

Modeling, Design and Control of Power Converters

by

Aratrik Sarkar

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor Of Philosophy

Approved October 2021 by the
Graduate Supervisory Committee:

Armando A. Rodriguez, Chair
Jennie Si
Hans D. Mittelmann
Konstantinos Tsakalis

ARIZONA STATE UNIVERSITY

December 2021

ABSTRACT

This dissertation examines modeling, design and control challenges associated with two classes of power converters: a direct current-direct current (DC-DC) step-down (buck) regulator and a 3-phase ($3-\phi$) 4-wire direct current-alternating current (DC-AC) inverter. These are widely used for power transfer in a variety of industrial and personal applications. This motivates the precise quantification of conditions under which existing modeling and design methods yield satisfactory designs, and the study of alternatives when they don't. This dissertation describes a method utilizing Fourier components of the input square wave and the inductor-capacitor (LC) filter transfer function, which doesn't require the small ripple approximation. Then, trade-offs associated with the choice of the filter order are analyzed for integrated buck converters with a constraint on their chip area. Design specifications which would justify using a fourth or sixth order filter instead of the widely used second order one are examined. Next, sampled-data (SD) control of a buck converter is analyzed. Three methods for the digital controller design are studied: analog design followed by discretization, direct digital design of a discretized plant, and a "lifting" based method wherein the sampling time is incorporated in the design process by lifting the continuous-time design plant before doing the controller design. Specifically, controller performance is quantified by studying the induced- \mathcal{L}_2 norm of the closed loop system for a range of switching/sampling frequencies. In the final segment of this dissertation, the inner-outer control loop, employed in inverters with an inductor-capacitor-inductor (LCL) output filter, is studied. Closed loop sensitivities for the loop broken at the error and the control are examined, demonstrating that traditional methods only address these properties for one loop-breaking point. New controllers are then provided for improving both sets of properties.

ACKNOWLEDGEMENTS

I would like to thank my Ph.D. advisor, Professor *Armando A. Rodriguez* for his insight and guidance during my doctoral study. I have learned a great deal about approaching engineering problems and looking for trade-offs in design from him. The work in this thesis wouldn't have been possible without his constant supervision.

Besides my advisor, several people have helped me reach this milestone.

I am immensely grateful to my committee members: Drs. *J. Si*, *H. Mittelman*, and *K. Tsakalis*.

I would also like to especially thank *Brent Wallace*, with whom I had very illuminating conversations on a variety of topics addressed in this thesis.

Finally, I thank my family for their unconditional support.

TABLE OF CONTENTS

	Page
LIST OF TABLES	vi
LIST OF FIGURES	viii
CHAPTER	
1 INTRODUCTION	1
1.1 Motivation and Fundamental Questions Answered	2
1.2 Related Work and Literature Survey	3
1.3 Contributions	5
1.4 Organization of Dissertation	6
1.5 Summary	7
2 MODELING OF DC-DC CONVERTER	8
2.1 Overview	8
2.2 Basic Switching Operation of DC-DC Converters	9
2.3 Output Filters	10
2.4 Basic Parts of a DC-DC Converter	11
2.5 State-Space Modeling	12
2.5.1 Assumptions for the Average Model	13
2.6 Nonlinear Large Signal State-Space Averaged Model	14
2.7 Equilibrium State-Space Averaged Model	15
2.8 Solution of Equilibrium Averaged Model	16
2.9 Small-Signal (AC) Averaged LTI Model	17
2.10 Linearized Small-Signal Equations in Laplace Domain	20
2.11 Summary	22
3 PLANT DESIGN FOR DC-DC BUCK CONVERTER	23

CHAPTER	Page
3.1 Buck Converter with a Second Order LC Filter	23
3.2 State Space Matrices	24
3.3 Traditional Plant design for a Buck converter with LC filter	32
3.4 Limitations of Traditional Filter Design	35
3.5 Accuracy in Predicting % Voltage Ripple	35
3.6 Requirements for a Fourth Order LCLC Filter	39
3.7 State-Space Matrices with LCLC Filters	43
3.8 Summary	50
4 CONTROL SYSTEM DESIGN FOR BUCK CONVERTER	51
4.1 Mathematical Preliminaries	51
4.2 Need for Sampled-Data Control	52
4.2.1 Analog Controller Design	54
4.2.2 Digital Controller Design	57
4.3 Computation of the Induced- \mathcal{L}_2 Norm	67
4.4 Weighting Function	70
4.5 “Lifted” Plant (\mathcal{H}^∞ Discretization)	71
4.6 Trade-off	72
4.7 Summary	73
5 MODELING AND CONTROL OF DC-AC INVERTER WITH LCL FIL- TER	81
5.1 3- ϕ DC-AC Inverter Averaged Model	81
5.2 Traditional Inner-Outer Control with a PR Controller	83
5.3 Properties at the input and output	84
5.4 Inner-Outer Control with Lag Network	90

CHAPTER	Page
5.5 Summary	91
6 SUMMARY and FUTURE DIRECTIONS	93
REFERENCES	94
APPENDIX	104
A MATLAB CODE: LIFTING	104
B MATLAB CODE: BUCK CONVERTER RIPPLE	238
C MATLAB CODE: HIGHER ORDER FILTERS	251
D MATLAB CODE: INVERTER INNER-OUTER LOOP	258

LIST OF TABLES

Table	Page
3.1 Volt. Ripple and Error	35
3.2 Attenuation and Filter Order	42
3.3 Switching Frequency and Power Loss	42
4.1 Norms	72
4.2 Trade-off: Stability vs. Power Loss	75
5.1 Plant Poles and Zeros	83
5.2 Critical Control-Relevant Properties (in dB)	90
5.3 Critical Control-Relevant Properties	90

LIST OF FIGURES

Figure	Page
2.1 Switching DC-DC Converter	9
2.2 The Basic Structure of DC-DC converter	11
3.1 DC-DC Buck Converter with LC filter	23
3.2 (a) Transistor ON $q(t) = 1$; (b) Transistor OFF $q(t) = 0$	24
3.3 Limitations of Traditional Filter Design Equations	36
3.4 Error vs. Ripple	37
3.5 Model 1: Error vs. Ripple	38
3.6 Model 2: Error vs. Ripple	39
3.7 Frequency Response of Higher Order Output Filters	41
3.8 DC-DC Converter with LCLC-Filter	43
3.9 (a) Transistor ON $q(t) = 1$; (b) Transistor OFF $q(t) = 0$	43
4.1 Actual System Versus System Obtained using Typical Discretization .	53
4.2 Schematic for DC-DC Buck Converter Circuit with Compensation [5], [6]	54
4.3 Visualization of Weighted Mixed Sensitivity Analog Controller Design [89]	55
4.4 Nominal Open Loop Frequency Response	56
4.5 Weight on Sensitivity	57
4.6 Visualization of Weighted Mixed Sensitivity Sampled-Data Controller Design [89]	58
4.7 SD System [89]	69
4.8 $\ T_{wz}\ $ vs. $h = T_s$ [89]	70

Figure	Page
4.9 Closed Loop System [89]	71
4.10 $\ T_{wz}\ $ vs. $h = T_s$	72
4.11 Lifted Generalized Plant ($G_{eq,d}$)	73
4.12 Output (SD System)	74
4.13 Error (SD System)	75
4.14 Control (SD System)	76
4.15 Effect of Variation in $h (= T_s)$ on $\ T_{ry}\ _{\mathcal{L}_2}$	77
4.16 Effect of Variation in $h (= T_s)$ on $\ T_{ry}\ _{\mathcal{L}_2}$ (Zoomed In)	78
4.17 Error Time Response ($f_s = 4.5$ kHz, $T_s = h = 220 \mu s$)	79
4.18 Error Time Response ($f_s = 100$ kHz, $T_s = h = 10 \mu s$)	80
5.1 Grid-Tie Inverter (LCL Filter) Circuit	82
5.2 Inner-Outer Loop Control Structure	83
5.3 Root Locus of $H_z(s)$ (Eqn. 5.7)	87
5.4 Root Locus of $H_p(s)$ (Eqn. 5.12)	88
5.5 $ S_c _\infty, T_c _\infty, \frac{1}{2\sin\frac{PMc_3}{2}}$ v.s. k_i	89
5.6 Open Loop (Broken at the Input)	91
5.7 $S_e : r \rightarrow e$	92

INTRODUCTION

This thesis addresses controller and system design for two classes of power converters. Traditional methods are typically satisfactory for most designs. However, this thesis attempts to precisely quantify when certain established methods yield “satisfactory” results; and under what specifications, more advanced methods are required.

DC-DC Buck Converters. We study these power electronic circuits as they are widely used in a variety of applications, in addition to capturing many of the issues faced by other types of DC-DC converters. We study the extent to which approximations made while designing the filter hold and present more advanced Fourier analysis based methods when they don’t. For integrated buck converters, the chip area is an important design constraint. The necessity of higher order output filters is analyzed in this context. We also use buck converters as an example of a sampled-data (SD) system in order to quantify the stability using the closed loop system’s induced \mathcal{L}_2 norm, which captures the inter-sample behavior using a process known as lifting.

3- ϕ 4-wire DC-AC Inverters. These are essentially 1- ϕ circuits. We study a system with an LCL output filter. The 3rd order plant is often controlled using an inner-outer control structure. Controller designs found in the literature usually address control-relevant sensitivities for the loop broken at the error. This thesis addresses both the properties at error and control.

1.1 Motivation and Fundamental Questions Answered

The motivation for this research comes from the following questions:

1. How do we quantify acceptable performance of the plant and controller?
2. Under what conditions do the traditional design methods yield acceptable designs?
3. And what are more advanced techniques which can be used in cases when conventional methods fail to yield an acceptable design ?

We study the following specific questions in 3 different areas viz. Modeling, Plant Design and Control Design:

Modeling & Analysis

- (DC-DC converter) When can we get by with a second-order plant and traditional plant design, and **when do we need a higher order plant** ?

Plant Design

- When do the traditional design equations based on a *small ripple approximation* work for the LC filter and for what value of % voltage ripple do we need **more advanced Fourier spectrum based** techniques ?

Control Design

- (DC-DC converter) For what plant and controller specifications can we design an analog controller and discretize, and for what specifications do we need a **direct digital design or a “lifting” based direct sampled-data design** that takes inter-sample behavior into account?

- (DC-AC inverter) For what specifications is the traditional Inner-Outer Loop Control design for Inverters suitable, and **what are better inner and outer controller design methods** when they aren't suitable ?

1.2 Related Work and Literature Survey

This section includes some of the literature related to DC-DC buck converter and DC-AC inverter plant and controller design, and also relevant control literature for SD systems.

DC-DC Buck Converters. DC-DC switched mode power converters are widely used in modern electronic systems (mobile devices, computers, communication and medical equipment etc.) to interface energy sources to load requirements (or vice versa) in a lightweight, reliable and efficient way [1]. In order to filter out switching harmonics generated in these converters, an output passive filter is usually used. While a second-order LC filter is most commonly used [55]; in recent years fully integrated voltage regulators (FIVRs) have gained importance in multi-core systems [61]. These FIVRs have an area constraint limiting the size of the components (and hence the minimum cutoff frequency), which is why high switching frequencies in the range of hundreds of megahertz are typically used to achieve the desired output ripple. In order to reduce the switching frequency, and hence the switching loss in these converters, fourth order LCLC low pass filters have been proposed [60]. The design of these converters as well as studying trade-offs with respect to the traditional LC filter remains an area of active research. Typically an analog compensator is used, but digital controllers are becoming popular [2], [3] because of reduced design cycle time, ability to implement more complex control laws and ease of integration with

other digital subsystems. Traditional digital controller design methods utilized in the literature [99] do not capture the effect of inter-sample behavior [14]. For low sampling frequencies, a SD closed loop system may go unstable unless we do a direct SD design utilizing modern “lifting” techniques [89].

SD Control Based on Lifting. The effect of sampling frequency on closed-loop stability has been studied in other applications like quad-copters [109], but the effect of switching frequency on the stability of digital control systems for buck converters has not been studied in the context of the induced- \mathcal{L}_2 norm [110]. Recent work has not focused on the lifting-based methodology or investigated the trade-off between switching power loss and stability for buck converters [5, 6], [103, 104, 105, 106, 107]. This is largely because DC-DC converters usually have “large” switching frequencies, and traditional design methods are typically suitable for these switching frequencies. Hence, a thorough quantitative analysis of switching loss and stability (measured by the induced- \mathcal{L}_2 norm) has not been conducted before. In addition, modern hybrid switched capacitor buck converters with large duty ratios [111, 112], make it possible for buck converters to have smaller switching frequencies and mitigate switching loss. For low switching frequencies, traditional methods may not yield a stabilizing design; however, the modern lifting-based technique allows for minimization of switching frequency while preserving closed-loop stability [84, 87, 89, 90, 91].

DC-AC 3- ϕ 4-wire Inverter. 3- ϕ voltage source inverters (VSIs) are used to interface DPGSs like wind and solar to utility grid [16]. LCL filters are now widely used in DPGSs [18]. Resonance in LCL filter may lead to instability [20]. Active damping is usually preferred for stabilizing the inverter since adding resistance (passive damping) leads to additional power loss [21]. Little in literature on achieving

specifications at distinct loop breaking points. Essential for control system to have good properties at different loop breaking points (e.g. error and control) to be Robust to uncertainty and/or disturbances at these points [24]. Various active damping strategies like capacitor current feedback [28] have been tried, but in recent years strategies which rely only on sensed grid side current gained prominence since they don't involve any extra sensing [36].

1.3 Contributions

In this dissertation, performance specifications have been defined for systematically quantifying plant and controller design performance. This quantification is done to determine conditions under which existing methods give suitable performance. More sophisticated techniques are shown to yield acceptable designs when traditional methods fail. Precisely studied specifications for which higher-order output filters are required for buck converters. For the 2^{nd} order LC filter plant, % error between actual and predicted ripple have been computed for traditional design methods to show what values of % output voltage ripple lead to unacceptable amounts of error. For these cases, certain number of Fourier harmonics as well as the linear model of the filter are used to accurately predict the ripple. For purposes of digital controller design, 3 methods:

1. Analog design followed by discretization
2. Direct digital design
3. Lifting based controller design

are studied and their relative performance for a range of switching frequencies is studied by computing the induced- \mathcal{L}_2 norm of the closed-loop system.

Finally, control-relevant properties (sensitivities) are computed for the inverter with an inner-outer loop control structure. Properties at the error are found to be suitable for designs in the literature, but properties for the loop broken at the control are not. These are subsequently improved using novel controller structures.

1.4 Organization of Dissertation

In chapter 2, the modeling of DC-DC buck converters is presented in great detail. Since these capture many features of other power converters, we discuss the derivation of an averaged large and small signal model in detail. Chapter 3 discusses the plant design methods for buck converters and compares the traditional small ripple approximation based method to one utilizing Fourier components of the input square wave along with the state-space model of the LC filter. It also discusses higher-order alternatives to the 2nd order plant and presents specifications which necessitate these higher-order filters. Chapter 4 focuses on controller design for the buck converter. More specifically, we utilize the induced- \mathcal{L}_2 norm to determine controller performance for 3 types of digital controllers. A comparative analysis is presented to show that which all digital controllers demonstrate acceptable performance at high switching frequencies, some methods are better than others at lower switching frequencies. We may wish to reduce the frequency in order to lower switching power loss wherever permitted, hence this analysis demonstrates an important trade-off and need for some of these more advanced digital controller designs. Chapter 5 studies a different power converter, a 3- ϕ 4-wire DC-AC inverter with an LCL output filter. Inner-outer loop based control is studied in order to show that sensitivities for the feedback loop broken at the error are different from those for the loop broken at the control. A solution for achieving sensitivities less than 6 dB at both loop broken points is presented. Finally,

Chapter 6 summarizes the results of the dissertation and suggests possible directions for future research.

1.5 Summary

This chapter motivates the work done in this thesis on the basis of a thorough literature survey. It also presents the main contributions and presents the organization of the thesis.

MODELING OF DC-DC CONVERTER

2.1 Overview

DC-DC converters are more power efficient, versatile and substantially smaller and lighter than the traditional linear power supplies [46]. But the output regulation in switching mode DC-DC conversion requires more complex feedback control loop structure compared to linear regulator causing an increase in overall cost. These are extensively used as energy conversion components in myriad domestic, wearable and portable electronic devices, for example, personal computers, mobile phones, etc. DC-DC converters have also many essential industry applications like in accelerator technologies [4], aerospace [47] and automobile industries.

Very fast cyclical switching actions of semiconductor switches (transistors) used in SMPS produce distortion or ripples in the output voltage and current. Fast transitions of current and voltage induced by the high frequency switching cause Electromagnetic Interference (EMI) with other electronic components in devices, such as, computers or communication equipment [4]. Radiated electric fields are produced by the rapid voltage changes at the Inductor node, while fast changing Inductor current produces magnetic field [46, 48] Furthermore, some sensitive loads such as Integrated digital circuits require almost constant dc power supply with little ripple for proper functioning [4]. To fulfill such stringent industry standards regarding power quality, the output ripple needs to be suppressed substantially employing low-pass-filters (LPF) in the power stage of converters.

2.2 Basic Switching Operation of DC-DC Converters

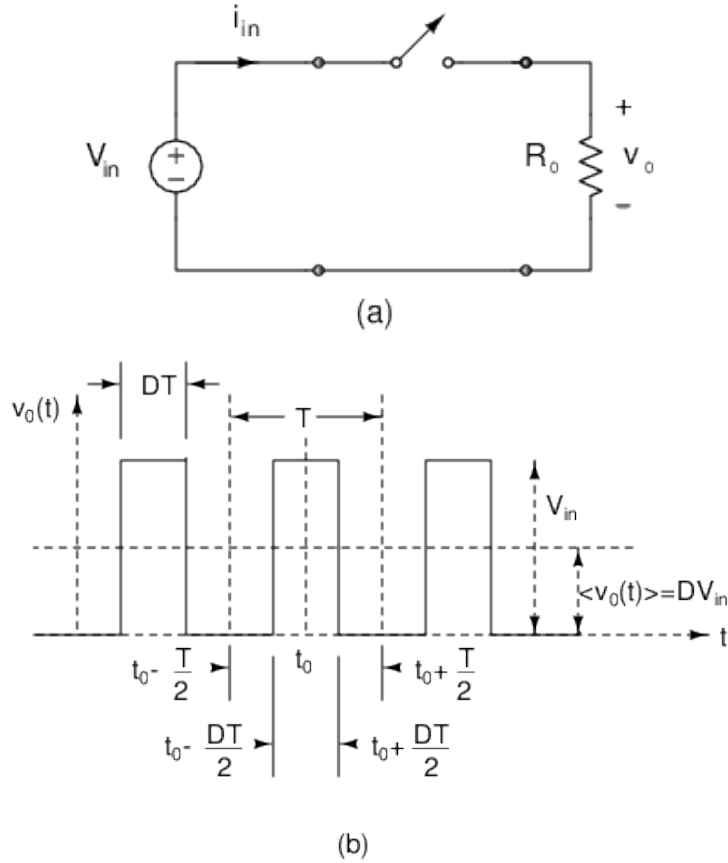


Figure 2.1: Switching DC-DC Converter

The simplest form of a switching DC-DC converter (without a Low Pass Filter) and its output is depicted in Fig. 2.1 [49]. An ideal switch is, by definition, a circuit element that can support both a non-zero average voltage and a non-zero average current without dissipating energy [4].

