frac{1}{C_3 R_2} s + \frac{1}{C_3^2 R_4 R_5}}{s^2 + \frac{1}{C_3 R_3} s + \frac{1}{C_3^2 R_4^2}} \quad (24.3)$$

## 24.2 MRROLQGI/LTR Compensator Construction

A practical example of the MRROLQGI/LTR compensator requires fewer components to implement than the MRLQGI/LTR circuit. It is simply an analog PID controller with a filtered derivative term [2]. Consider the circuit in Figure 24.2. With the inverting opamp configuration and some algebraic manipu-

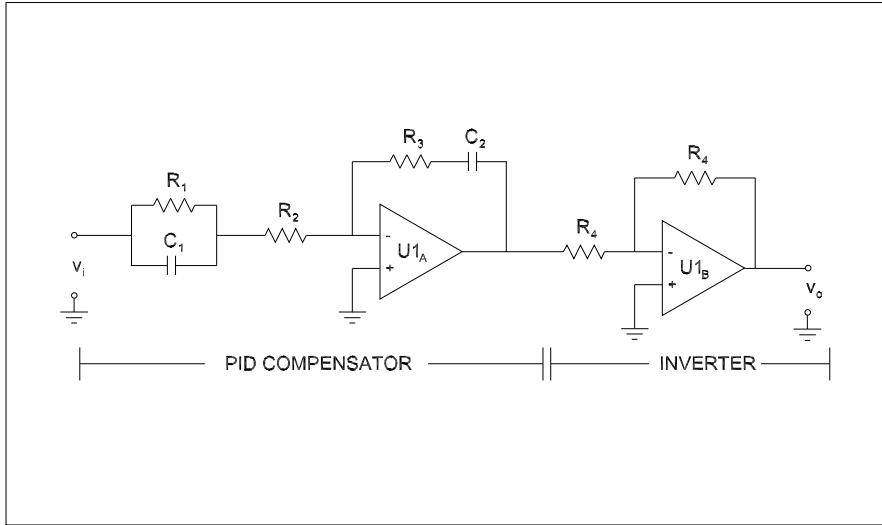


Figure 24.2: The analog MRROLQGI/LTR compensator implementation.

lation, the transfer function is shown to be:

$$T(s) = \frac{R_3 \left( s + \frac{1}{C_2 R_3} \right) \left( s + \frac{1}{R_1 C_1} \right)}{s \left( s + \frac{R_2 + R_1}{R_1 R_2 C_1} \right)} \quad (24.4)$$

which shows the following zeros, poles, and gain:

$$\begin{aligned} \text{poles} &= \left[ 0 \quad \frac{-(R_2 + R_1)}{R_1 R_2 C_1} \right] \\ \text{zeros} &= \left[ \frac{-1}{C_2 R_3} \quad \frac{-1}{R_1 C_1} \right] \\ \text{gain} &= \frac{R_3}{R_2} \end{aligned} \quad (24.5)$$

Component values may be determined by equating the expressions in Equation 24.5 with the compensator values given in Equation 23.11. Since there are fewer equations than unknowns, one of the component values must be fixed before the other component values may be determined. The final design used the component values given in Table 24.1.

Table 24.1: Component values for MRROLQGI/LTR compensator implementation.

<b>Component</b>	<b>Value</b>
$R_1$	$100\text{ k}\Omega$
$R_2$	$1.4\text{ k}\Omega$
$R_3$	$100\text{ k}\Omega$
$C_1$	$0.29\text{ nF}$
$C_2$	$31.3\text{ nF}$

# References

- [1] R. C. Jaeger, *Microelectronic Circuit Design*. Boston, MA: McGraw-Hill, 1997.
- [2] W. J. Palm, *Control Systems Engineering*. New York, NY: Wiley, 1986.





## Chapter 25

# Power Electronic Circuit Simulation

The MRROLQGI/LTR compensator required the least amount of circuitry and was therefore selected for implementation and testing in PECS. The PECS environment provides for relatively short run times compared to SPICE-based circuit simulation environments since power electronics circuits can typically be simulated with simpler component models than other types of analog circuits.

### 25.1 Simulating the Controlled Ćuk Converter in PECS

The PECS implementation of the MRROLQGI/LTR compensator design from Section 24.2 connected to the Ćuk converter can be seen in Figure 25.1.