The average value of the output can be controlled by varying the ratio of the *ON* to *OFF* times of the switch. The switching action causes the instantaneous values of the input current and output voltage differ from their average values.

The signal produced by the periodic switching action at the input of a DC-DC

converter can be expressed as a Fourier series that is a sum of sine waves with frequencies that are integer multiples of the switching frequency (f_s) [4, 49, 50].

The switch opens and closes at a frequency f_s ($= 1/T$). The *duty ratio*, D , is defined as the ratio of the ON-time to the period T .

$$D = \frac{t_{ON}}{T} = \frac{t_{ON}}{t_{ON} + t_{OFF}} = f_s \cdot t_{ON} \quad (2.1)$$

$$t_{ON} = DT, t_{OFF} = (1 - D)T$$

The resulting load voltage v_0 is a chopped version of the input [4] – a series of pulses having an amplitude V_{in} , and an average, or DC value of v_0 : $\langle v_0 \rangle = DV_{in}$. But this DC value comes with a substantial amount of *ripple*. This ripple is present in the load voltage v_0 as well as in the source current I_{in} .

$$\begin{aligned} v_0(t) &= DV_{in} + \frac{2V_{in}}{\pi} \sum_{n=1}^{\infty} \frac{\sin(n\pi D)}{n} \cos(n\omega_s t - n\phi_0) \\ \omega_s &= \frac{2\pi}{T} = 2\pi f_s, \quad \phi_0 = \omega t_0. \end{aligned} \quad (2.2)$$

2.3 Output Filters

As expressed in eq. (2.2), presence of switching frequency harmonics along with the DC component produces the unwanted ripple or distortion in the output voltage [51]. This necessitates an output low-pass-filters(LPF) interface so as to produce the desired output terminal variables with very little ripple. Inductances (L) and capacitances (C) are the usual components of a filter circuit. Traditionally a second order inductor-capacitor (LC) filter is used for this purpose. In some applications, for example, accelerator technology, the high-stability converters must have a high

closed-loop band width so that it can react to errors very quickly. The closed-loop band width gets limited by the output filter and this requires that the cut-off frequency is to be made as high as possible. But the filter with high cut-off frequency cannot reduce the output ripple effectively (as explained later by eq. (3.43)). In such cases, higher-order (higher than 2^{nd}) filters with more components of L and C need to be employed [51].

2.4 Basic Parts of a DC-DC Converter

There are two main parts of a typical DC-DC converter [52]: the power stage (PS) and the control circuit (CC) as shown in Fig. 2.2.

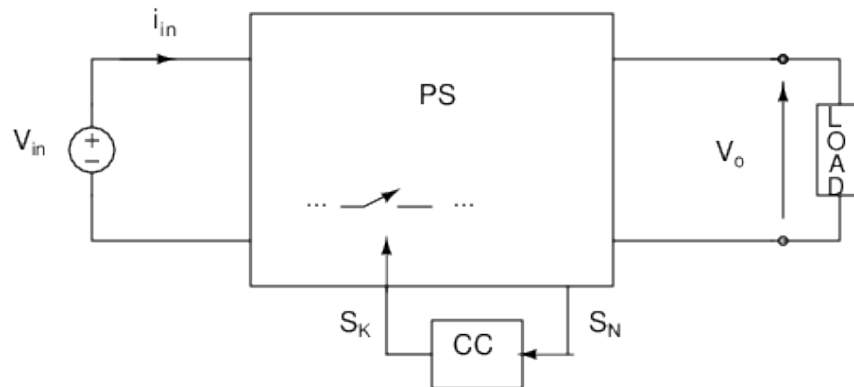


Figure 2.2: The Basic Structure of DC-DC converter

Depending on the power stage circuit topology, DC-DC converter can step down or up the DC input voltage or can be used both ways. The three most commonly used DC-DC converters are Buck (step down) or Boost (step up) and Buck-Boost (bidirectional) [47]. The power stage of these converters consists of semiconductor devices acting as switches and low loss components such as inductors, capacitors,

which are usually part of a low-pass-filter, which regulates the output by suppressing the switching frequency ripples.

The control circuit senses the state S_N of power stages, usually the output voltage, and generates the control signal S_K , which controls the state of the main or active switch referred to as “transistor”. The auxiliary switch in the power stage is referred to as “diode”.

There are two approaches to regulate the output of a DC-DC converter, namely, *voltage-mode* control and *current-mode* control, respectively [6],[12].

In this document we shall be dealing with the pulse-width modulated (PWM) voltage-mode control where the output voltage is compared with a reference signal to generate a control signal which drives the pulse-width modulator via a feedback loop.

2.5 State-Space Modeling

For a typical power converter circuit, the physical state variables are the inductor currents (i_L) and the capacitor voltages (v_C). The inductor voltage across an inductor L is $\frac{d(Li_L)}{dt}$; the voltage across a capacitor is v_C , the capacitor current is $\frac{d(Cv_C)}{dt}$. A switching function $q(t)$ may be defined to represent the effect of control inputs.

Let us assume, in general, the power converter switches between N circuit topologies during one switching cycle. If \mathbf{x} denotes the state vector and D_j be the fraction of the period T in which the circuit belongs to the j^{th} topology. It follows that

$$D_1 + D_2 + \cdots + D_j = 1 \tag{2.3}$$

The state equations of the converter for the first period can now be rewritten as:

$$\dot{\mathbf{x}}(t) = \begin{cases} A_1\mathbf{x}(t) + B_1\mathbf{u}(t), & 0 \leq t < D_1T \\ A_2\mathbf{x}(t) + B_2\mathbf{u}(t), & D_1T \leq t < (D_1 + D_2)T \\ \vdots & \vdots \\ A_N\mathbf{x}(t) + B_N\mathbf{u}(t), & (1 - D_N)T \leq t < T \end{cases} \quad (2.4)$$

$$\mathbf{y}(t) = \begin{cases} E_1\mathbf{x}(t) + F_1\mathbf{u}(t), & 0 \leq t < D_1T \\ E_2\mathbf{x}(t) + F_2\mathbf{u}(t), & D_1T \leq t < (D_1 + D_2)T \\ \vdots & \vdots \\ E_N\mathbf{x}(t) + F_N\mathbf{u}(t), & (1 - D_N)T \leq t < T \end{cases} \quad (2.5)$$

where $\mathbf{x}(t)$ is a state vector, $\mathbf{u}(t)$ and $\mathbf{y}(t)$ are input and output vectors respectively. A_j, B_j, E_j and F_j matrices depend on the circuit topology of a particular switching state j and the characteristics of its components.

2.5.1 Assumptions for the Average Model

In state-space averaging, the switching period T is considered as the averaging interval. The assumptions for the averaged model are the following [4]:

- The small ripple assumption:
for a given value of $T = \frac{1}{f_s}$, the ripple in the instantaneous value of a state variable $x(t)$ is small enough that it can be ignored and the variable can be approximated by its average value $\langle x(t) \rangle$ at time t .
- The slow variation assumption:
the average values of the variables don't vary substantially over an averaging

interval T , i.e. the averaged values vary much slower than one half of the switching frequency.

Well defined high-frequency switching converters operating in CCM generally satisfy these two assumptions.

2.6 Nonlinear Large Signal State-Space Averaged Model

A PWM converter, operating in continuous conduction mode (CCM), two circuit topologies corresponding to two subintervals during each switching period are possible.

During subinterval 1 or the ON-stage, let (A_1, B_1, E_1, F_1) denote the state space matrices for the power stage. The converter described by a linear circuit can be represented by the following state equations:

$$\begin{aligned}\frac{d\mathbf{x}(t)}{dt} &= A_1\mathbf{x}(t) + B_1\mathbf{u}(t) \\ \mathbf{y}(t) &= E_1\mathbf{x}(t) + F_1\mathbf{u}(t).\end{aligned}\tag{2.6}$$

Similarly, during subinterval 2, the state equations are:

$$\begin{aligned}\frac{d\mathbf{x}(t)}{dt} &= A_2\mathbf{x}(t) + B_2\mathbf{u}(t) \\ \mathbf{y}(t) &= E_2\mathbf{x}(t) + F_2\mathbf{u}(t).\end{aligned}\tag{2.7}$$

The durations of subintervals ON and OFF are related to switching period T or frequency $f_s(= 1/T)$ and duty ratio d by formulas:

$$d(t) = \frac{T_{ON}}{T}.\tag{2.8}$$

Using *classical averaging ideas* [53, 54] one obtains a *nonlinear large signal averaged model possessing the following structure* [53], pp 217-221:

$$\frac{d}{dt}\langle \mathbf{x}(t) \rangle = [d(t)A_1 + (1 - d(t))A_2]\langle \mathbf{x}(t) \rangle + [d(t)B_1 + (1 - d(t))B_2]\langle \mathbf{u}(t) \rangle \quad (2.9)$$

where

$$\langle \mathbf{z}(t) \rangle \stackrel{\text{def}}{=} \frac{1}{T} \int_t^{t+T} z(\tau) d\tau \quad (2.10)$$

denotes the moving time-average of the large signal quantity $\mathbf{z}(\tau)$.

Given this, $\langle \mathbf{u}(t) \rangle$, $\langle \mathbf{x}(t) \rangle$ and $\langle \mathbf{y}(t) \rangle$ denote moving time-averages of the power stages input, state, and output variables. This model may be viewed as LTV, LPV, or nonlinear in d .

2.7 Equilibrium State-Space Averaged Model

The analysis described here is based on [53]. Provided that the natural frequencies of the converter, as well as the frequencies of variations of the converter inputs, are much slower than the switching frequency, then the State-space averaged model that describes the converter in equilibrium is

$$\begin{aligned} \mathbf{0} &= \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{U} \\ \mathbf{Y} &= \mathbf{E}\mathbf{X} + \mathbf{F}\mathbf{U} \end{aligned} \quad (2.11)$$

where the averaged matrices are

$$\begin{aligned}A &= DA_1 + (1 - D)A_2 \\B &= DB_1 + (1 - D)B_2 \\E &= DE_1 + (1 - D)E_2 \\F &= DF_1 + (1 - D)F_2\end{aligned}\tag{2.12}$$

and the equilibrium DC components are

\mathbf{X} = equilibrium State vector

\mathbf{U} = equilibrium input vector

\mathbf{Y} = equilibrium output vector

D = equilibrium duty ratio

2.8 Solution of Equilibrium Averaged Model

Eq. (2.11) may be solved as

$$\begin{aligned}\mathbf{X} &= -A^{-1}B\mathbf{U} \\ \mathbf{Y} &= (-EA^{-1}B + F)\mathbf{U}\end{aligned}\tag{2.13}$$

2.9 Small-Signal (AC) Averaged LTI Model

Let us consider the following large signal average model.

$$\begin{aligned}\langle \mathbf{x}(t) \rangle &= \mathbf{X} + \hat{\mathbf{x}}(t) \\ \langle \mathbf{u}(t) \rangle &= \mathbf{U} + \hat{\mathbf{u}}(t) \\ \langle \mathbf{y}(t) \rangle &= \mathbf{Y} + \hat{\mathbf{y}}(t) \\ \langle d(t) \rangle &= D + \hat{d}(t)\end{aligned}\tag{2.14}$$

where $\mathbf{X}, \mathbf{U}, \mathbf{Y}$ and D denote equilibrium moving averages for the state input, and output respectively and $\hat{\mathbf{x}}(t), \hat{\mathbf{u}}(t), \hat{\mathbf{y}}(t), \hat{d}(t)$ are the corresponding small-signal (AC) perturbations

$$\begin{aligned}\|\mathbf{U}\| &\gg \|\hat{\mathbf{u}}(t)\| \\ \|\mathbf{X}\| &\gg \|\hat{\mathbf{x}}(t)\| \\ \|\mathbf{Y}\| &\gg \|\hat{\mathbf{y}}(t)\| \\ \|D\| &\gg \|\hat{d}(t)\|.\end{aligned}\tag{2.15}$$

Substituting the above equations in eq. (16) and using eq. (18) and eq. (19), we get

$$\begin{aligned}
\frac{d}{dt}(\mathbf{X} + \hat{\mathbf{x}}(t)) &= \left[(D + \hat{d}(t)) A_1 + (1 - (D + \hat{d}(t))) A_2 \right] \cdot (\mathbf{X} + \hat{\mathbf{x}}(t)) \\
&+ \left[(D + \hat{d}(t)) B_1 + (1 - (D + \hat{d}(t))) B_2 \right] \cdot (\mathbf{U} + \hat{\mathbf{u}}(t)) \\
\mathbf{Y} + \hat{\mathbf{y}}(t) &= \left[(D + \hat{d}(t)) E_1 + (1 - (D + \hat{d}(t))) E_2 \right] \cdot (\mathbf{X} + \hat{\mathbf{x}}(t)) \\
&+ \left[(D + \hat{d}(t)) F_1 + (1 - (D + \hat{d}(t))) F_2 \right] \cdot (\mathbf{U} + \hat{\mathbf{u}}(t))
\end{aligned} \tag{2.16}$$

Setting $\frac{d}{dt}\mathbf{x} = 0$ (for equilibrium) and using eq. (18) and eq. (19), eq. (23) can be reorganized as

$$\begin{aligned}
\frac{d}{dt}\hat{\mathbf{x}}(t) &= [DA_1 + (1 - D)A_2]\hat{\mathbf{x}}(t) + [A_1 - A_2]\hat{\mathbf{x}}(t)\hat{d}(t) \\
&+ [DA_1 + (1 - D)A_2]X + [A_1 - A_2]X\hat{d}(t) \\
&+ [DB_1 + (1 - D)B_2]\hat{\mathbf{u}}(t) + [B_1 - B_2]\hat{\mathbf{u}}(t)\hat{d}(t) \\
&+ [DB_1 + (1 - D)B_2]U + [B_1 - B_2]U\hat{d}(t)
\end{aligned} \tag{2.17}$$

Similarly,

$$\begin{aligned}
\mathbf{Y} + \hat{\mathbf{y}}(t) &= [DE_1 + (1 - D)E_2]\hat{\mathbf{x}}(t) + [E_1 - E_2]\hat{\mathbf{x}}(t)\hat{d}(t) \\
&+ [DE_1 + (1 - D)E_2]X + [E_1 - E_2]X\hat{d}(t) \\
&+ [DF_1 + (1 - D)F_2]\hat{\mathbf{u}}(t) + [F_1 - F_2]\hat{\mathbf{u}}(t)\hat{d}(t) \\
&+ [DF_1 + (1 - D)F_2]U + [F_1 - F_2]U\hat{d}(t)
\end{aligned} \tag{2.18}$$

Using (2.17),

$$\begin{aligned} \frac{d}{dt}\hat{\mathbf{x}}(t) &= A\hat{\mathbf{x}}(t) + [A_1 - A_2]\hat{\mathbf{x}}(t)\hat{d}(t) + A\mathbf{X} + [A_1 - A_2]\mathbf{X}\hat{d}(t) \\ &\quad + B\hat{\mathbf{u}}(t) + [B_1 - B_2]\hat{\mathbf{u}}(t)\hat{d}(t) + B\mathbf{U} + [B_1 - B_2]\mathbf{U}\hat{d}(t) \end{aligned} \quad (2.19)$$

Similarly,

$$\begin{aligned} \mathbf{Y} + \hat{\mathbf{y}}(t) &= E\hat{\mathbf{x}}(t) + [E_1 - E_2]\hat{\mathbf{x}}(t)\hat{d}(t) + E\mathbf{X} + [E_1 - E_2]\mathbf{X}\hat{d}(t) \\ &\quad + F\hat{\mathbf{u}}(t) + [F_1 - F_2]\hat{\mathbf{u}}(t)\hat{d}(t) + F\mathbf{U} + [F_1 - F_2]\mathbf{U}\hat{d}(t) \end{aligned} \quad (2.20)$$

Further simplification gives

$$\begin{aligned} \frac{d}{dt}\hat{\mathbf{x}}(t) &= A\mathbf{X} + B\mathbf{U} + A\hat{\mathbf{x}}(t) + B\hat{\mathbf{u}}(t) + \{(A_1 - A_2)\mathbf{X} + (B_1 - B_2)\mathbf{U}\}\hat{d}(t) \\ &\quad + [A_1 - A_2]\hat{\mathbf{x}}(t)\hat{d}(t) + [B_1 - B_2]\hat{\mathbf{u}}(t)\hat{d}(t) \end{aligned} \quad (2.21)$$

$$\begin{aligned} \mathbf{Y} + \hat{\mathbf{y}}(t) &= E\mathbf{X} + F\mathbf{U} + E\hat{\mathbf{x}}(t) + F\hat{\mathbf{u}}(t) + \{(E_1 - E_2)\mathbf{X} + (F_1 - F_2)\mathbf{U}\}\hat{d}(t) \\ &\quad + [E_1 - E_2]\hat{\mathbf{x}}(t)\hat{d}(t) + [F_1 - F_2]\hat{\mathbf{u}}(t)\hat{d}(t) \end{aligned} \quad (2.22)$$

When the small-signal assumption is satisfied, the second-order (non-linear) terms are very small. So, *linearizing around the converter equilibrium point*, and using eq. (2.21)

$$\frac{d\hat{\mathbf{x}}(t)}{dt} = A\hat{\mathbf{x}}(t) + B\hat{\mathbf{u}}(t) + \{(A_1 - A_2)\mathbf{X} + (B_1 - B_2)\mathbf{U}\}\hat{d}(t) \quad (2.23)$$

$$\hat{\mathbf{y}}(t) = E\hat{\mathbf{x}}(t) + F\hat{\mathbf{u}}(t) + \{(E_1 - E_2)\mathbf{X} + (F_1 - F_2)\mathbf{Y}\}\hat{d}(t) \quad (2.24)$$

Eq. (2.23) and eq. (2.24) are the derived result which describes the *linearized small-signal state equations*.