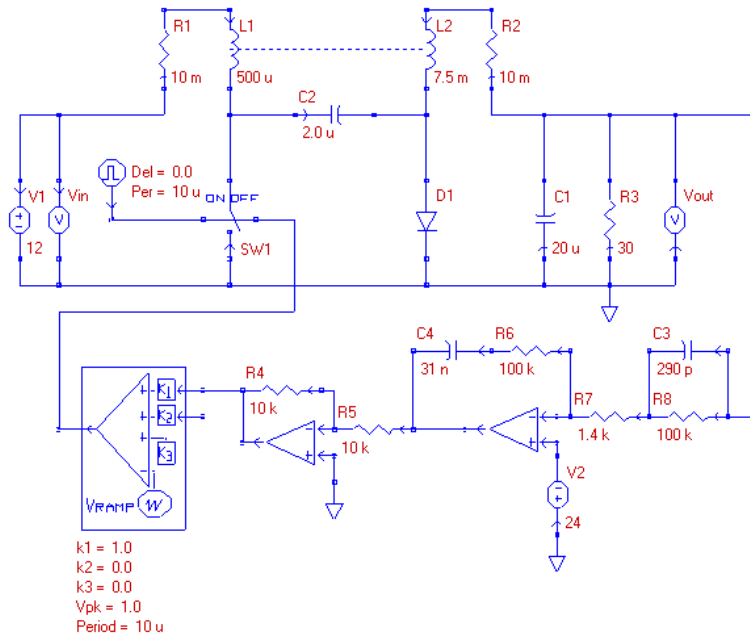


Figure 25.1: PECS simulation of Ćuk converter - MRROLQGI/LTR compensator.

First, the small-signal operation was tested using a step disturbance in  $v_g$  as was used during controller design. A simulation was set up to provide for input voltage steps up and down from the nominal 12 V using a unit step to 13 V on the Ćuk input. The simulation used a time step of  $1 \mu\text{s}$  and a run time of 0.1 s. The results of this simulation are shown in Figure 25.2.

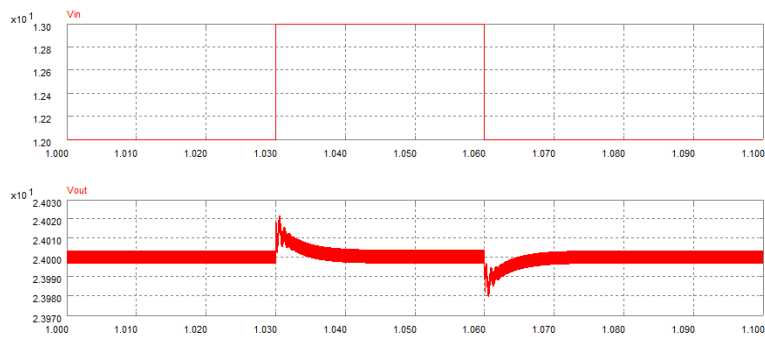


Figure 25.2: Unit step  $v_g$  disturbances and  $v_o$  response - MRROLQGI/LTR compensator.

The maximum deviation in the output voltage is only 0.022 V in response to a step input of 1 V. (Note that ripple voltage exists in the simulation which is not present in the state space averaged model simulations in MATLAB. Neglecting the ripple did not pose a problem, however, during the control system design process.)

Next, a large-signal simulation was set up to provide for input voltage steps up and down from the nominal 12 V through a specified operating range of 9-14 V on the Ćuk input. The results of the large-signal simulation are shown in Figure 25.3.

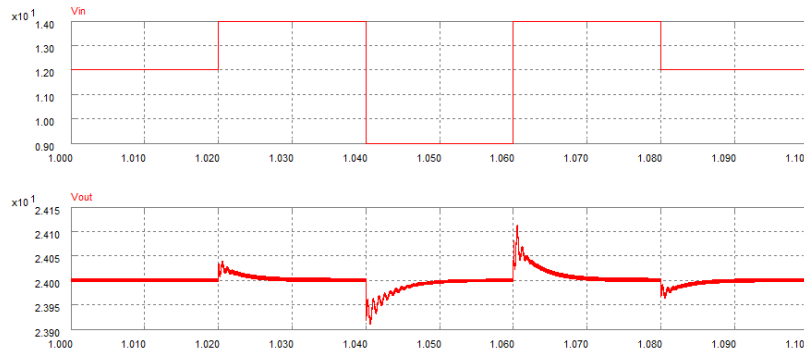


Figure 25.3: Large signal  $v_g$  disturbances and  $v_o$  response - MRROLQGI/LTR compensator.

The maximum deviation of 0.11 V in the output voltage occurred when the input voltage drops sharply from 14 V to 9 V but is rapidly brought back under control. This value is well within the 0.24 V tolerance that was specified for controller performance.

Finally, a simulation was set up to provide for 25% load current steps (0.214 A) around the nominal load current of 0.857 A. The results of this simulation are shown in Figure 25.4.

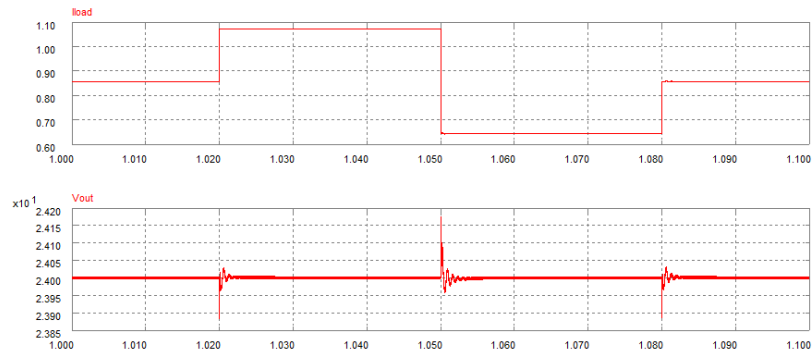


Figure 25.4: Load current disturbance and  $v_o$  response - MRROLQGI/LTR compensator.

The maximum deviation of 0.175 V in the output voltage occurred when the load current dropped sharply from 1.071 A to 0.643 A, but this is within the transient design specification of regulation to within 1% of the nominal output voltage.

The ability to reject large scale input voltage and load disturbances shows that the compensator design is excellent. The MRROLQGI/LTR controller allowed the system to not only meet the performance specifications, but to exceed them, and achieved these results after two separate reduction of order techniques (reduced-order observer and model reduction) were applied during the design process.

## Chapter 26

# Conclusion

Part II demonstrated the process of applying modern control design methods to regulator design for DC-DC converters, and prototypically, to a Ćuk DC-DC converter. Full state feedback control for pole placement was applied - first without integral effort, then with an integrator added. Output compensation using state estimation techniques with full- and reduced-order observers was discussed and simulated. LQRI and LQGI/LTR techniques were then used to design compensators that were optimized with respect to quadratic performance indices based on state and control effort transients. Balanced realization of the LQGI/LTR compensator identified weakly controllable/observable states, which were then removed using truncation to create a third-order compensator (MR-LQGI/LTR) that could be implemented with a combination PI/Tow-Thomas biquad circuit.

The steps up to this point were typical application of modern state space control methods found in most textbooks. The method that followed (in Section 23.2) resulted in a minimal-order compensator. Namely, when the reduced-order Kalman filter-based compensator design (ROLQGI/LTR) was combined with model reduction using a balanced realization, the result was a second-order compensator (MRROLQGI/LTR) that could be implemented by an analog PID circuit. This MRROLQGI/LTR controller was applied to the Ćuk converter in a power electronics modeling environment and showed excellent input voltage and load current disturbance rejection abilities. The two-step order reduction combination of reduced-order observer design and truncated balanced realization is a novel design method that, while applied to one of the more complex DC-DC converters, appears to be applicable to compensator design in general.