2.10 Linearized Small-Signal Equations in Laplace Domain

The linearized small signal equations in time domain, eq. (2.23) and eq. (2.24), are further reorganized and written as

$$\begin{aligned}\frac{d\hat{\mathbf{x}}(t)}{dt} &= A\hat{\mathbf{x}}(t) + B\hat{\mathbf{u}}(t) + M\hat{d}(t) \\ \hat{\mathbf{y}}(t) &= E\hat{\mathbf{x}}(t) + F\hat{\mathbf{u}}(t) + G\hat{d}(t)\end{aligned}\tag{2.25}$$

where

$$\begin{aligned}M &= (A_1 - A_2)\mathbf{X} + (B_1 - B_2)\mathbf{U} \\ G &= (E_1 - E_2)\mathbf{X} + (F_1 - F_2)\mathbf{U}\end{aligned}\tag{2.26}$$

Laplace transformation of eq. (2.23) and use of eq. (2.24) give

$$s\hat{\mathbf{x}}(s) = A\hat{\mathbf{x}}(s) + B\hat{\mathbf{u}}(s) + M\hat{d}(s)\tag{2.27}$$

$$\hat{\mathbf{y}}(s) = E\hat{\mathbf{x}}(s) + F\hat{\mathbf{u}}(s) + G\hat{d}(s)\tag{2.28}$$

This implies,

$$\hat{\mathbf{x}}(s) = [sI - A]^{-1}B\hat{\mathbf{u}}(s) + [sI - A]^{-1}M\hat{d}(s) \quad (2.29)$$

$$\begin{aligned} \hat{\mathbf{y}}(s) &= E \left\{ [sI - A]^{-1}B\hat{\mathbf{u}}(s) + [sI - A]^{-1}M\hat{d}(s) \right\} \\ &\quad + F\hat{\mathbf{u}}(s) + G\hat{d}(s) \end{aligned} \quad (2.30)$$

$$\Rightarrow \hat{\mathbf{y}}(s) = \{E[sI - A]^{-1}B + F\} \hat{\mathbf{u}}(s) + \{E[sI - A]^{-1}M + G\} \hat{d}(s) \quad (2.31)$$

When $\hat{d}(s) = 0$,

$$\hat{\mathbf{y}}(s) = \{E[sI - A]^{-1}B + F\} \hat{\mathbf{u}}(s) \quad (2.32)$$

Input to output transfer function of the system or the Plant Transfer-function

$$Tr_{Plant} = \left. \frac{\hat{\mathbf{y}}(s)}{\hat{\mathbf{u}}(s)} \right|_{\hat{d}(s)=0} = E[sI - A]^{-1}B + F. \quad (2.33)$$

If the perturbations in the inputs are zero, that is, $\hat{\mathbf{u}}(t) = 0$, then *the control to output transfer function* is given by

$$\left. \frac{\hat{\mathbf{y}}(s)}{\hat{d}(s)} \right|_{\hat{\mathbf{u}}(s)=0} = E[sI - A]^{-1}M + G \quad (2.34)$$

In the next section, we shall obtain *linearized small-signal state equations* for a Buck converter with LC filter and obtain the *plant and control-to-output transfer functions*.

2.11 Summary

This chapter presented the modeling of the DC-DC buck converters and studied various stages of modeling namely the circuit model, state-space nonlinear averaged model, equilibrium state-space averaged model and the final linearized small-signal (AC) averaged LTI model which will be used for control design.

PLANT DESIGN FOR DC-DC BUCK CONVERTER

3.1 Buck Converter with a Second Order LC Filter

Fig. 5 shows a circuit diagram of a Buck converter with LC filter. The transistor operates at a fixed switching frequency with period T . It is assumed to operate in the continuous current conduction mode (CCM). Corresponding to two subintervals, ON-time and OFF-time of a switching cycle, there are two equivalent circuit configurations (see Fig. 5). The inductor current i_L and the capacitor voltage v_C are the two state variables $x_1(t)$ and $x_2(t)$ respectively for the converter with the initial condition $\mathbf{x}(0) = \mathbf{x}_0$. R_L and R_c are the parasitic resistances of the inductor and the capacitor respectively.

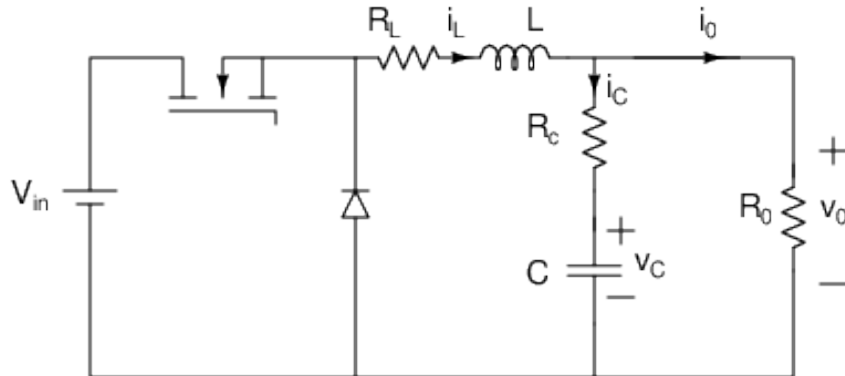


Figure 3.1: DC-DC Buck Converter with LC filter

The PWM process generating a rectangular pulsed line voltage $u(t)$ at the input is defined by:

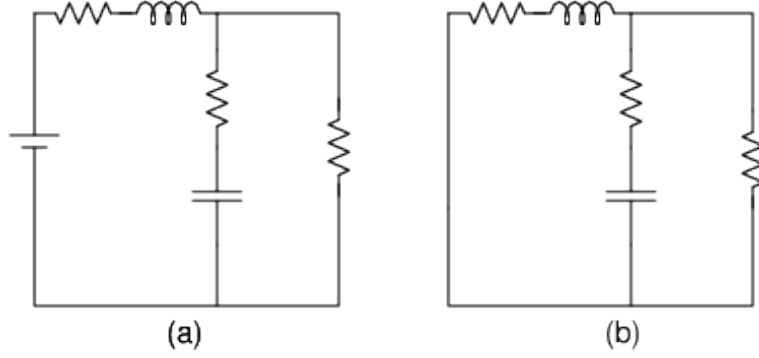


Figure 3.2: (a) Transistor ON $q(t) = 1$; (b) Transistor OFF $q(t) = 0$.

Transistor ON ($q(t) = 1$) for $nT \leq t < (n + D(t))T$, $u(t) = V_{in}$

Transistor OFF ($q(t) = 0$) elsewhere, $u(t) = 0$,

$D(t) \in \{0, 1\}$ denotes the duty cycle.

3.2 State Space Matrices

KVL and KCL relationships corresponding to the low pass filter (LPF) of Fig. 3.1 yield the following differential equations in terms of the state variables i_L and v_C [49].

$$L \frac{di_L(t)}{dt} = q(t)V_{in} - i_L R_L - v_0 \quad (3.1)$$

$$C \frac{dv_C(t)}{dt} = i_L - \frac{v_0}{R_0} \quad (3.2)$$

$$\text{Also, } v_C = v_0 - i_C R_C \quad (3.3)$$

To calculate v_0 in terms of i_L, v_C :

$$\begin{aligned}
v_0 &= v_C + i_C R_C \\
i_C &= i_L - i_0 = i_L - \frac{v_0}{R_0} \\
v_0 &= i_L R_0 - i_C R_0 \\
&= i_L R_0 - \left[\frac{v_0 - v_C}{R_C} \right] R_0 \\
\Rightarrow v_0 R_C &= i_L R_0 R_C - v_0 R_0 + v_C R_0 \\
\Rightarrow v_0(t) &= y = i_L \left[\frac{R_0 R_C}{R_0 + R_C} \right] + v_C \left[\frac{R_0}{R_0 + R_C} \right] \tag{3.4}
\end{aligned}$$

Eliminating v_0 from eq. (3.2) and eq. (3.3) and writing in the matrix form:

$$\begin{bmatrix} \frac{di_L(t)}{dt} \\ \frac{dv_C(t)}{dt} \end{bmatrix} = \begin{bmatrix} \frac{-R_L - kR_C}{L} & \frac{-k}{L} \\ \frac{k}{C} & \frac{-k}{R_0 C} \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} \frac{1}{L} u(t) \\ 0 \end{bmatrix} \tag{3.5}$$

where k is the divider constant, $k = \frac{R_0}{R_0 + R_C}$ and $u(t) = q(t)V_{in}$.

We get two sets of state space equations as:

Transistor – ON ($q(t) = 1$):

$$\begin{aligned}
\dot{\mathbf{x}}(t) &= A_1 \mathbf{x}(t) + B_1 \mathbf{u}(t) \\
\mathbf{y}(t) &= E_1 \mathbf{x}(t) + F_1 \mathbf{u}(t) \tag{3.6}
\end{aligned}$$

and Transistor – OFF ($q(t) = 0$):

$$\begin{aligned}\dot{\mathbf{x}}(t) &= A_2\mathbf{x}(t) + B_2\mathbf{u}(t) \\ \mathbf{y}(t) &= E_2\mathbf{x}(t) + F_2\mathbf{u}(t)\end{aligned}$$

where

$$\begin{aligned}\mathbf{x}(t) &= \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} i_L(t) \\ v_C(t) \end{bmatrix} \\ \mathbf{u}(t) &= \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} V_{in} \\ 0 \end{bmatrix}\end{aligned}\tag{3.7}$$

and

$$A_1 = A_2 = \begin{bmatrix} -\frac{1}{L}(R_L + kR_C) & -\frac{k}{L} \\ \frac{k}{C} & -\frac{k}{R_0C} \end{bmatrix}\tag{3.8}$$

$$B_1 = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.\tag{3.9}$$

Eq. (3.4) gives

$$\begin{aligned}\mathbf{y}(t) &= y = v_0(t) = i_L(t)kR_C + v_C(t)k \\ &= \begin{bmatrix} kR_C & k \end{bmatrix} \begin{bmatrix} i_L(t) \\ v_C(t) \end{bmatrix} + 0\end{aligned}\tag{3.10}$$

$$\Rightarrow E_1 = E_2 = \begin{bmatrix} kR_C & k \end{bmatrix} \quad (3.11)$$

$$\text{and } F_1 = F_2 = \begin{bmatrix} 0 & 0 \end{bmatrix} \quad (3.12)$$

Now that we know the two sets of matrices (A_1, B_1, E_1, F_1) and (A_2, B_2, E_2, F_2) for the Buck converter with LC-filter considering parasitic resistances of the inductance and the capacitance, we can use eq. (2.12), (2.25) and (2.26) to generate the small signal model.

Let us consider the small-signal perturbations in $i_L(t)$, $v_C(t)$ and $v_0(t)$ as

$$\begin{aligned} \langle i_L(t) \rangle &= I_L + \hat{i}_L(t) \\ \langle v_C(t) \rangle &= V_C + \hat{v}_C(t) \\ \langle v_0(t) \rangle &= V_0 + \hat{v}_0(t) \\ V_{in} &\Rightarrow V_{in} + \hat{v}_{in}(t) \end{aligned} \quad (3.13)$$

$$\begin{aligned} \frac{d\hat{\mathbf{x}}(t)}{dt} &= A\hat{\mathbf{x}}(t) + B\hat{\mathbf{u}}(t) + M\hat{d}(t) \\ \hat{\mathbf{y}}(t) &= E\hat{\mathbf{x}}(t) \end{aligned} \quad (3.14)$$

where

$$\hat{\mathbf{x}}(t) = \begin{bmatrix} \hat{i}_L(t) \\ \hat{v}_C(t) \end{bmatrix}, \quad \hat{\mathbf{y}}(t) = \hat{v}_0(t), \quad \hat{\mathbf{u}}(t) = \hat{v}_{in}(t), \quad (3.15)$$

$$A = \begin{bmatrix} -\frac{1}{L}(R_L + kR_C) & -\frac{k}{L} \\ \frac{k}{C} & -\frac{k}{R_0C} \end{bmatrix} \quad (3.16)$$

$$B = \begin{bmatrix} \frac{D}{L} \\ 0 \end{bmatrix} \quad M = \begin{bmatrix} \frac{V_{in}}{L} \\ 0 \end{bmatrix}. \quad (3.17)$$

$$E = \begin{bmatrix} kR_C & k \end{bmatrix}, \quad F = 0 \quad (3.18)$$

where

$$k = \frac{R_0}{R_0 + R_C} \quad (3.19)$$

The detailed averaged small signal LTI model is as follows:

$$\begin{aligned} \begin{bmatrix} \hat{i}_L(t) \\ \hat{v}_C(t) \end{bmatrix} &= \begin{bmatrix} -\frac{1}{L}(R_L + kR_C) & -\frac{k}{L} \\ \frac{k}{C} & -\frac{k}{R_0C} \end{bmatrix} \begin{bmatrix} \hat{i}_L(t) \\ \hat{v}_C(t) \end{bmatrix} \\ &+ \begin{bmatrix} \frac{D}{L} \\ 0 \end{bmatrix} \hat{v}_{in}(t) + \begin{bmatrix} \frac{V_{in}}{L} \\ 0 \end{bmatrix} \hat{d}(t) \end{aligned} \quad (3.20)$$

$$\hat{y}(t) = \hat{v}_0(t) = \begin{bmatrix} kR_C & k \end{bmatrix} \begin{bmatrix} \hat{i}_L(t) \\ \hat{v}_C(t) \end{bmatrix} \quad (3.21)$$

It is worth mentioning at this juncture that we have not considered any fluctuation in the load current as this is minimally zero. The PWM gain has also not been included separately.

Using eq. (2.31) we get the Laplace transformed small-signal output as

$$\hat{y}(s) = \hat{v}_0(s) = \{E[sI - A]^{-1}B\} \hat{v}_{in}(s) + \{E[sI - A]^{-1}M\} \hat{d}(s) \quad (3.22)$$

The converter transfer functions are now derived as: Plant transfer function, Tr_{Plant}

$$Tr_{Plant}(s) = \left. \frac{\hat{v}_0(s)}{\hat{v}_{in}(s)} \right|_{\hat{d}(s)=0} = E[sI - A]^{-1}B \quad (3.23)$$

control-to-output transfer function is given by

$$Tr_{control-to-output} = \left. \frac{\hat{v}_0(s)}{\hat{d}(s)} \right|_{\hat{v}_{in}(s)=0} = E[sI - A]^{-1}M \quad (3.24)$$

Here $\hat{d}(s)$ is the control parameter.

$$\begin{aligned}
Tr_{Plant} &= \begin{bmatrix} kR_C & k \end{bmatrix} \begin{bmatrix} \left(s + \frac{1}{L}(R_L + kR_C)\right) & \frac{k}{L} \\ -\frac{k}{C} & \left(s + \frac{k}{R_0C}\right) \end{bmatrix}^{-1} \begin{bmatrix} \frac{D}{L} \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} kR_C & k \end{bmatrix} \left\{ \frac{1}{\alpha} \right\} \begin{bmatrix} \left(s + \frac{k}{R_0C}\right) & -\frac{k}{L} \\ \frac{k}{C} & \left(s + \frac{1}{L}(R_L + kR_C)\right) \end{bmatrix} \begin{bmatrix} \frac{D}{L} \\ 0 \end{bmatrix} \quad (3.25) \\
&= \begin{bmatrix} kR_C & k \end{bmatrix} \left\{ \frac{1}{\alpha} \right\} \begin{bmatrix} \frac{D}{L} \left(s + \frac{k}{R_0C}\right) \\ \frac{Dk}{LC} \end{bmatrix} \\
&= \begin{bmatrix} R_C & 1 \end{bmatrix} \left\{ \frac{Dk}{\alpha} \right\} \begin{bmatrix} \frac{1}{L} \left(s + \frac{k}{R_0C}\right) \\ \frac{k}{LC} \end{bmatrix} \\
&= \left\{ D \left(\frac{R_0}{R_0 + R_C} \right) \frac{1}{\alpha} \right\} \left[\frac{R_C}{L} \left(s + \frac{k}{R_0C} \right) + \frac{k}{LC} \right] \\
&= \left\{ D \left(\frac{R_0}{R_0 + R_C} \right) \frac{1}{\alpha} \right\} \left[\frac{R_C}{L} \left(s + \frac{1}{(R_0 + R_C)C} \right) + \frac{R_0}{(R_0 + R_C)LC} \right] \\
&= \left\{ D \left(\frac{R_0}{R_0 + R_C} \right) \frac{1}{\alpha} \frac{1}{LC} \right\} \left[CR_Cs + \frac{R_C}{(R_0 + R_C)} + \frac{R_0}{(R_0 + R_C)} \right] \quad (3.26)
\end{aligned}$$

where

$$\begin{aligned}
\alpha &= \left\{ \left(s + \frac{1}{L}(R_L + kR_C) \right) \left(s + \frac{k}{R_0 C} \right) + \frac{k^2}{LC} \right\} \\
&= s^2 + \left(\frac{k}{R_0 C} + \frac{1}{L}(R_L + kR_C) \right) s + \frac{k}{R_0 C L}(R_L + kR_C) + \frac{k^2}{LC} \\
&= s^2 + \left(\frac{1}{(R_0 + R_C)C} + \frac{1}{L} \left(R_L + \frac{R_0 R_C}{R_0 + R_C} \right) \right) s \\
&\quad + \left(\frac{1}{R_0 + R_C} \right) \left(R_L + \frac{R_0 R_C}{R_0 + R_C} + \frac{R_0}{R_0 + R_C} \right) \frac{1}{LC} \\
&= s^2 + \left(\frac{1}{(R_0 + R_C)C} + \frac{1}{L} \left(R_L + \frac{R_0 R_C}{R_0 + R_C} \right) \right) s \\
&\quad + \left(\frac{R_0}{R_0 + R_C} \right) \left(\frac{R_L}{R_0} + \frac{R_C}{R_0 + R_C} + \frac{R_0}{R_0 + R_C} \right) \frac{1}{LC} \\
&= s^2 + \left(\frac{1}{(R_0 + R_C)C} + C \left(R_L + \frac{R_0 R_C}{R_0 + R_C} \right) \right) s \\
&\quad + \left(\frac{R_0}{R_0 + R_C} \right) \left(\frac{R_L}{R_0} + \frac{R_C}{R_0 + R_C} + \frac{R_0}{R_0 + R_C} \right) \frac{1}{LC} \\
&= \frac{1}{LC} \left\{ LCs^2 + \left(\frac{L}{(R_0 + R_C)} + C \left(R_L + \frac{R_0 R_C}{R_0 + R_C} \right) \right) s + \left(\frac{R_L + R_0}{R_0 + R_C} \right) \right\}
\end{aligned} \tag{3.27}$$

We thus obtain,

$$Tr_{Plant} = D \left(\frac{R_0}{R_0 + R_C} \right) \left[\frac{CR_C s + 1}{LCs^2 + \left(\frac{L}{R_0 + R_C} + C \left(R_L + \frac{R_0 R_C}{R_0 + R_C} \right) \right) s + \left(\frac{R_L + R_0}{R_0 + R_C} \right)} \right] \tag{3.28}$$

In the special case: $R_L = R_C = 0$,

$$Tr_{Plant}(s) = \frac{D}{LCs^2 + \frac{L}{R_0}s + 1} \tag{3.29}$$

3.3 Traditional Plant design for a Buck converter with LC filter

Traditionally, the plant design for a PWM Buck converter with LC filter (in CCM) is based on the following assumptions [50]:

- 1 The power MOSFET and the diode are ideal switches.
- 2 The transistor output capacitance, the diode capacitance, and the lead inductances are zero, and thus switching losses are neglected.
- 3 Passive components are linear time-invariant, and frequency independent.
- 4 The output impedance of the input voltage source v_{in} is zero for both DC and AC components.
- 5 The converter is operating in steady state.
- 6 The switching period $T = \frac{1}{f_s}$ is much shorter than the time constants of reactive components.
- 7 **Small Ripple Assumption:** we assume that the output voltage ripple ΔV_o is negligible, such that $V_O = V_C$ is a near ideal DC voltage.
- 8 **No Restrictions on Component Size:** we assume that in order to meet a desired % ripple specification for V_O , we do not have any restrictions on how large L and C can be and that the switching frequency choice is primarily dictated by power consumption in the MOSFET switch.