# Appendix C

## PECS

### Power Electronics Circuit Simulator

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

#### C.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.

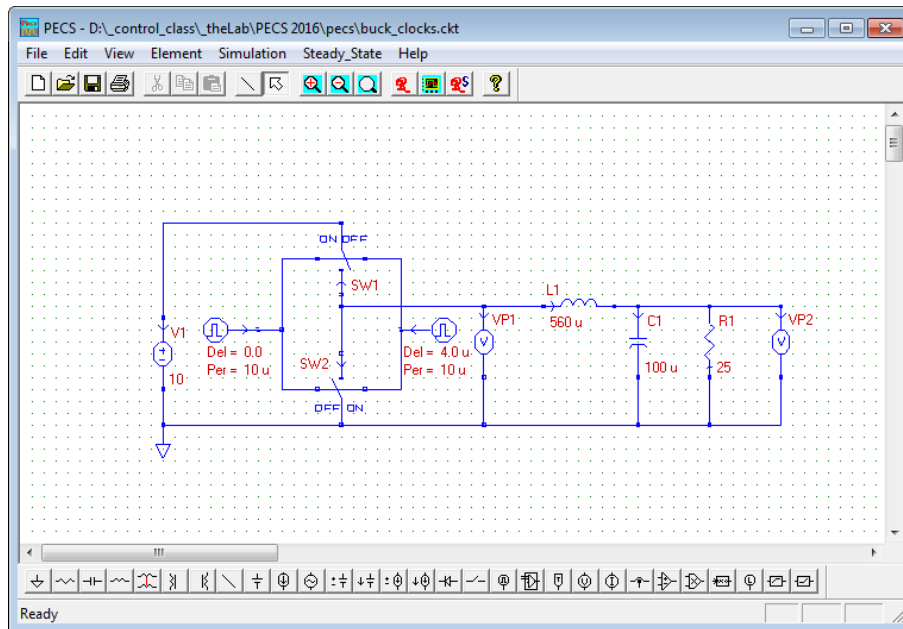


Figure C.1: The PECS program showing a circuit schematic.

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

These will be expanded on next:

### C.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.

## C.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 C.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*.

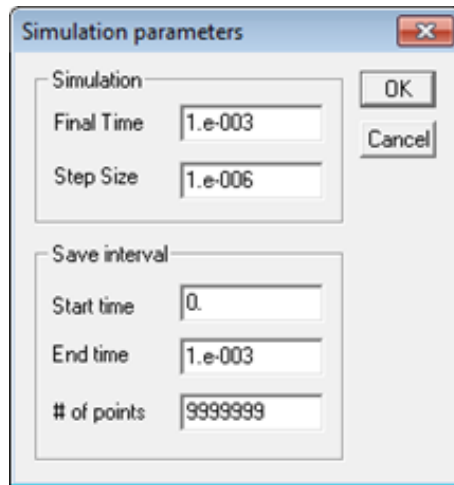


Figure C.2: Simulation parameters dialog window showing the default values.

*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.

### C.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.

### C.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.

## C.3 Elements

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

#### Resistor:

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



Figure C.3: Resistor schematic symbol

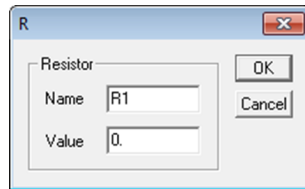


Figure C.4: Resistor dialog window

#### Capacitor:

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



Figure C.5: Capacitor schematic symbol

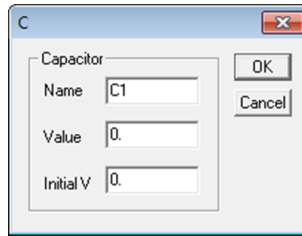


Figure C.6: Capacitor dialog window

**Inductor:**

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



Figure C.7: Inductor schematic symbol

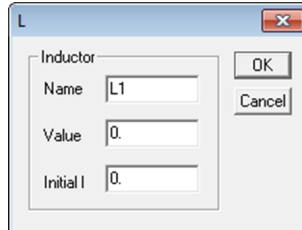


Figure C.8: Inductor dialog window

**C.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 C.9. The value of the voltage is a user specified parameter that is entered in the dialog window, shown in Figure C.10.



Figure C.9: DC Voltage Source schematic symbol

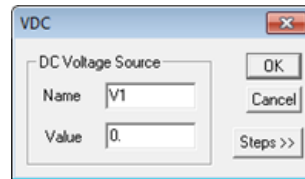


Figure C.10: DC Voltage Source

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 C.11. If step values are present, the dialog window will appear fully expanded on initiation.

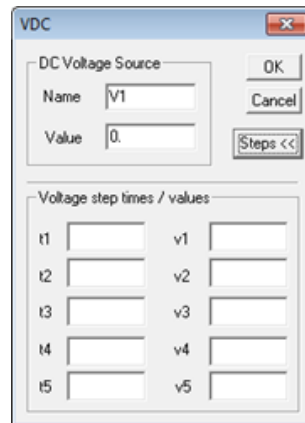


Figure C.11: DC Voltage Source, showing voltage step times and values

#### **DC Current Source:**

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



Figure C.12: DC Current Source schematic symbol

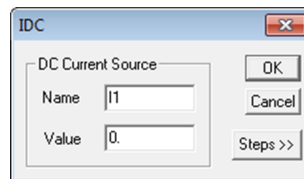


Figure C.13: DC Current Source dialog window

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 C.14. If step values are present, the dialog window will appear fully expanded on initiation.

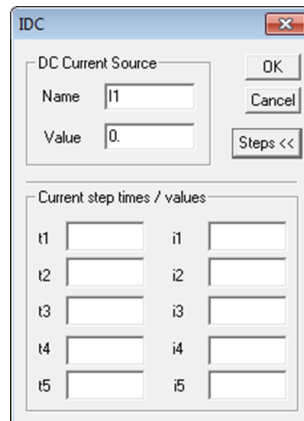


Figure C.14: DC Current Source dialog window, showing voltage step times and values

### AC Voltage Source:

The schematic symbol of an AC voltage source is shown in Figure C.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 C.16, which will specify its characteristics.





Figure C.15: AC Voltage Source schematic symbol

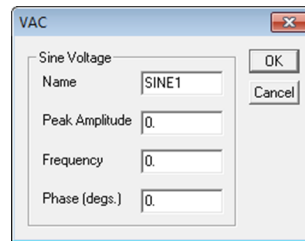


Figure C.16: AC Voltage Source dialog window

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

#### Voltage Port:

The schematic symbol of a voltage port is shown in Figure C.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 C.18.



Figure C.17: Voltage port schematic symbol

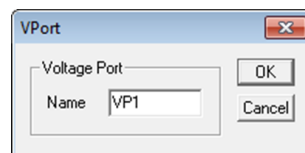


Figure C.18: Voltage Port dialog window

#### Current Port:

The schematic symbol of a current port is shown in Figure C.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 C.20.

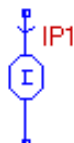


Figure C.19: Current Port schematic symbol

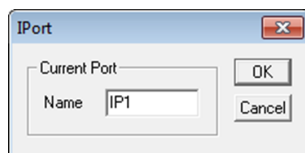


Figure C.20: Current Port dialog window

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

#### Switch:

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



Figure C.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 C.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.

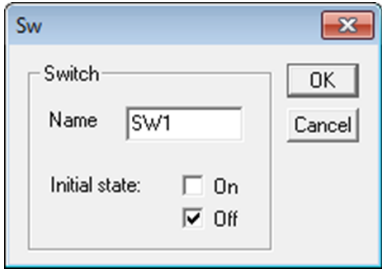


Figure C.22: Switch dialog window

**Diode:**

The schematic symbol of a diode is shown in Figure C.23, with the dialog window shown in Figure C.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 C.23: Diode schematic symbol

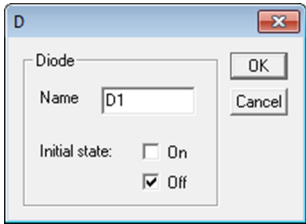


Figure C.24: Diode dialog window

### C.3.5 Switch control elements: Clk (Clock) and (MOD) PWM Modulator

#### Clock:

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



Figure C.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 C.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 C.27.

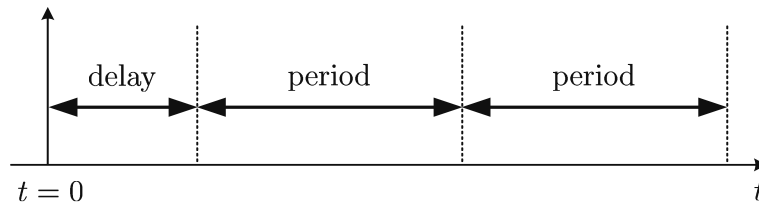


Figure C.26: Delay and period parameters shown pictorially

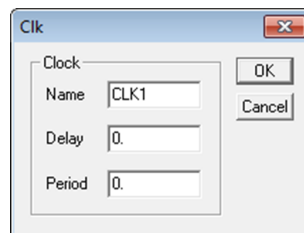


Figure C.27: Clock dialog window

### Pulse Width Modulator:

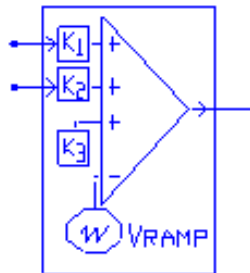


Figure C.28: Pulse Width Modulator schematic symbol

The schematic symbol of a PWM modulator is shown in Figure C.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 voltages  $v_1$  and  $v_2$  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 C.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.