Case 1. $R_L = R_C = 0$: For easy analysis in the steady state, it is assumed that the capacitor C is large enough (that is with low impedance) so that the output ripple is negligible [59]. This is the small-ripple assumption.

Over the one period T of a switching cycle, average value of the Inductor current I_L is equal to the average output current [50]

$$I_L = I_0 \quad (3.30)$$

The inductor current starts at the initial value

$$I_L(0) = I_{L,min}$$

and changes to a peak value $I_{L,max}$ at the end of the switch-closure period (DT).

$$I_{L,max} = I_L(DT) \quad (3.31)$$

$$I_{L,max} - I_{L,min} = \frac{(V_{in} - V_0)}{L} \cdot DT \quad (3.32)$$

$$I_{L,min} - I_{L,max} = -\frac{V_0}{L} \cdot (1 - D)T \quad (3.33)$$

$$\Delta I_L(\text{peak-peak}) = |I_{L,max} - I_{L,min}| = |I_{L,min} - I_{L,max}| \quad (3.34)$$

$$\Rightarrow \boxed{\Delta I_L(\text{peak-peak}) = \frac{V_{in} - V_0}{L} \cdot DT = \frac{V_0}{L} \cdot (1 - D)T} \quad (3.35)$$

$$V_0 = DV_{in} \quad (3.36)$$

$$I_{L,max} = DV_{in} \left[\frac{1}{R_0} + \frac{(1 - D)}{2L} \cdot T \right] \quad (3.37)$$

$$I_{L,min} = DV_{in} \left[\frac{1}{R_0} - \frac{(1 - D)}{2L} \cdot T \right] \quad (3.38)$$

- Minimum inductance that results in a continuous current conduction (CCM) in

the inductor is obtained by putting $I_{L,min} = 0$ in eq. (3.38),

$$L_{min} = \frac{(1-D)}{2}TR_0 \quad (3.39)$$

Voltage ripple ΔV_0 is given by the formula:

$$\Delta V_0(\text{peak-peak}) = \Delta V_0 = \left(\frac{1}{8Cf_s} \right) \Delta I_L(\text{peak-peak}) \quad (3.40)$$

$$\Rightarrow \boxed{\Delta V_0 = \left(\frac{1}{8Cf_s} \right) \frac{V_0}{L} \cdot (1-D)T} \quad (3.41)$$

$$\begin{aligned} &= \left(\frac{1}{8Cf_s^2} \right) \frac{V_0}{L} \cdot (1-D)T \\ \Rightarrow \frac{\Delta V_0}{V_0} &= \frac{(1-D)}{8} \left(\frac{1}{\sqrt{LC}} \right)^2 \left(\frac{1}{f_s^2} \right) \\ \Rightarrow \frac{\Delta V_0}{V_0} &= \frac{(1-D)}{8} (2\pi f_c)^2 \left(\frac{1}{f_s^2} \right) \end{aligned} \quad (3.42)$$

where $f_c = \frac{1}{2\pi\sqrt{LC}}$, is corner or cut-off frequency

$$\Rightarrow \boxed{\frac{\Delta V_0}{V_0} = \frac{\pi^2}{2}(1-D) \left(\frac{f_c}{f_s} \right)^2} \quad (3.43)$$

Eq. (3.43) shows that the voltage ripple can be minimized by selecting a corner frequency f_c of the low-pass filter at the output such that $f_c \ll f_s$. The percentage ripple in the output voltage is usually specified to be less than 1% [59].

Volt. Ripple	Error
$\leq 7.1173\%$	$\leq 10\%$
$\leq 3.0217\%$	$\leq 1\%$
$\leq 1.7344\%$	$\leq 0.1\%$

Table 3.1: Volt. Ripple and Error

3.4 Limitations of Traditional Filter Design

Fig. 3.3 shows that keeping the switching frequency fixed, if we reduce the capacitance, thus increasing the output voltage ripple; then the error in predicting the ripple also increases. This is a quantification of the small ripple assumption. The equations tested are eq. (3.41) and eq. (3.43). Tab. 3.1 shows that if the output voltage ripple is in the range of 0 to 1%, then the error is less than 0.1%.

3.5 Accuracy in Predicting % Voltage Ripple

Small Ripple Approximation E.g. voltage ripple $> 3\%$ will result in a prediction error $> 1\%$

New Model for Predicting Ripple

- We approximate the input square wave by n harmonics of the Fourier Spectrum:

$$v_{in}(t) = \sum_{k=1}^n 2|C'| |P| \cos(k\omega_0 t + \angle C' + \angle P) \quad (3.44)$$

$$C' = C_0 \frac{\sin(nD\pi)}{(nD\pi)} e^{-jnD\pi} \quad (3.45)$$

$$C_0 = V_{in_{DC}(nom)} \times D \quad (3.46)$$

- 1st harmonic has a frequency of $\omega_0 = 2\pi f_s$ rad/s where the f_s is the switching frequency.
- Input is applied to:

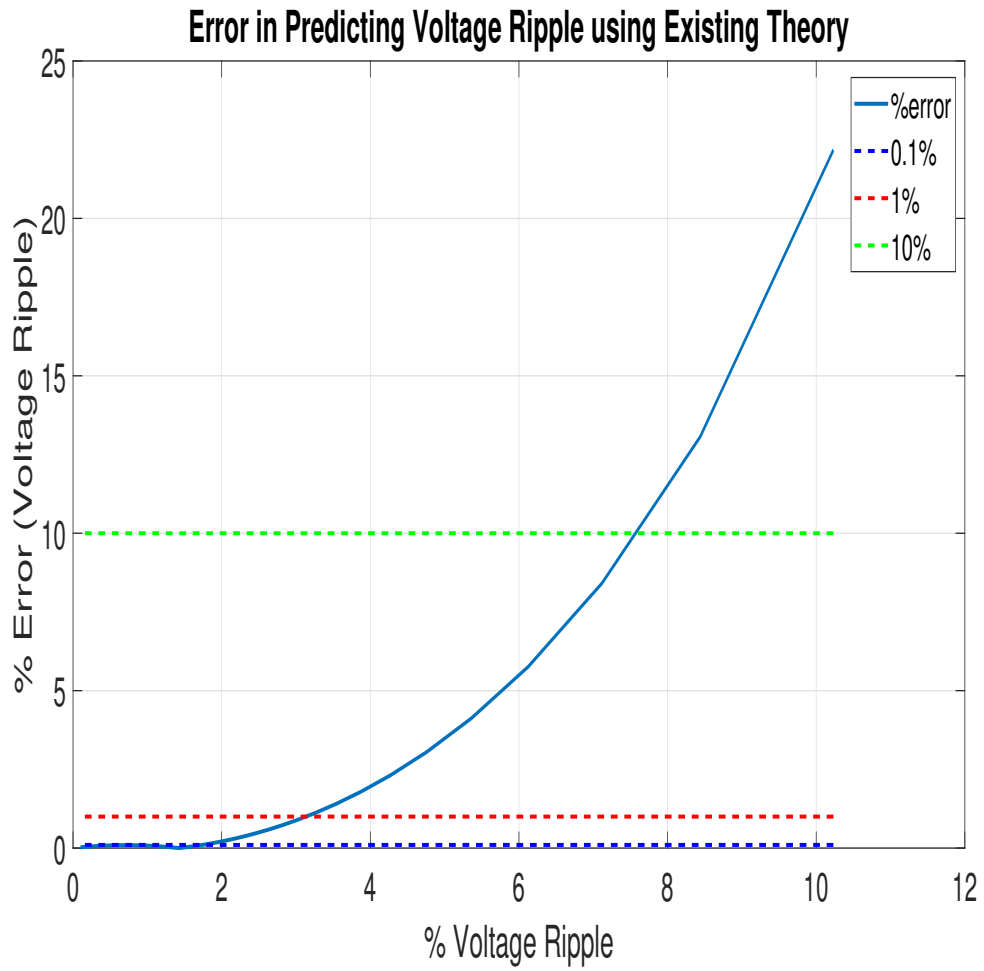


Figure 3.3: Limitations of Traditional Filter Design Equations

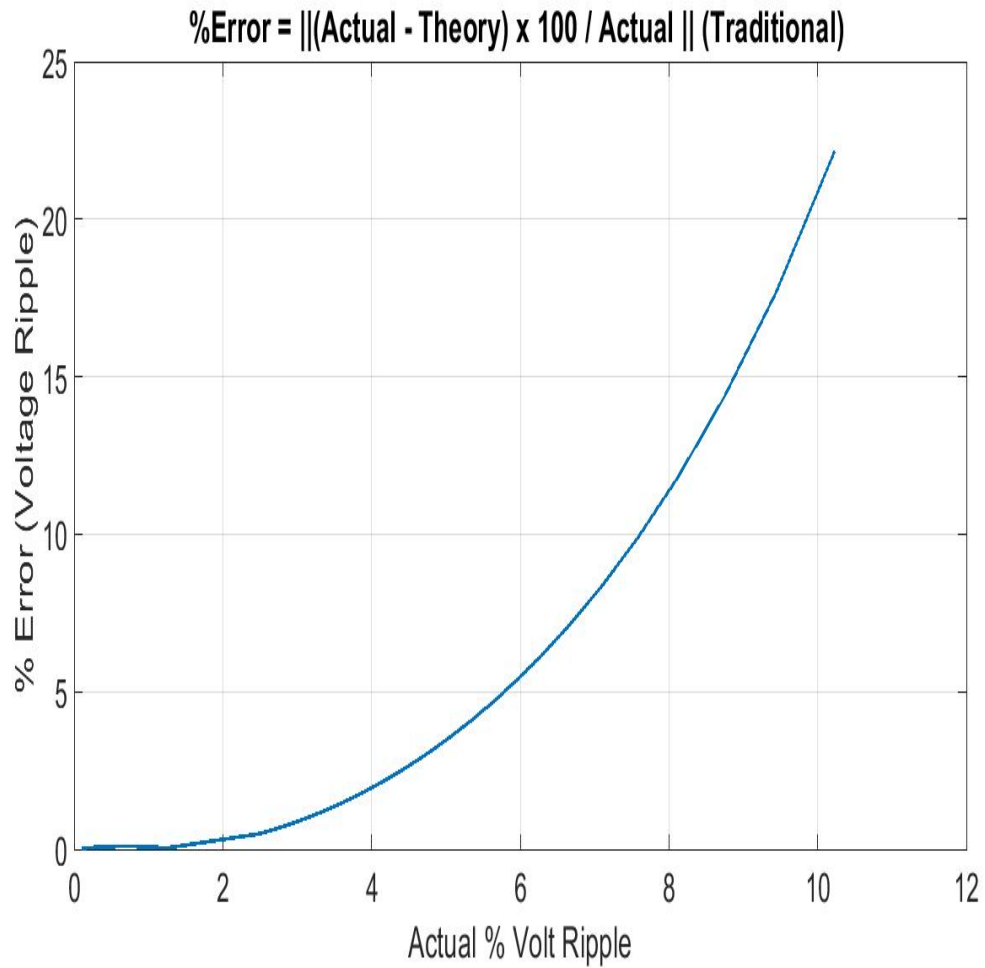


Figure 3.4: Error vs. Ripple

- Model 1: Full Transfer Function

$$P(s) = \frac{\frac{1}{LC}}{s^2 + \frac{1}{RC}s + \frac{1}{LC}} \quad (3.47)$$

- Model 2: High switching frequency approx. ($f_s \gg f_c$)

$$P = \frac{1}{s^2} \quad (3.48)$$

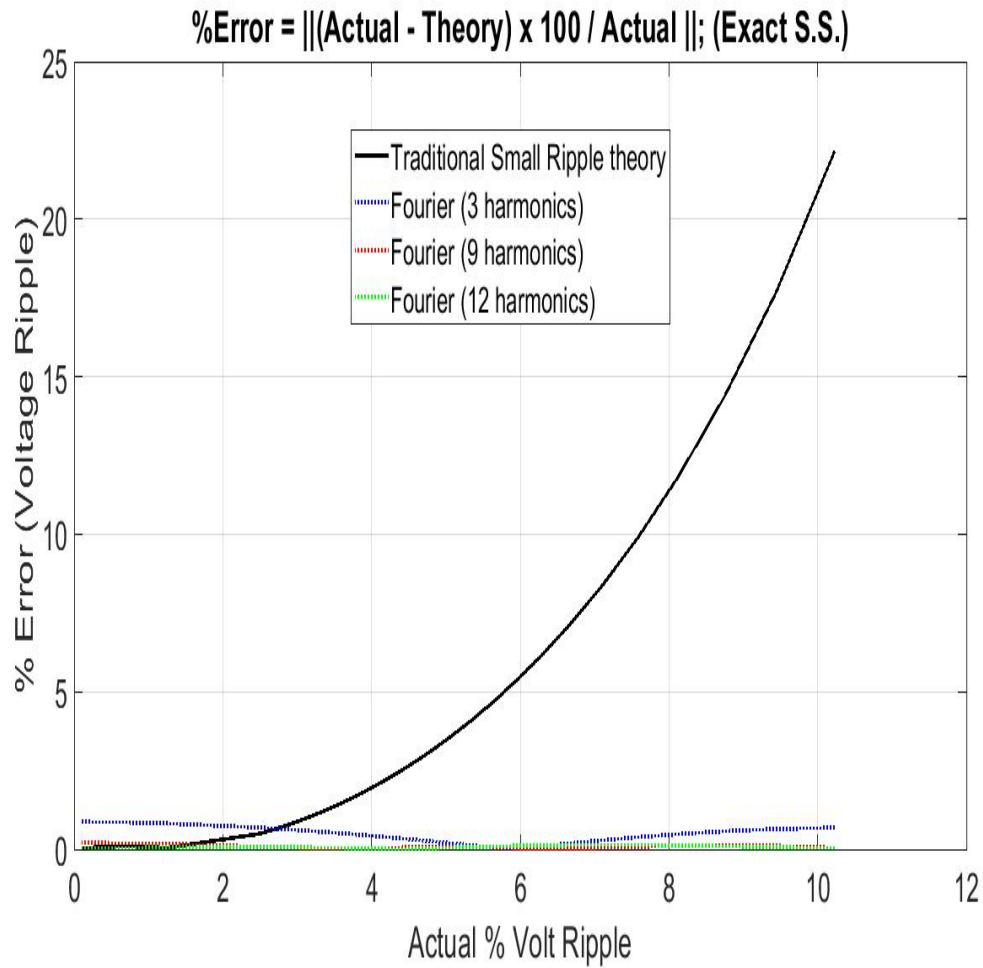


Figure 3.5: Model 1: Error vs. Ripple

Fourier Analysis - Conclusions

- State-space model with 9 harmonics is a better model.
- We can potentially reduce the number of design iterations with a more accurate model.
- Small-ripple approximation is conceptually similar to assuming $f_s \gg f_c$.

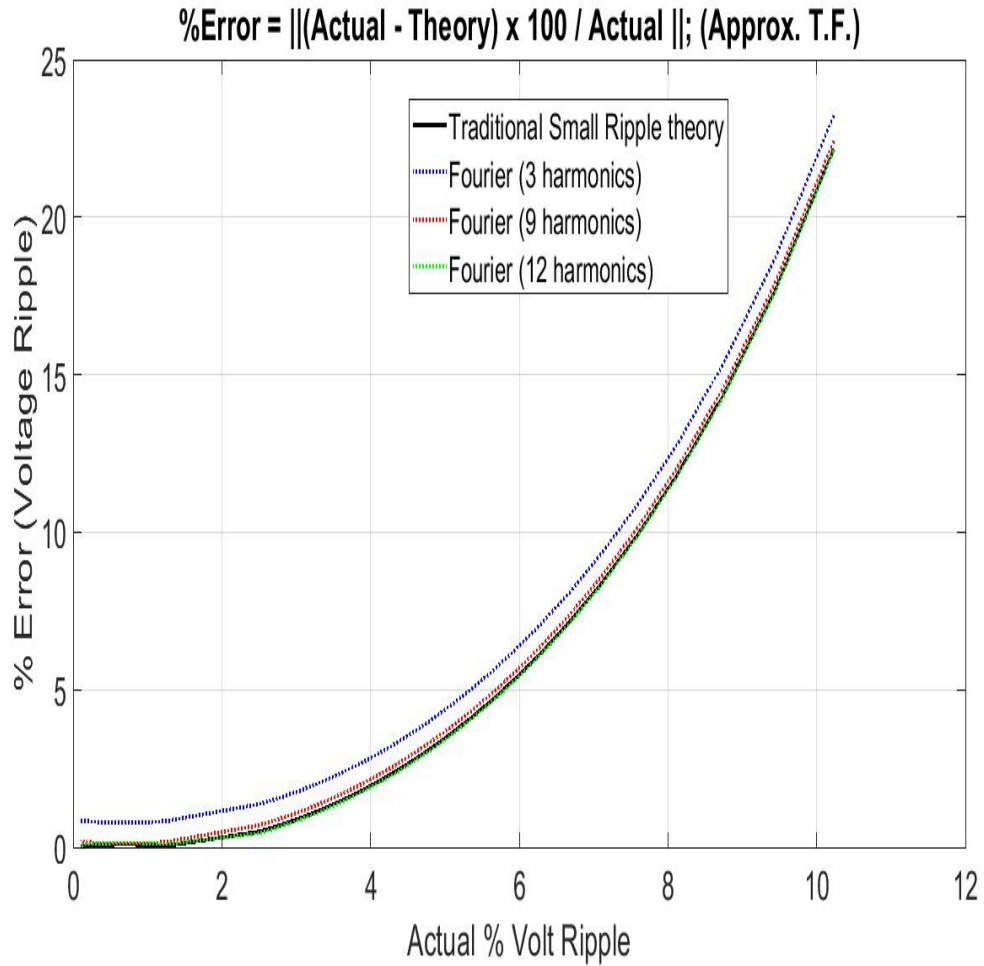


Figure 3.6: Model 2: Error vs. Ripple

3.6 Requirements for a Fourth Order LCLC Filter

The plant parameters and specifications are taken from the work on integrated buck converters published in the Transactions [60]. As seen in Fig. 3.7, the 4th Order LC-LC Output Filter achieves -40 dB of attenuation at a lower switching frequency. This analysis did not consider factors affecting the physical implementation of integrated buck converters like for example the mutual inductance between L_1 and L_2 . In reality we can achieve a more significant reduction in the switching frequency.