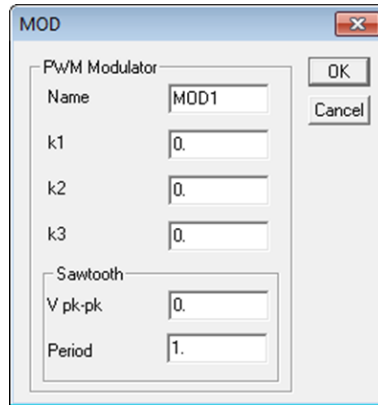


Figure C.29: Pulse Width Modulator dialog window

### C.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 C.30.

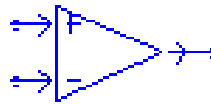


Figure C.30: Operational Amplifier schematic symbol

The model of the op-amp used in PECS is shown in Figure C.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 C.32. Generally the default values will need not be changed.

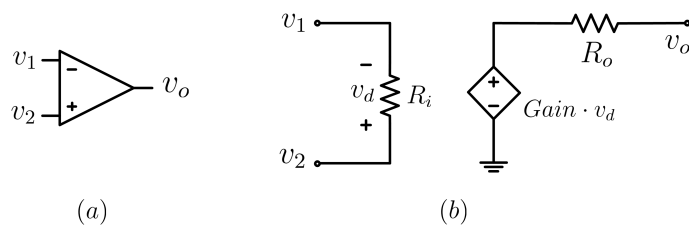


Figure C.31: (a) Op-amp, and (b) its corresponding model

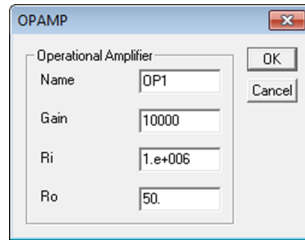


Figure C.32: Operational Amplifier dialog window, showing the default parameter values.

**Current sensor:**

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



Figure C.33: Current sensor schematic symbol

**Ground node:**

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



Figure C.34: Ground node schematic symbol

## C.4 PECSPLOT

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.

## C.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 C.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.

## C.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 C.36.

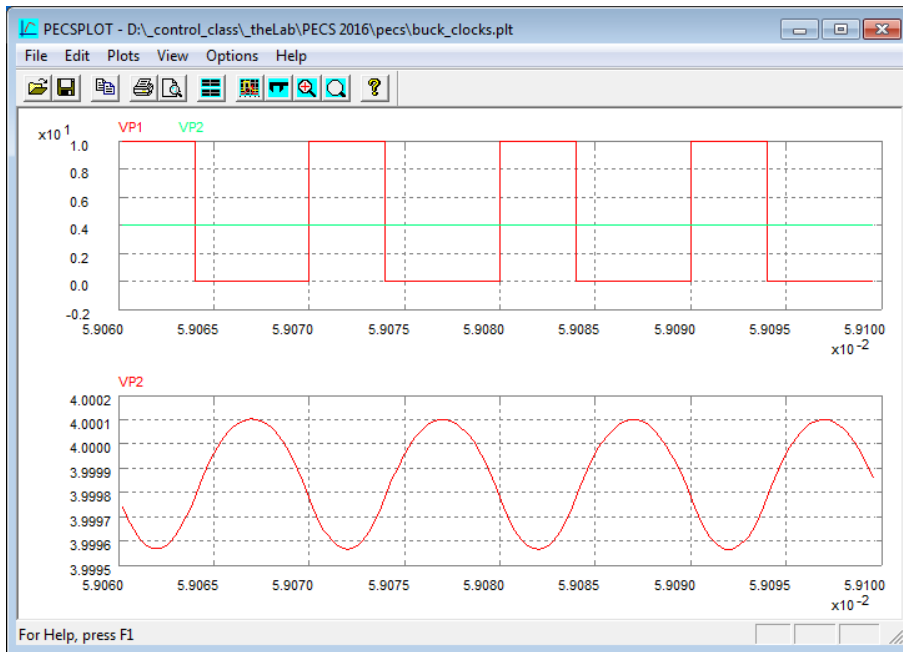


Figure C.35: PECS PLOT displaying two plots. The top plot show two waveforms and the bottom show only one.

## C.7 Menu items

As seen in Figure C.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.

### C.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

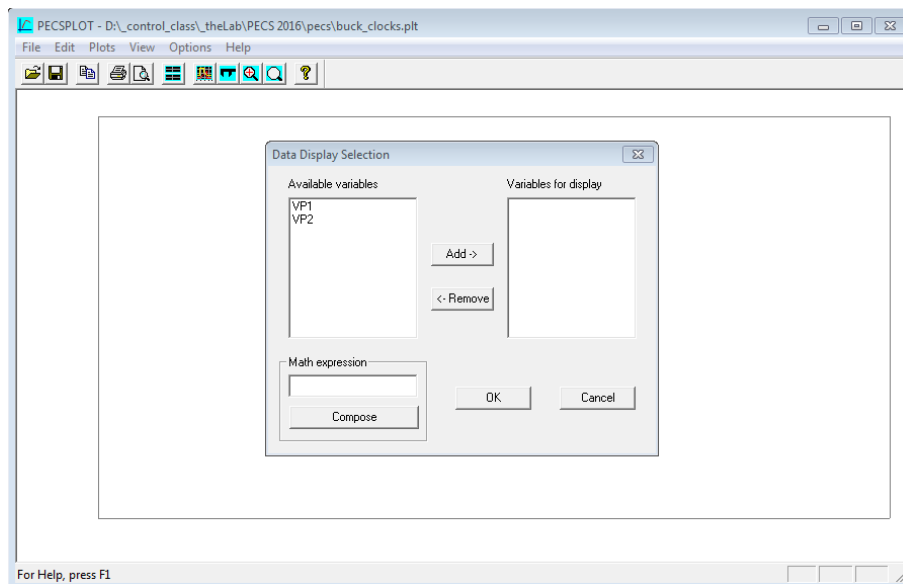


Figure C.36: Selection of waveforms dialog window.

## C.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 PECSPLOT file name (top left) and the current date and time (top right).

## C.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 C.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 C.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 C.1.

Table C.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.

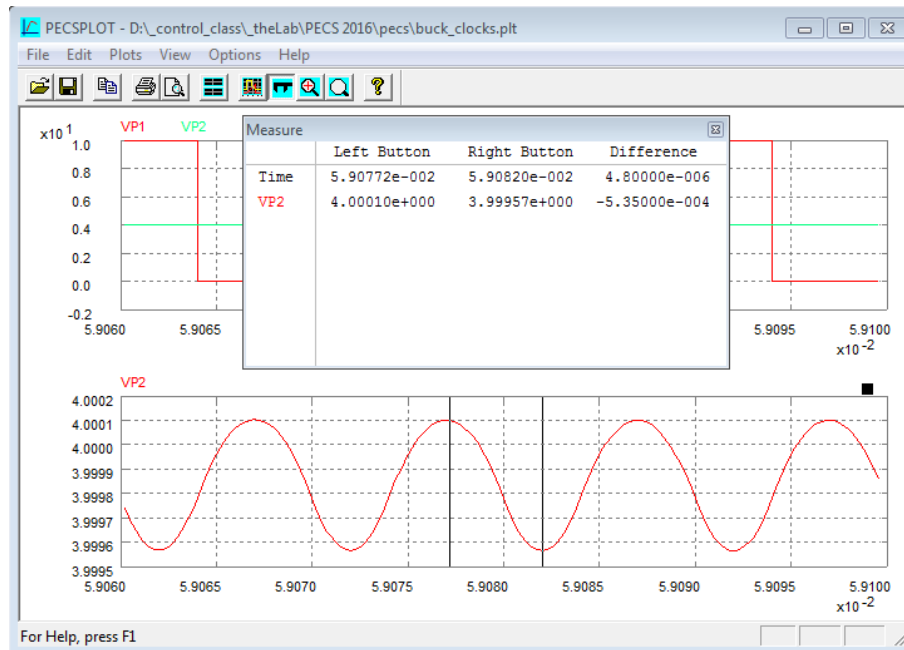


Figure C.37: The measure display dialog usually appears at the lower left corner of the computer screen. Here it has been moved to the position shown for display purposes. Clicking the left and right mouse buttons in the bottom plot area at the appropriate positions results in the vertical lines appearing 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.

### C.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 PECS PLOT 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.

### C.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.

### C.7.6 Help menu commands

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

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

## C.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.