- For integrated buck converters, there is a finite chip area for placing the L and

C, thus limiting the maximum amount of total L and C that can be placed [60]

- Previous designs of integrated buck converters used high switching frequencies in **range of hundreds of megahertz** to decrease the size of components [65], [66].
- These lead to degradation in the efficiency:
 - due to increased switching loss at higher switching frequencies
 - due to the low quality factor of on-chip inductors
- Fourth-order LC low-pass filter is advocated in [60] as it can potentially deliver better performance without incurring an area penalty

NOTE: an **inductor is required at the input** for keeping the **current ripple** to a value acceptable to the FET while a **capacitor is required at the output** to keep the **voltage ripple** to a value acceptable to the load; thus, all filters are a series of LC circuits and the filter order is a multiple of 2.

Things to Consider while Choosing Filter Order

- Amount of attenuation needed. As shown in the next slide, for -20 dB of attenuation, a simple 2^{nd} order LC filter might be better than a 4^{th} order LCLC or a 6^{th} LCLCLC.
- Switching power loss. By reducing the switching frequency (f_s), we can reduce the switching power loss (P_{sw}), based on the equation [115, eq. (5.7)]:

$$P_{sw} = 10f_s C_{ds} \sqrt{V_I^3} \quad (3.49)$$

where

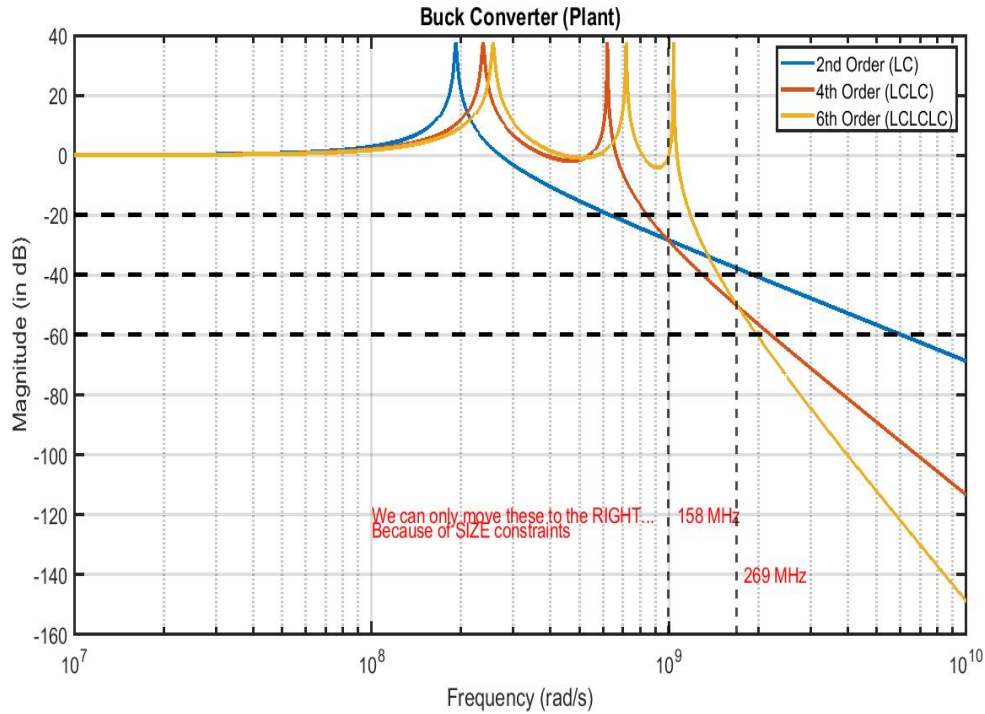


Figure 3.7: Frequency Response of Higher Order Output Filters

- f_s : the switching frequency
- V_I : input DC voltage
- C_{ds} : the drain to source capacitance can be computed using the data-sheet parameters (CSD13383F4):

$$C_{ds} = C_{oss} - C_{rss} = (68 - 47) \text{ pF} = 21 \text{ pF} \quad (3.50)$$

- Mutual inductance between the multiple inductors. This can increase or decrease the attenuation achieved.
- Total power loss including the conduction loss.

Family of Higher-Order Filters

- Fig. (3.7) above shows that below 158 MHz, the 2nd order filter will be superior in terms of attenuation.
- Between 158 MHz and 269 MHz the 4th order filter will be the best and above 269 MHz a 6th order filter will be the best for attenuation.
- Tab. 3.2 shows the filter order that is suitable based on the amount of attenuation desired at the switching frequency.

Attenuation	-20 dB	-40 dB	-60 dB
Filter Order	2	4	6

Table 3.2: Attenuation and Filter Order

- EXAMPLE: $V_I = 1$ V, $V_O = 0.8$ V

f_s	750 MHz	450 MHz
P_{sw}	157.5 mW	94.5 mW

Table 3.3: Switching Frequency and Power Loss

- Which is a 40 % reduction in switching power loss.
- If -60 dB of attenuation is desired, we can reduce the f_s from 750 MHz to 450 MHz by increasing the order of the filter from 2nd to 4th order.

Main challenges in LCLC Filter Design [60]:

- Coupling between the 2 inductors affects attenuation produced by the filter.
- Possible configuration of the LCLC filter based on coupling between the 2 inductors:
 1. No coupling

2. (Placed at minimum distance allowed by the process) Negative coupling
 3. (Placed at minimum distance allowed by the process) Positive coupling
- Relationship between inductance and dimension of on-chip planar spiral inductor

3.7 State-Space Matrices with LCLC Filters

KVL and KCL equations from Fig. 3.8.

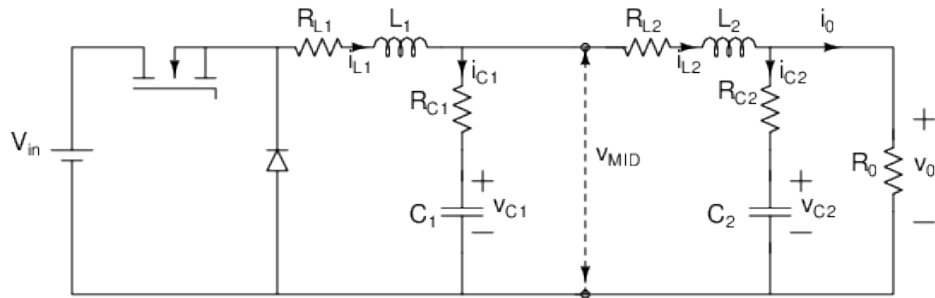


Figure 3.8: DC-DC Converter with LCLC-Filter

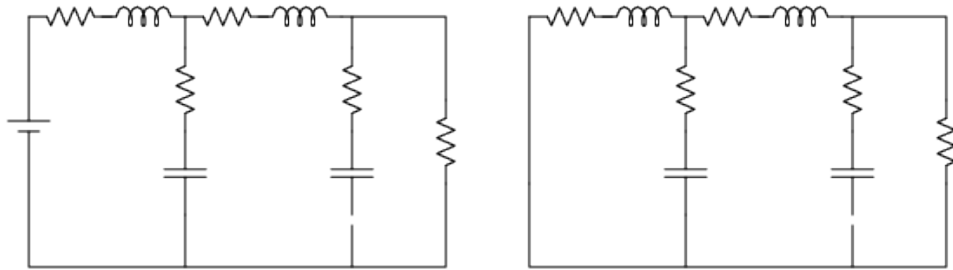


Figure 3.9: (a) Transistor ON $q(t) = 1$; (b) Transistor OFF $q(t) = 0$.

$$L_1 \frac{di_{L_1}(t)}{dt} = q(t)V_{in} - i_{L_1}R_{L_1} - v_{MID} \quad (3.51)$$

$$i_{C_1}R_{C_1} + v_{C_1} = v_{MID} \quad (3.52)$$

$$i_{L_1} = i_{C_1} + i_{L_2} \quad (3.53)$$

$$i_{C_1} = i_{L_1} - i_{L_2} \quad (3.54)$$

$$C_1 \frac{dv_{C_1}(t)}{dt} = i_{L_1} - i_{L_2} \quad (3.55)$$

$$L_2 \frac{di_{L_2}(t)}{dt} = v_{MID} - i_{L_2}R_{L_2} - v_0 \quad (3.56)$$

$$i_{C_2} = C_2 \frac{dv_{C_2}(t)}{dt} = i_{L_2} - \frac{v_0}{R_0} \quad (3.57)$$

$$v_0 = v_{C_2} + i_{C_2}R_{C_2} \quad (3.58)$$

$$\Rightarrow v_0 = v_{C_2} + \left(i_{L_2} - \frac{v_0}{R_0} \right) R_{C_2} \quad (3.59)$$

$$(i_{L_1} - i_{L_2})R_{C_1} + v_{C_1} = v_{MID} \quad (3.60)$$

$$L_1 \frac{di_{L_1}(t)}{dt} = q(t)V_{in} - i_{L_1}R_{L_1} - (i_{L_1} - i_{L_2})R_{C_1} - v_{C_1} \quad (3.61)$$

$$L_2 \frac{di_{L_2}(t)}{dt} = (i_{L_1} - i_{L_2})R_{C_1} + v_{C_1} - i_{L_2}R_{L_2} - v_0 \quad (3.62)$$

$$\begin{aligned}
v_0 + v_0 \frac{R_{C_2}}{R_0} &= v_{C_2} + i_{L_2} R_{C_2} \\
\Rightarrow v_0 \left(1 + \frac{R_{C_2}}{R_0} \right) &= v_{C_2} + i_{L_2} R_{C_2} \\
\Rightarrow v_0 &= i_{L_2} \left(\frac{R_0 R_{C_2}}{R_0 + R_{C_2}} \right) + v_{C_2} \left(\frac{R_0}{R_0 + R_{C_2}} \right) \quad (3.63)
\end{aligned}$$

$$\begin{aligned}
L_2 \frac{di_{L_2}(t)}{dt} &= (i_{L_1} - i_{L_2}) R_{C_1} + v_{C_1} - i_{L_2} R_{L_2} \\
&\quad - i_{L_2} \left(\frac{R_0 R_{C_2}}{R_0 + R_{C_2}} \right) - v_{C_2} \left(\frac{R_0}{R_0 + R_{C_2}} \right) \\
\Rightarrow L_2 \frac{di_{L_2}(t)}{dt} &= i_{L_1} R_{C_1} - i_{L_2} \left(R_{C_1} + R_{L_2} + \frac{R_0 R_{C_2}}{R_0 + R_{C_2}} \right) + v_{C_1} - v_{C_2} \left(\frac{R_0}{R_0 + R_{C_2}} \right) \quad (3.64)
\end{aligned}$$

$$C_2 \frac{dv_{C_2}(t)}{dt} = i_{L_2} - i_{L_2} \left(\frac{R_{C_2}}{R_0 + R_{C_2}} \right) - v_{C_2} \left(\frac{1}{R_0 + R_{C_2}} \right) \quad (3.65)$$

After reorganization,

$$\begin{aligned}
\frac{di_{L_1}(t)}{dt} &= q(t) \left(\frac{V_{in}}{L_1} \right) - i_{L_1} \left(\frac{R_{L_1} + R_{C_1}}{L_1} \right) - v_{C_1} \left(\frac{1}{L_1} \right) + i_{L_2} \left(\frac{R_{C_1}}{L_1} \right) \\
\frac{dv_{C_1}(t)}{dt} &= i_{L_1} \left(\frac{1}{C_1} \right) - i_{L_2} \left(\frac{1}{C_1} \right) \\
\frac{di_{L_2}(t)}{dt} &= i_{L_1} \left(\frac{R_{C_1}}{L_2} \right) - i_{L_2} \left(\frac{1}{L_2} \right) \left(R_{C_1} + R_{L_2} + \frac{R_0 R_{C_2}}{R_0 + R_{C_2}} \right) \\
&\quad + v_{C_1} \left(\frac{1}{L_2} \right) - v_{C_2} \left(\frac{1}{L_2} \right) \left(\frac{R_0}{R_0 + R_{C_2}} \right) \\
\frac{dv_{C_2}(t)}{dt} &= i_{L_2} \left(\frac{R_0}{C_2(R_0 + R_{C_2})} \right) - v_{C_2} \left(\frac{1}{C_2(R_0 + R_{C_2})} \right)
\end{aligned} \tag{3.66}$$

Above equations can be represented in the matrix form as

$$\begin{bmatrix} \frac{di_{L_1}(t)}{dt} \\ \frac{dv_{C_1}(t)}{dt} \\ \frac{di_{L_2}(t)}{dt} \\ \frac{dv_{C_2}(t)}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{L_1} (R_{L_1} + R_{C_1}) & -\frac{1}{L_1} & \frac{R_{C_1}}{L_1} & 0 \\ \frac{1}{C_1} & 0 & -\frac{1}{C_1} & 0 \\ \frac{R_{C_1}}{L_2} & \frac{1}{L_2} & k_2 & k_3 \\ 0 & 0 & -R_0 k_4 & k_4 \end{bmatrix} \begin{bmatrix} i_{L_1}(t) \\ v_{C_1}(t) \\ i_{L_2}(t) \\ v_{C_2}(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L_1} u(t) \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.67}$$

- $k_2 = \left(-\frac{1}{L_2} \right) \left(R_{C_1} + R_{L_2} + \frac{R_0 R_{C_2}}{R_0 + R_{C_2}} \right)$
- $k_3 = \left(-\frac{1}{L_2} \right) \left(\frac{R_0}{R_0 + R_{C_2}} \right)$
- $k_4 = \left(-\frac{1}{C_2} \right) \left(\frac{1}{R_0 + R_{C_2}} \right)$

- $u(t) = q(t)V_{in}$.

Therefore, we get the two sets of state space equations as:

Transistor – ON ($q(t) = 1$):

$$\begin{aligned}\dot{\mathbf{x}}(t) &= A_1\mathbf{x}(t) + B_1\mathbf{u}(t) \\ \mathbf{y}(t) &= E_1\mathbf{x}(t) + F_1\mathbf{u}(t)\end{aligned}\tag{3.68}$$

and Transistor – OFF ($q(t) = 0$):

$$\begin{aligned}\dot{\mathbf{x}}(t) &= A_2\mathbf{x}(t) + B_2\mathbf{u}(t) \\ \mathbf{y}(t) &= E_2\mathbf{x}(t) + F_2\mathbf{u}(t)\end{aligned}$$

where

$$A_1 = A_2 = \begin{bmatrix} -\frac{1}{L_1}(R_{L_1} + R_{C_1}) & -\frac{1}{L_1} & \frac{R_{C_1}}{L_1} & 0 \\ \frac{1}{C_1} & 0 & -\frac{1}{C_1} & 0 \\ \frac{R_{C_1}}{L_2} & \frac{1}{L_2} & k_2 & k_3 \\ 0 & 0 & \frac{1}{C_2} \frac{R_0}{(R_0 + R_{C_2})} & k_4 \end{bmatrix}\tag{3.69}$$

- $k_2 = \left(-\frac{1}{L_2}\right) \left(R_{C_1} + R_{L_2} + \frac{R_0 R_{C_2}}{R_0 + R_{C_2}}\right)$
- $k_3 = \left(-\frac{1}{L_2}\right) \left(\frac{R_0}{R_0 + R_{C_2}}\right)$

- $k_4 = \left(-\frac{1}{C_2}\right) \left(\frac{1}{R_0 + R_{C_2}}\right)$

$$B_1 \mathbf{u} = \begin{bmatrix} \frac{V_{in}}{L_1} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad B_2 \mathbf{u} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (3.70)$$

$$E_1 = E_2 = \begin{bmatrix} 0 & 0 & \frac{R_0 R_{C_2}}{R_0 + R_{C_2}} & \frac{R_0}{R_0 + R_{C_2}} \end{bmatrix}, \quad (3.71)$$

$$F_1 = F_2 = \begin{bmatrix} \mathbf{0} \end{bmatrix} \quad (3.72)$$

and the vectors of state variables, output and input be:

$$\mathbf{x} = \begin{bmatrix} i_{L_1} \\ v_{C_1} \\ i_{L_2} \\ v_{C_2} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} v_0 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} V_{in} \end{bmatrix} \quad (3.73)$$

Following a similar analysis as in Buck converter with LC filter we obtain linear averaged small-signal model as:

$$\begin{aligned} \frac{d\hat{\mathbf{x}}(t)}{dt} &= A\hat{\mathbf{x}}(t) + B\hat{\mathbf{u}}(t) + M\hat{d}(t) \\ \mathbf{y}(t) &= E\hat{\mathbf{x}}(t) \end{aligned} \quad (3.74)$$

$$\text{where, } \hat{\mathbf{x}}(t) = \begin{bmatrix} \hat{i}_{L_1}(t) \\ \hat{v}_{C_1}(t) \\ \hat{i}_{L_2}(t) \\ \hat{v}_{C_2}(t) \end{bmatrix}, \quad \hat{\mathbf{y}}(t) = \hat{v}_0(t), \quad \hat{\mathbf{u}}(t) = \hat{v}_{in}(t), \quad (3.75)$$

are small perturbations in the state, output and input variables.

$$A = \begin{bmatrix} -\frac{1}{L_1}(R_{L_1} + R_{C_1}) & -\frac{1}{L_1} & \frac{R_{C_1}}{L_1} & 0 \\ \frac{1}{C_1} & 0 & -\frac{1}{C_1} & 0 \\ \frac{R_{C_1}}{L_2} & \frac{1}{L_2} & k_2 & k_3 \\ 0 & 0 & \frac{1}{C_2} \frac{R_0}{(R_0 + R_{C_2})} & k_4 \end{bmatrix} \quad (3.76)$$

- $k_2 = \left(-\frac{1}{L_2}\right) \left(R_{C_1} + R_{L_2} + \frac{R_0 R_{C_2}}{R_0 + R_{C_2}}\right)$
- $k_3 = \left(-\frac{1}{L_2}\right) \left(\frac{R_0}{R_0 + R_{C_2}}\right)$
- $k_4 = \left(-\frac{1}{C_2}\right) \left(\frac{1}{R_0 + R_{C_2}}\right)$

$$B = \begin{bmatrix} \frac{D}{L_1} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad M = \begin{bmatrix} \frac{V_{in}}{L_1} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (3.77)$$

$$E = \begin{bmatrix} 0 & 0 & \frac{R_0 R_{C_2}}{R_0 + R_{C_2}} & \frac{R_0}{R_0 + R_{C_2}} \end{bmatrix}, \quad (3.78)$$

$$F = \begin{bmatrix} \mathbf{0} \end{bmatrix} \quad (3.79)$$

Like in the case of Buck converter with LC filter, plant transfer function for LCLC

filter can be obtained as:

$$Trplant(s) = E [sI - A]^{-1} B.$$

For a special case : $L_1 = L_2 = L$ and $C_1 = C_2 = C$,

$$Trplant(s) = D \frac{\frac{1}{L^2 C^2}}{s^3 \left(s + \frac{1}{R_0 C} \right) + \frac{s^2}{LC} + \frac{2s}{LC} \left(s + \frac{1}{R_0 C} \right) + \frac{1}{L^2 C^2}} \quad (3.80)$$

3.8 Summary

This chapter presented the state-space model of the buck converter plant along with the traditional plant design. It also presented the limitations of the two traditional plant design equations which are based on the Small Ripple Approximation. This inaccuracy was numerically quantified and an alternate method was presented which doesn't rely on the same assumption. Finally, conditions which necessitate a fourth order filter are analyzed, and the model of the buck converter with the fourth order filter is presented.

CONTROL SYSTEM DESIGN FOR BUCK CONVERTER

4.1 Mathematical Preliminaries

We focus on the design of causal, finite dimensional, linear time invariant (LTI) controllers in this thesis. With C , R and Z being the set of all complex, real, and integer numbers, respectively; the following definitions are pertinent to the discussion in this chapter. We begin with the definition of a norm [113]:

Definition 4.1.1. A norm $\|\cdot\|_{\mathcal{V}}$ on a vector space \mathcal{V} is a function mapping $\mathcal{V} \rightarrow [0, \infty)$ which, for each $v \in \mathcal{V}$ satisfies

- a. $\|v\|_{\mathcal{V}} = 0$ iff $v = 0$;
- b. $|\alpha| \cdot \|v\|_{\mathcal{V}} = \|\alpha v\|_{\mathcal{V}}$, for all scalars α ;
- c. $\|u + v\|_{\mathcal{V}} \leq \|u\|_{\mathcal{V}} + \|v\|_{\mathcal{V}}$, for all u in \mathcal{V} .

We are interested in certain specific norms, which will be used to quantify controller performance. These norms are defined on function spaces. The pertinent ones are the \mathcal{L}_2 , \mathcal{H}_∞ and $R\mathcal{H}_\infty$ spaces defined below [114]:

Definition 4.1.2. $\mathcal{L}_2 \stackrel{\text{def}}{=} \mathcal{L}_2(\mathbb{R})$ denotes the space of Lebesgue square integrable complex-valued functions with support on \mathbb{R} . $\mathcal{L}_2(\mathbb{R}_+)$ and $\mathcal{L}_2(\mathbb{R}_-)$ are similarly defined. \mathcal{L}_2 is a normed linear space over the field C , when endowed with the following norm

$$\|f\|_{\mathcal{L}_2} \stackrel{\text{def}}{=} \sqrt{\int_{-\infty}^{\infty} |f(t)|^2 dt} \quad (4.1)$$

Definition 4.1.3. $\mathcal{H}_\infty \stackrel{\text{def}}{=} \mathcal{H}_\infty(C_+)$ denotes the *Hardy space* of complex-valued functions which are analytic and essentially bounded in C_+ . \mathcal{H}_∞ is a normed linear space over the field C , when endowed with the norm

$$\|F\|_{\mathcal{H}_\infty} \stackrel{\text{def}}{=} \sup_{\sigma>0} \sup_{\omega \in \mathbb{R}_e} |F(\sigma + j\omega)| = \sup_{\mathbb{R}_e s > 0} [F(s)] < \infty \quad (4.2)$$

$R\mathcal{H}_\infty$ denotes the corresponding subspace of real-rational \mathcal{H}_∞ functions.

The first norm that we study is the $\mathcal{L}_2(\mathbb{R}_+)$ -induced norm of a system T_{wz} . The \mathcal{H}_∞ norm is the \mathcal{L}_2 -induced norm of a causal, stable, linear time invariant system [113].

We study the internal stability of these systems are defined below [114]:

Definition 4.1.4 (Induced- \mathcal{L}_2 Finite-Gain Stability). An LTI system F is said to be induced- \mathcal{L}_2 finite-gain stable if

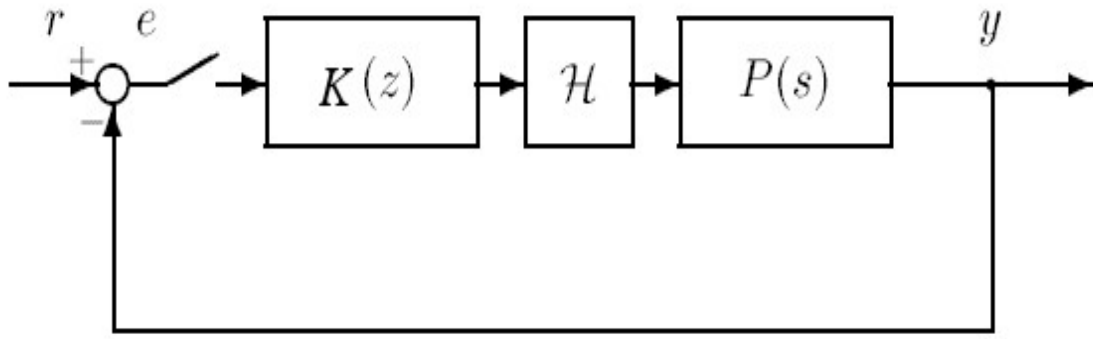
$$\|F\|_{\mathcal{L}_2 \rightarrow \mathcal{L}_2} \stackrel{\text{def}}{=} \sup_{x \neq 0} \frac{\|Fx\|_{\mathcal{L}_2}}{\|x\|_{\mathcal{L}_2}} < \infty \quad (4.3)$$

i.e., there exists a finite constant $\gamma \in [0, \infty)$ such that

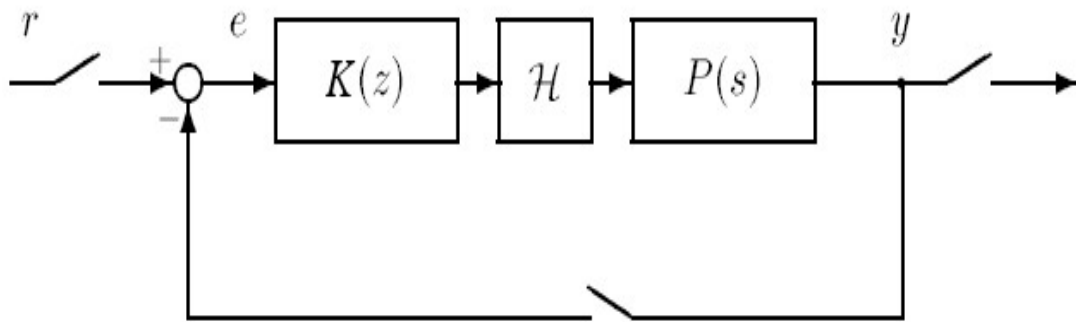
$$\|Fx\|_{\mathcal{L}_2} \leq \gamma \|x\|_{\mathcal{L}_2} \quad (4.4)$$

4.2 Need for Sampled-Data Control

A SD control system, like the one shown in Fig. 4.2 has both analog and digital components. The plant is an analog plant (in this case an LC filter circuit with a resistive load R_o) and the controller is digital. Since this digital subsystem operates within an analog control loop, the controller has an ADC or sampler (S_{T_s}) and a DAC or hold (H_{T_s}). The hold circuit is often approximated by a zero-order-hold (ZOH) of the form $\frac{1-e^{-sT_s}}{sT_s}$, but when using the direct discretization method, we add more



(a) *Actual System*



(b) *Discretized System*

Figure 4.1: Actual System Versus System Obtained using Typical Discretization

fictitious samplers to create a fully digital system as shown in Fig. 4.1 (b). This might lead to loss of inter-sample behavior.

Fig. 2.1 shows the schematic of the buck converter with SD feedback control. As we attempt to design the controller K_d , we start by considering the weighted \mathcal{H}_∞ mixed sensitivity control design framework for analog controller design in Fig. 4.3. The SD version of this is shown in Fig. 4.6.

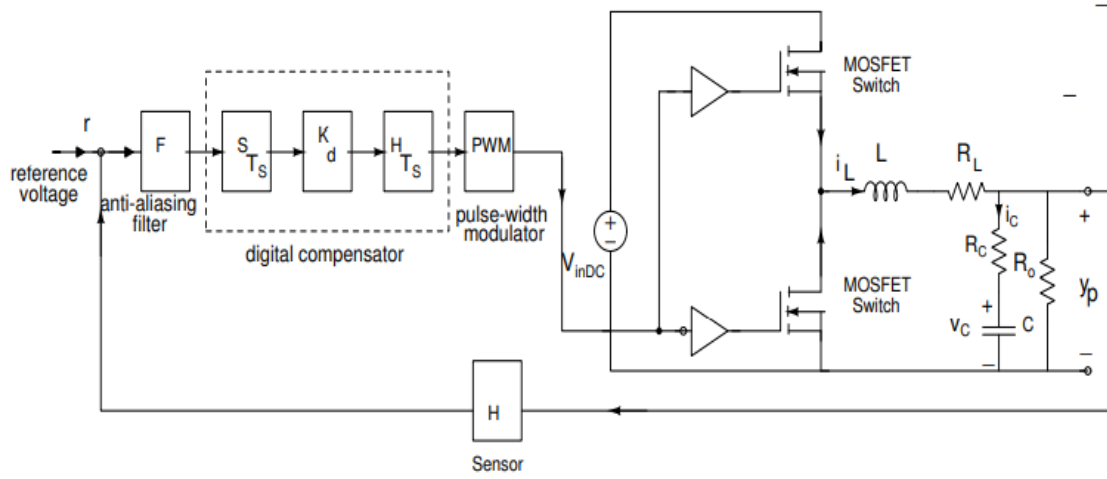


Figure 4.2: Schematic for DC-DC Buck Converter Circuit with Compensation [5], [6]

4.2.1 Analog Controller Design

Consider the weighted \mathcal{H}_∞ mixed sensitivity problem shown in Fig. 4.3 [89, 94, 86, 100].

The analog controller design involves minimizing the \mathcal{H}_∞ norm:

$$\|T_{wz}\| = \left\| \begin{bmatrix} W_1 S_o \\ W_2 K S_o \\ W_3 T_o \end{bmatrix} \right\|_\infty, \quad (4.5)$$

where

- $S_o = T_{w \rightarrow e}$
- $K S_o = T_{w \rightarrow u}$
- $T_o = T_{w \rightarrow \hat{z}_3}$.

We only choose a weight on the error e for designing the controller as we are

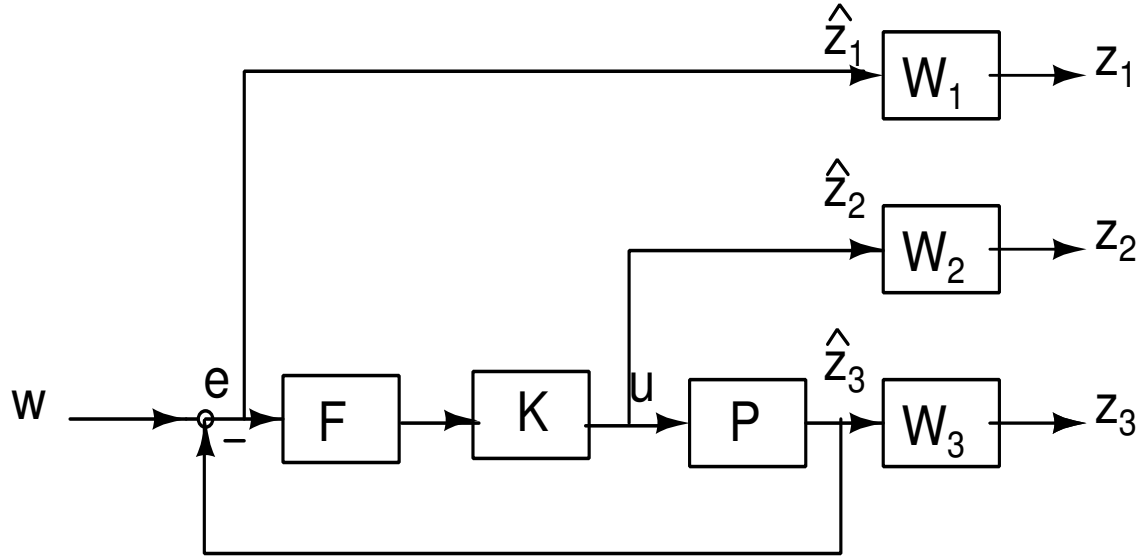


Figure 4.3: Visualization of Weighted Mixed Sensitivity Analog Controller Design [89]

studying the trade-off between power loss and controller performance (which is measured in terms of a small induced- \mathcal{L}_2 norm of T_o and a small steady-state error). The open-loop frequency response of the analog design is shown in Fig. 4.4. The unity gain crossover is 130 kHz and the phase margin is 111.5°.

The the following structure was chosen for the weighting function W_1 on the error e :

$$W_1(s) = \frac{1}{M_s} \left[\frac{s + M_s \omega_b}{s + \epsilon \omega_b} \right] \left[\frac{a_w}{s + a_w} \right] \quad (4.6)$$

where the parameters in eq. (4.6) are chosen to be:

- $\epsilon = 0.1$
- $M_s = 5$
- $\omega_b = 1000 \text{ rad/s}$

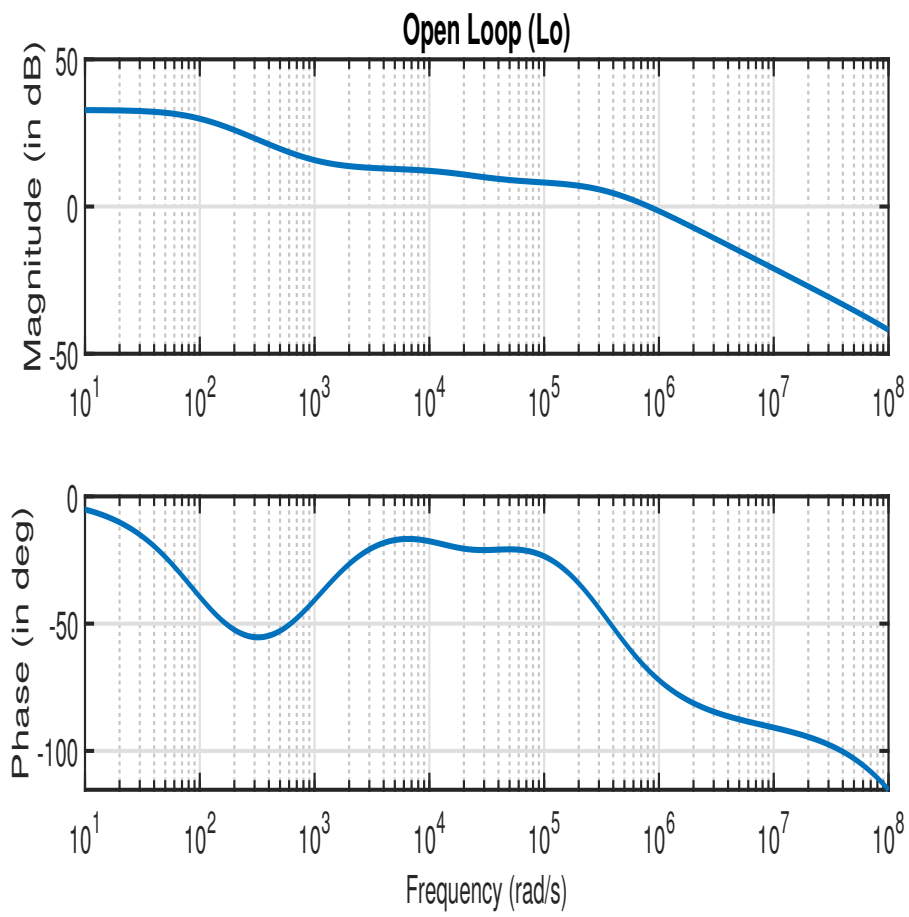


Figure 4.4: Nominal Open Loop Frequency Response

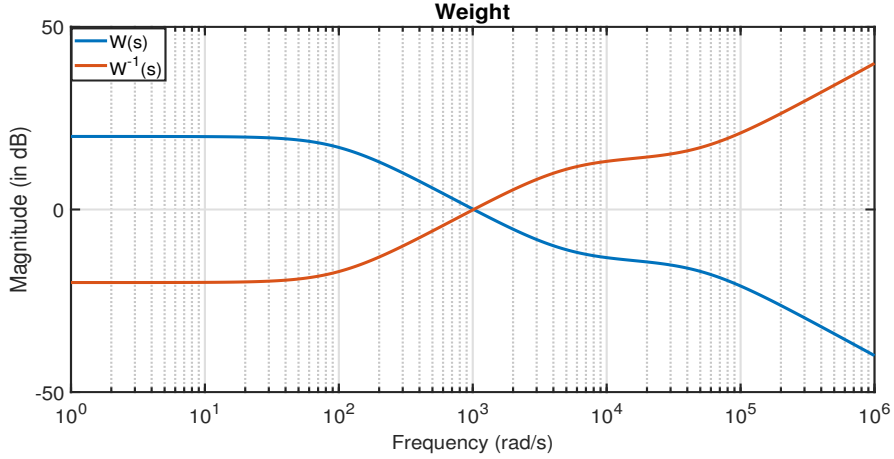


Figure 4.5: Weight on Sensitivity

- $a_w = 10M_s\omega_b$

The magnitude responses of W_1 and its inverse are shown in Fig. 4.5.

4.2.2 Digital Controller Design

For the SD control system shown in Fig. 4.6, the ADC and DPWM both operate at the same frequency. This switching/sampling frequency (both denoted f_s hereafter) is varied in the range $f_s \in (4.546, 100)$ kHz. The anti-aliasing filter (AAF) F shown in Fig. 4.3 is chosen as the first-order roll-off,

$$F(s) = \frac{\omega_b}{s + \omega_b}. \quad (4.7)$$

As discussed above, the indirect design method does not capture the inter-sample behavior. This is apparent after examining Fig. 4.1.

As seen in this figure, the actual SD digital control system is the one shown in Fig. 4.1 (a). However, when we utilize the *indirect design method* for controller design, we implicitly insert fictitious samplers into the closed loop system. Hence,

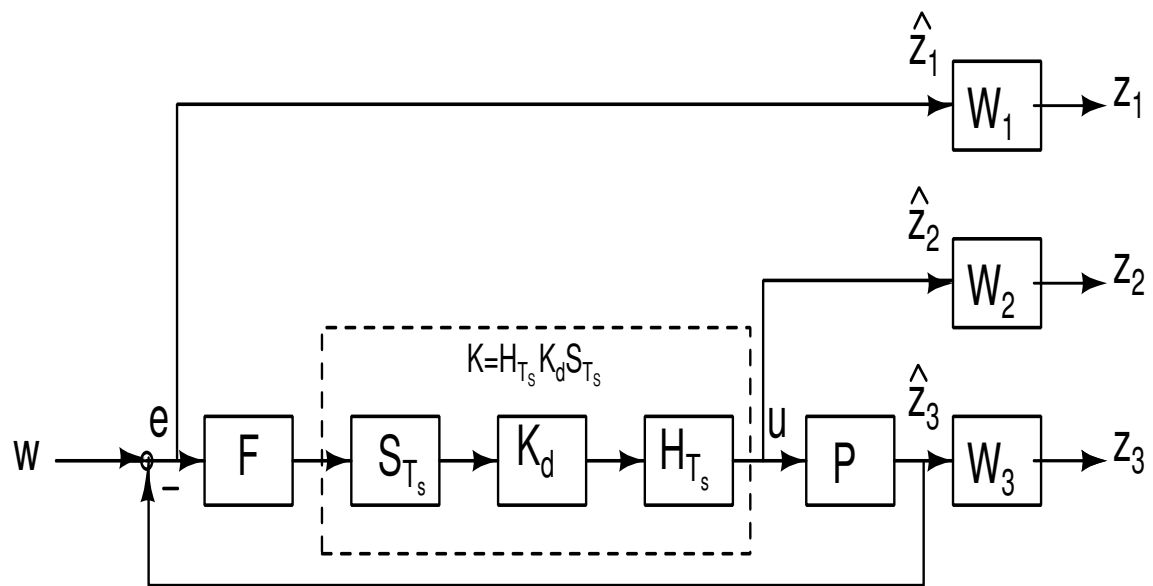


Figure 4.6: Visualization of Weighted Mixed Sensitivity Sampled-Data Controller Design [89]

the system we design for using the indirect method is the one depicted in Fig. 4.1 (b) instead of the one we wish to design for, i.e., the one in Fig. 4.1 (a) [14]. The disadvantage of this *indirect* approach is that the behavior of the SD system is only described at the sampling instants and all inter-sample behavior is left out of the model. Naturally, this means that if the sampling time interval is too large, the closed-loop stability/performance of the actual SD system may be lost even if the model in Fig. 4.1 (b) is stable with an acceptable closed-loop performance.

Indirect design Method: This is the traditional method for discretizing controllers [89] and involves the bilinear transformation:

$$s = \frac{1 - \lambda}{1 + \lambda} \quad (4.8)$$

where $\lambda = \frac{1}{z}$ with z being the discrete-time z -transform variable. This should not be confused with the outputs of the LFT in the generalized feedback framework.

The effect of utilizing a model like the one in Fig. 4.1 (b) while designing the discrete controller is that we are only studying the effect of disturbances and reference commands on the controlled signals (e.g. closed loop error or plant output) at the sampling instants. Meanwhile, the plant evolves in continuous-time and disturbances are also continuous-time signals. We can address this issue directly using the concept of lifting as mentioned earlier in the Introduction. The lifted model incorporates inter-sample behavior and hence represents a more accurate model for controller design.

Quantifying the closed loop performance in terms of norms, the \mathcal{H}_∞ norm of discrete-time system (fig 4.1 (b)) might be small even if the \mathcal{L}_2 -induced norm of the lifted system (fig 4.1 (a)) is large. This second norm is a more accurate measure of the closed loop system's performance as it takes inter-sample behavior into account.

Direct Design Method: This involves a discrete-time \mathcal{H}_∞ -optimization using the bilinear transformation [89].

- **Step 1**: We start with a discrete-time model of the system:

$$G_d(\lambda) = \left[\begin{array}{c|c} A_d & B_d \\ \hline C_d & D_d \end{array} \right] = \left[\begin{array}{c|cc} A_d & B_{d1} & B_{d2} \\ \hline C_{d1} & D_{d11} & D_{d12} \\ C_{d2} & D_{d21} & D_{d22} \end{array} \right] \quad (4.9)$$

This can be obtained using the step-invariant transformation of the continuous-time plant for the given value of T_s .

- **Step 2**: Define an artificial continuous-time system G_c :

$$G_c(s) = \left[\begin{array}{c|c} A_c & B_c \\ \hline C_c & D_c \end{array} \right] = \left[\begin{array}{c|cc} A_c & B_{c1} & B_{c2} \\ \hline C_{c1} & D_{c11} & D_{c12} \\ C_{c2} & D_{c21} & D_{c22} \end{array} \right] \quad (4.10)$$

where

$$A_c = (A_d - I)(A_d + I)^{-1} \quad (4.11)$$

$$B_c = (I - A_c)B_d \quad (4.12)$$

$$C_c = C_d(A_d + I)^{-1} \quad (4.13)$$

$$D_c = D_d - C_c B_d \quad (4.14)$$

- **Step 3**: Once we have G_c , we design an $\mathcal{H}_\infty(\mathbb{C}_+)$ -optimal controller for it. Let

the state-space model for this controller be

$$K_c(s) = \left[\begin{array}{c|c} A_{K_c} & B_{K_c} \\ \hline C_{K_c} & D_{K_c} \end{array} \right] \quad (4.15)$$

- **Step 4:** The final $\mathcal{H}_\infty(\mathbb{D})$ -optimal controller for G_d is

$$K_d(\lambda) = \left[\begin{array}{c|c} A_{K_d} & B_{K_d} \\ \hline C_{K_d} & D_{K_d} \end{array} \right] \quad (4.16)$$

where

$$A_{K_d} = (I - A_{K_c})^{-1}(I + A_{K_c}) \quad (4.17)$$

$$B_{K_d} = (I - A_{K_c})^{-1}B_{K_c} \quad (4.18)$$

$$C_{K_d} = C_{K_c}(I + A_{K_d}) \quad (4.19)$$

$$D_{K_d} = D_{K_c} + C_{K_c}B_{K_d} \quad (4.20)$$

This procedure requires that $A_d + I$ and $I - A_{K_c}$ are invertible. Furthermore, the continuous-time \mathcal{H}_∞ problem in eq. (4.15) must be regular.

Lifting-Based Design Method: The lifting-based method used in this work follows from the results in [89]. Firstly, denote the closed-loop mapping from exogenous reference command w to regulated signals z_1, z_2 , and z_3 by $\mathcal{F}(G, H_{T_s}K_dS_{T_s})$. The sampled-data (i.e., consisting of both continuous- and discrete-time signals) and time-varying nature of the closed-loop system $\mathcal{F}(G, H_{T_s}K_dS_{T_s})$ is undesirable from a synthesis perspective. However, using the T_s -periodicity of $\mathcal{F}(G, H_{T_s}K_dS_{T_s})$, we may lift the generalized plant G to obtain a lifted plant \underline{G} for which the associated closed-loop system $\mathcal{F}(\underline{G}, K_d)$ is discrete-time LTI. For the lifting, we begin with a

stabilizable and detectable realization of G ,

$$G = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ C_2 & 0 & 0 \end{array} \right]. \quad (4.21)$$

We have taken $D_{21} = 0$ because the sampler must be low-pass filtered for the induced- \mathcal{L}_2 norm of the closed-loop system to be finite. $D_{11} = 0$ is assumed as a technical simplification, more details can be found in the literature [89]. We lift G to obtain the lifted generalized plant \underline{G} , which has state model

$$\underline{G} = \left[\begin{array}{c|cc} A_d & \underline{B}_1 & B_{2d} \\ \hline \underline{C}_1 & \underline{D}_{11} & \underline{D}_{12} \\ C_2 & 0 & 0 \end{array} \right]. \quad (4.22)$$

Where $A_d := e^{AT_s}$, $B_{2d} := \int_0^{T_s} e^{A\tau} d\tau B_2$ and the the operator-valued entries [89]:

$$\underline{B}_1 : \mathcal{L}_2[0, T_s) \rightarrow \mathbb{R}^n, \quad \underline{B}_1 u = \int_0^{T_s} e^{A(T_s-\tau)} B_1 u(\tau) d\tau \quad (4.23)$$

$$\underline{C}_1 : \mathbb{R}^n \rightarrow \mathcal{L}_2[0, T_s), \quad (\underline{C}_1 x)(t) = C_1 e^{At} x \quad (4.24)$$

$$\underline{D}_{11} : \mathcal{L}_2[0, T_s) \rightarrow \mathcal{L}_2[0, T_s), \quad (\underline{D}_{11} w)(t) = C_1 \int_0^t e^{A(t-\tau)} B_1 w(\tau) d\tau \quad (4.25)$$

$$\underline{D}_{12} : \mathbb{R}^m \rightarrow \mathcal{L}_2[0, T_s), \quad (\underline{D}_{12} u)(t) = D_{12} u + C_1 \int_0^t e^{A(t-\tau)} d\tau B_2 u. \quad (4.26)$$

The lifted system \underline{G} is discrete-time LTI, with infinite-dimensional input and output spaces. Furthermore, under the above assumptions on the generalized plant

G , a controller K_d internally stabilizes $\mathcal{F}(G, H_{T_s} K_d S_{T_s})$ if and only if K_d internally stabilizes $\mathcal{F}(\underline{G}, K_d)$, and

$$\|\mathcal{F}(G, H_{T_s} K_d S_{T_s})\|_{\mathcal{L}_2} < \gamma \iff \|\mathcal{F}(\underline{G}, K_d)\|_{\mathcal{L}_2} < \gamma \quad (4.27)$$

for all $\gamma > \|\underline{D}_{11}\|$, where $\|\underline{D}_{11}\|$ denotes the Hilbert-Schmidt norm of the operator \underline{D}_{11} (whose numerical calculation is described below). Finally, for the purposes of synthesis, we reduce the infinite-dimensional system \underline{G} to a finite-dimensional LTI \mathcal{H}_∞ -equivalent system $G_{eq,d}$, whose state-space representation is given in (4.46). Similarly, under the above assumptions on the generalized plant G , a controller K_d internally stabilizes $\mathcal{F}(G, H_{T_s} K_d S_{T_s})$ if and only if K_d internally stabilizes the closed-loop system $\mathcal{F}(G_{eq,d}, K_d)$, and

$$\|\mathcal{F}(G, H_{T_s} K_d S_{T_s})\|_{\mathcal{L}_2} < \gamma \iff \|\mathcal{F}(G_{eq,d}, K_d)\|_{\mathcal{H}_\infty} < \gamma \quad (4.28)$$

for all $\gamma > \|\underline{D}_{11}\|$. For the numerical calculation of the matrices composing $G_{eq,d}$ shown in eq. (4.46), we perform the following procedure [89]:

- **Step 1:** formulate the linear fractional transformation (LFT) for the generalized plant and ensure that it has the structure shown in eq. (4.21). Choose the sampling interval T_s .
- **Step 2:** compute the $\mathcal{L}_2(0, T_s)$ -induced norm of the system $\|\underline{D}_{11}\|$ as follows [89]:

a. define the following matrix exponential

$$Q(T_s) = \begin{bmatrix} Q_{11}(T_s) & Q_{12}(T_s) \\ Q_{21}(T_s) & Q_{22}(T_s) \end{bmatrix} := e^M \quad (4.29)$$

$$M := \exp \left\{ T_s \begin{bmatrix} -A^T & -C^T C \\ \gamma^{-2} B B^T & A \end{bmatrix} \right\} \quad (4.30)$$

where the partitioning of $Q(T_s)$ is comfortable with that of e^M .

Theorem 4.2.1. [89] For any $\gamma > 0$, γ^2 is an eigenvalue of $\underline{D}\underline{D}^*$ iff $\det[Q_{11}T_s] = 0$.

b. $Q_{11}(T_s)$ is computed over a range of γ .

$$\gamma \in (\gamma_{min}, \gamma_{max}) \quad (4.31)$$

c. $\|\underline{D}_{11}\|$ is the largest γ for which $Q_{11}(T_s)$ has a zero eigenvalue.

• **Step 3:** for $\gamma > \|\underline{D}_{11}\|$, compute the \mathcal{H}_∞ discretization $G_{eq,d}$ for the system in eq. (4.21) as follows:

a. Define:

$$\underline{A} = \begin{bmatrix} A & B_2 \\ 0 & 0 \end{bmatrix}. \quad (4.32)$$

b. Compute J_∞ :

$$J_\infty = \int_0^{T_s} e^{t\underline{A}^T} [C_1 \ D_{12}]^T [C_1 \ D_{12}] e^{t\underline{A}} dt. \quad (4.33)$$

c. Define:

$$E = \begin{bmatrix} -A^T & -C_1^T C_1 \\ B_1 B_1^T / \gamma^2 & A \end{bmatrix} \quad (4.34)$$

$$X = \begin{bmatrix} C_1 & D_{12} \end{bmatrix}^T \begin{bmatrix} 0 & C_1 \end{bmatrix} \quad (4.35)$$

$$Y = \begin{bmatrix} C_1 & 0 \end{bmatrix}^T \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \quad (4.36)$$

in order to compute the matrices P, M, L, Q, N and R from the equation

$$\begin{bmatrix} P & M & L \\ 0 & Q & N \\ 0 & 0 & R \end{bmatrix} = \exp \left\{ T_s \begin{bmatrix} -\underline{A}^T & X & 0 \\ 0 & E & Y \\ 0 & 0 & \underline{A} \end{bmatrix} \right\}. \quad (4.37)$$

Partition Q and R conformably with E and \underline{A} , respectively:

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \quad (4.38)$$

$$R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & I \end{bmatrix}. \quad (4.39)$$

Put $A_d = R_{11}$ and $B_{2d} = R_{12}$.

d. Compute:

$$F = \begin{bmatrix} F_1 & F_2 \end{bmatrix} = \begin{bmatrix} (Q_{11}^{-1})^T & 0 \end{bmatrix} M^T R \quad (4.40)$$

$$A_{dd} = A_d + F_1 \quad (4.41)$$

$$B_{2dd} = B_{2d} + F_2. \quad (4.42)$$

e. Calculate B_{1d} via the Cholesky decomposition:

$$B_{1d}B_{1d}^T = \gamma^2 Q_{21}Q_{11}^{-1}. \quad (4.43)$$

f. Put:

$$J = R^T M \begin{bmatrix} Q_{11}^{-1} & 0 \\ 0 & 0 \end{bmatrix} N - R^T L + J_\infty \quad (4.44)$$

and compute C_{1d} and D_{12d} via Cholesky factorization:

$$[C_{1d} \ D_{12d}]^T [C_{1d} \ D_{12d}] = J. \quad (4.45)$$

g. Finally,

$$G_{eq,d} = \left[\begin{array}{c|cc} A_{dd} & B_{1d} & B_{2dd} \\ \hline C_{1d} & 0 & D_{12d} \\ C_2 & 0 & 0 \end{array} \right]. \quad (4.46)$$

- **Step 4:** we define $T_{eq,d}$ as the closed loop map from w to z with discrete-time plant $G_{eq,d}$ and controller K_d . We now wish to find a γ such that there exists a K_d that achieves internal stability of $T_{eq,d}$ and $\|T_{eq,d}\|_\infty < \gamma$ (discrete-time \mathcal{H}_∞ problem). Once this upper bound on γ (γ_u) has been found, we know that $\|\underline{D}_{11}\| \leq \gamma_{opt} < \gamma_u$, where γ_{opt} is the optimal performance.
- **Step 5:** once a satisfactory γ has been found, we solve the discrete-time \mathcal{H}_∞ problem for $\|T_{eq,d}\|_\infty < \gamma$ for a stabilizing K_d . This controller will also achieve $\|\mathcal{F}(G, H_{T_s} K_d S_{T_s})\|_{\mathcal{L}_2} < \gamma$. The procedure for this is as follows:

- a. Start with $G_{eq,d}$ above and define the artificial continuous-time system $g_c(s)$ using the bilinear transformation:

$$\lambda = \frac{1-s}{1+s}. \quad (4.47)$$

- b. Design an $\mathcal{H}_\infty(\mathbf{C}_+)$ -optimal controller $K_c(s)$ for $g_c(s)$.
- c. Transform $K_c(s)$ back to discrete-time using the bilinear transformation shown in eq. (4.8) to get the final SD controller K_d .

4.3 Computation of the Induced- \mathcal{L}_2 Norm

In this work, we'll be quantifying the performance of the SD control system using the induced- \mathcal{L}_2 norm as defined earlier in this chapter, i.e. $\|T_{wz}\|_{\mathcal{L}_2 \rightarrow \mathcal{L}_2}$. Methods for computing this norm have been described in the literature [84], [89]. MATLAB's *sdhinfnorm* function utilizes the algorithm presented by Bamieh et al. (1992) [84] and this will be compared to the algorithm in Chen et al. (1995) [89] to show that the norm computed is the same for a small example problem.

The method for computing the norm described in Chen et. al. (1995) [89] is as follows:

- **Step 1:** Start with continuous-time system $G(s)$, discrete-time controller K_d and time-delay T_s where $G(s)$ has the form

$$G(s) = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ C_2 & 0 & 0 \end{array} \right]. \quad (4.48)$$

and $K_d(z)$ has the form

$$K_d(z) = \left[\begin{array}{c|c} A_k & B_k \\ \hline C_k & D_k \end{array} \right]. \quad (4.49)$$

and assume internal stability.

- **Step 2**: Compute $\|\underline{D}_{11}\|$ using eq. (4.29) - (4.31).
- **Step 3**: For $\gamma > \|\underline{D}_{11}\|$, compute the \mathcal{H}_∞ discretization shown in eq. (4.46).
- **Step 4**: Form

$$\left[\begin{array}{c|c} A_{cld} & B_{cld} \\ \hline C_{cld} & 0 \end{array} \right] = \left[\begin{array}{cc|c} A_{dd} + B_{2dd}D_kC_2 & B_{2dd}C_k & B_{1d} \\ B_kC_2 & A_k & 0 \\ \hline C_{1d} + D_{12d}D_kC_2 & D_{12d}C_k & 0 \end{array} \right] \quad (4.50)$$

then take a minimal realization and compute the symplectic pair

$$(S_l, S_r) = \left(\left[\begin{array}{cc} A_{cld} & 0 \\ -C'_{cld}C_{cld}/\gamma & I \end{array} \right], \left[\begin{array}{cc} A_{cld} & 0 \\ -C'_{cld}C_{cld}/\gamma & I \end{array} \right] \right) \quad (4.51)$$

in order to compute

$$\gamma_{max} := \max\{\gamma : (S_l, S_r) \text{ has an eigenvalue on } \partial\mathbf{D}\} \quad (4.52)$$

A bisection search can be used in Steps 3 and 4 to compute γ_{max} .

- **Step 5**:

$$\|T_{wz}\|_{\mathcal{L}_2 \rightarrow \mathcal{L}_2} = \max\{\|\underline{D}_{11}\|, \gamma_{max}\} \quad (4.53)$$

Example:

We replicate example 13.7.2 from Chen et al. (1995) [89]. Consider the closed loop system shown in Fig. 4.7. Let $P = \frac{1}{s}$ be the plant and $K_d = 1$ be the controller. We wish to compute the induced- \mathcal{L}_2 norm $\|T_{wz}\|$ as we vary T_s . The generalized plant is then given by [89]:

$$G = \left[\begin{array}{c|c} P & -P \\ \hline P & -P \end{array} \right] = \left[\begin{array}{c|cc} 0 & 1 & -1 \\ \hline 1 & 0 & 0 \\ 1 & 0 & 0 \end{array} \right] \quad (4.54)$$

We utilize both the method described in this chapter as well as the MATLAB function *sdhinfnorm* to compute and plot the norm as a function of T_s . This is shown in Fig. 4.8 where *sdl2norm* is the method presented in this section. As evident from the figure, both methods yield the same norm.

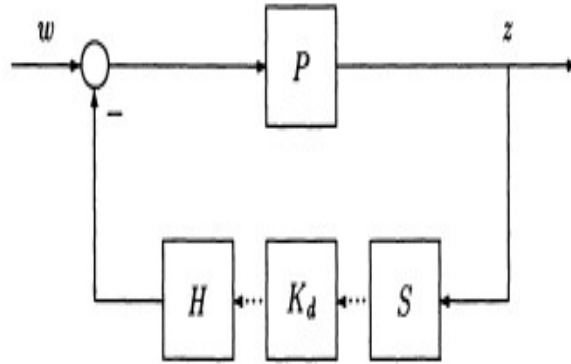


Figure 4.7: SD System [89]

Example:

We consider the following problem

- Plant - $P = \frac{1}{s+0.01}$ an integrator

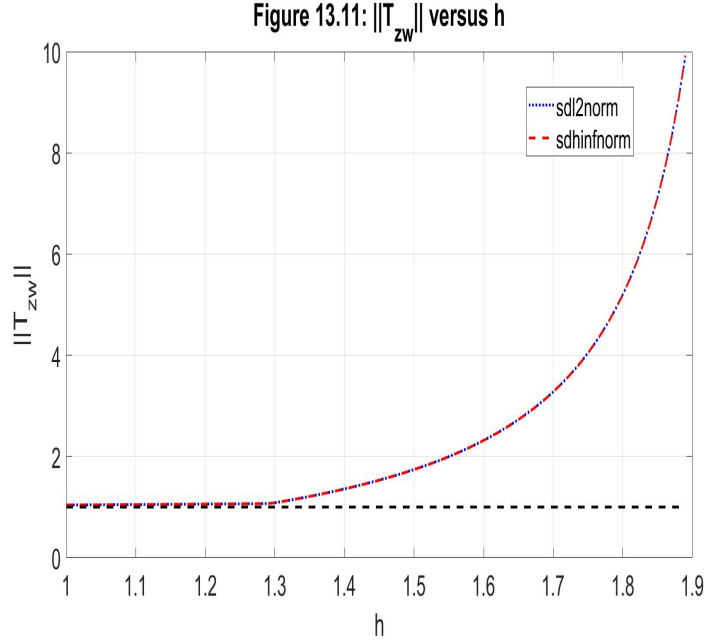


Figure 4.8: $\|T_{wz}\|$ vs. $h = T_s$ [89]

- AAF - $F = \frac{1}{\left(\frac{T_s}{\pi}\right)^{s+1}}$ is the anti-aliasing filter
- W - $W = \frac{1}{\left(\left(\frac{5T_s}{\pi}\right)^{s+1}\right)^2}$ is the weight on sensitivity

In this problem, we add a Pade approximation of the time delay to the plant before discretizing the controller using the Tustin/bilinear and step-invariant/ZOH transformations. The results are compared to the γ obtained from the “lifting” based controller design method in Fig. 4.10. The system is shown in Fig. 4.9.

4.4 Weighting Function

For designing the SD controller (K_d), the error ($T_{r \rightarrow e}$) was weighted using the following WEIGHTING FUNCTION while solving the optimization

$$W_{error} = \frac{1}{M_s} \left[\frac{s + M_s \omega_b}{s + \epsilon \omega_b} \right] \left[\frac{a_w}{s + a_w} \right] \quad (4.55)$$

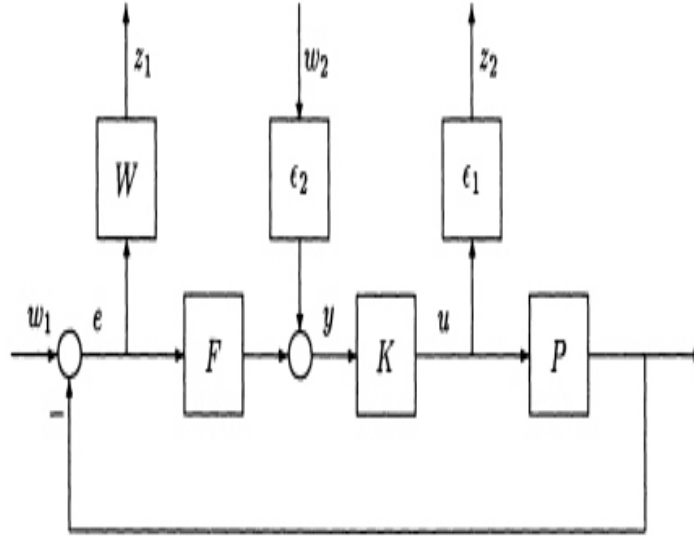


Figure 4.9: Closed Loop System [89]

Parameters chosen are:

- $\omega_b = 1000$ rad/s (Closed Loop Bandwidth)
- $\epsilon = 0.1$ (Low Frequency Gain of $T_{r \rightarrow e}$)
- $M_s = 5$ (High Frequency Gain of $T_{r \rightarrow e}$)
- $a_w = 10M_s\omega_b = 50,000$ (Roll-off Frequency)

$$W_1 \approx \begin{cases} \frac{1}{\epsilon}, & 0 \leq |s| \ll \epsilon\omega_b \\ \frac{\omega_b}{s}, & \epsilon\omega_b \ll |s| \ll M_s\omega_b \\ \frac{1}{M_s}, & M_s\omega_b \ll |s| \ll a_w \end{cases} \quad (4.56)$$

4.5 “Lifted” Plant (\mathcal{H}^∞ Discretization)

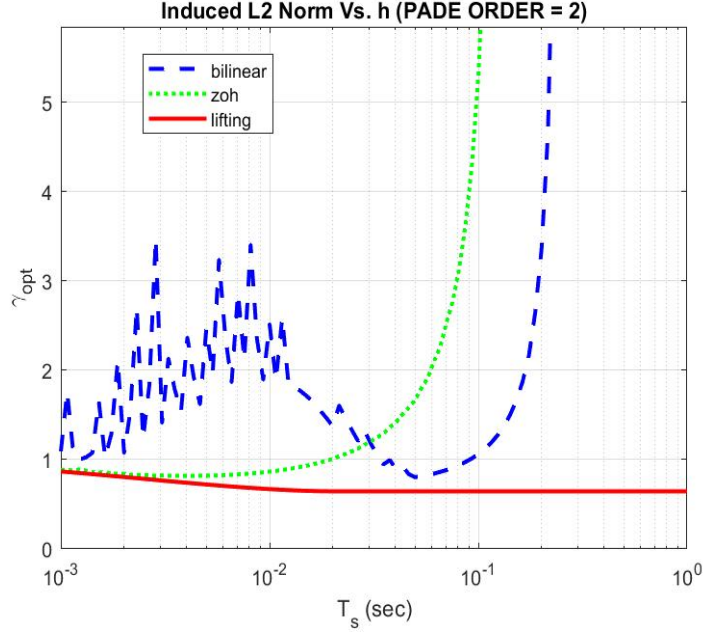


Figure 4.10: $\|T_{wz}\|$ vs. $h = T_s$

	$h = 10 \mu s$	$50 \mu s$	$100 \mu s$
$\ T_o\ _{\mathcal{L}_2 \rightarrow \mathcal{L}_2}$	0.9789	0.9806	0.9983
$\ KS_o\ _{\mathcal{L}_2 \rightarrow \mathcal{L}_2}$	0.3978	0.3985	0.4056
$\ S_o\ _{\mathcal{L}_2 \rightarrow \mathcal{L}_2}$	0.1498	0.2899	0.6263

Table 4.1: Norms

4.6 Trade-off

Figs. 4.15-4.18 display the performance of the three methods and the analog design. As seen from Figs. 4.15 and 4.16, $\|\mathcal{F}(G, H_{T_s} K_d S_{T_s})\|_{\mathcal{L}_2}$ for the indirect design increases drastically with T_s . The degradation in norm is less severe for the direct design, but both traditional methods are inferior to the lifting-based method, which achieves near-analog performance for all sampling periods T_s . For $T_s \rightarrow 0$, $\|\mathcal{F}(G, H_{T_s} K_d S_{T_s})\|_{\mathcal{L}_2}$ for all three digital controllers converge to $\|\mathcal{F}(G, K_c)\|_{\mathcal{H}_\infty}$. That is, analog performance is recovered as the sampling period tends to zero.

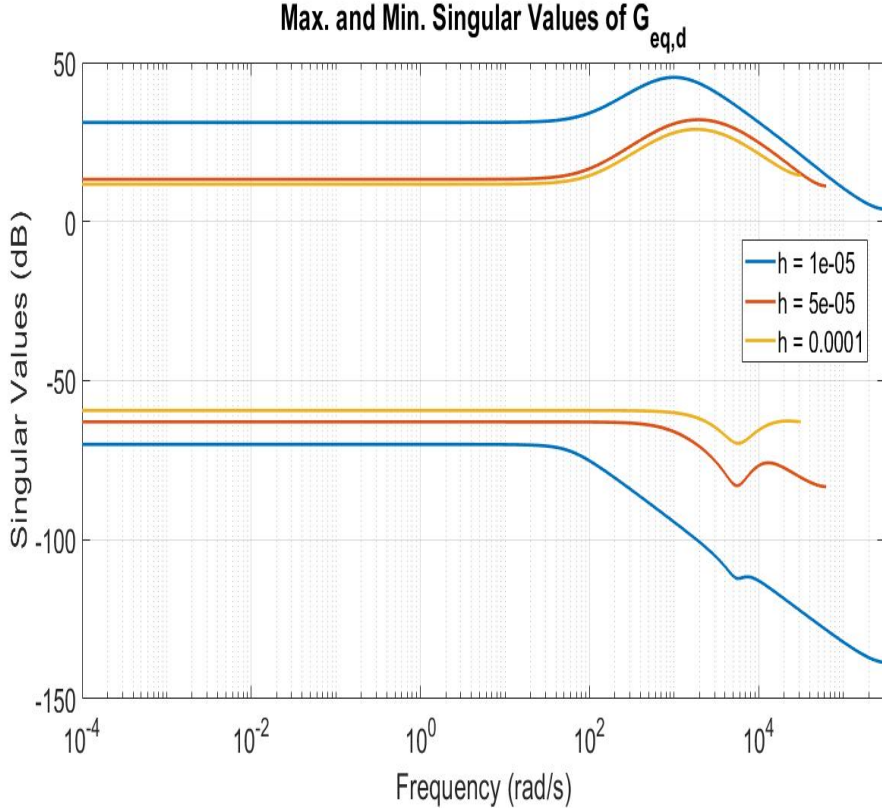


Figure 4.11: Lifted Generalized Plant ($G_{eq,d}$)

However, $T_s = \frac{1}{f_s}$. Thus, from eq. (3.49), the switching loss increases as $T_s \rightarrow 0$. This is shown by considering two cases, $f_s = 4.5$ kHz and $f_s = 100$ kHz. The closed-loop stability and power loss of the design methods are shown in Tab. 4.2. The discrete control system is stable for high switching frequencies like 100 kHz (Fig. 4.17), but they have a much higher power loss. If we wish to lower the loss, we would want to use $f_s = 4.5$ kHz, but the indirect method would yield an unstable system; this can be remedied by using either direct digital design or a lifting-based design (Fig. 4.17).

4.7 Summary

This chapter describes the controller design for the buck converter and compares 3 methods for digital controller design. The traditional method of designing an analog

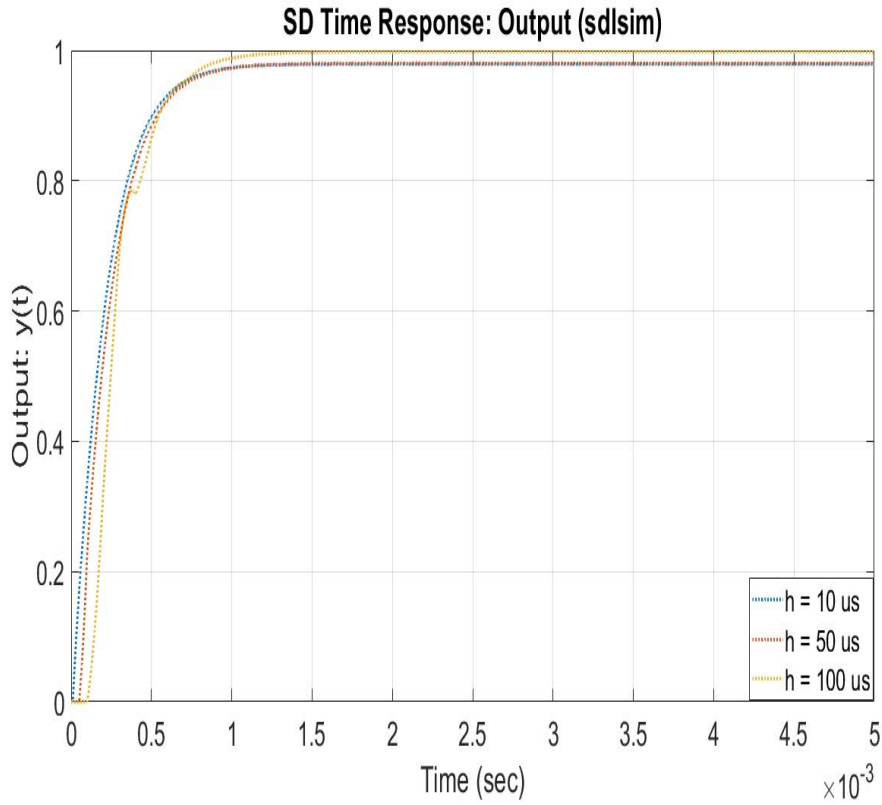


Figure 4.12: Output (SD System)

controller and then discretizing it is found to give the worst performance as the sampling time delay increases. The “lifting” method accounts for the effect of the delay. This is thus found to provide acceptable controllers even as the time delay increases.

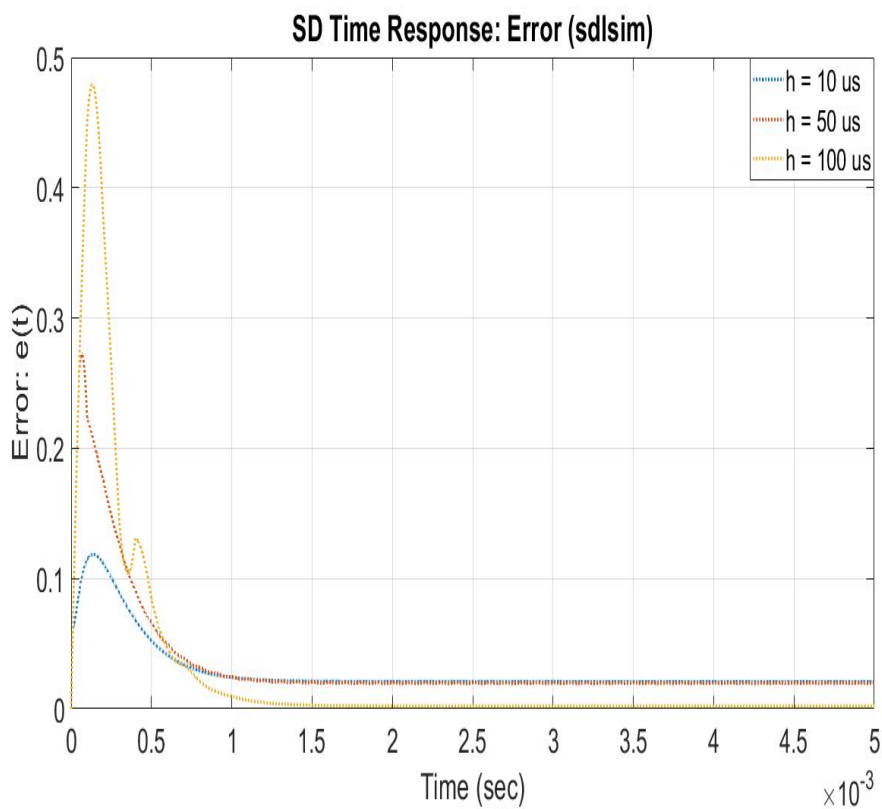


Figure 4.13: Error (SD System)

f_s (kHz)	Stability (Indirect-Design)	Stability (Direct-Design, Lifting-Based)	Loss (in W)
4.5	UNSTABLE	STABLE	0.0645
100	STABLE	STABLE	1.4338

Table 4.2: Trade-off: Stability vs. Power Loss

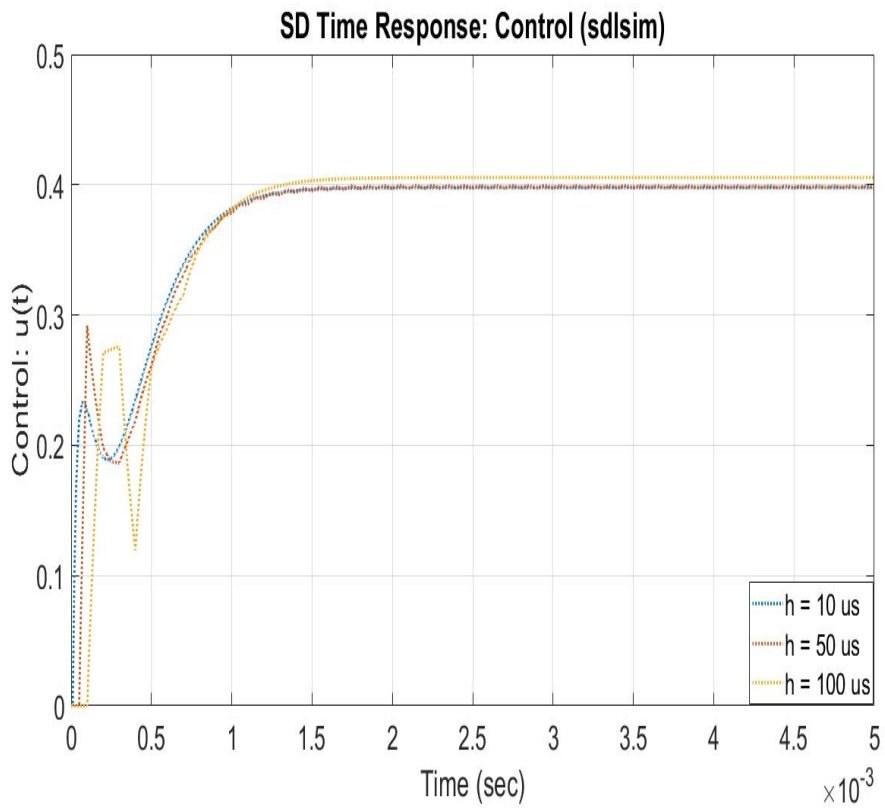


Figure 4.14: Control (SD System)

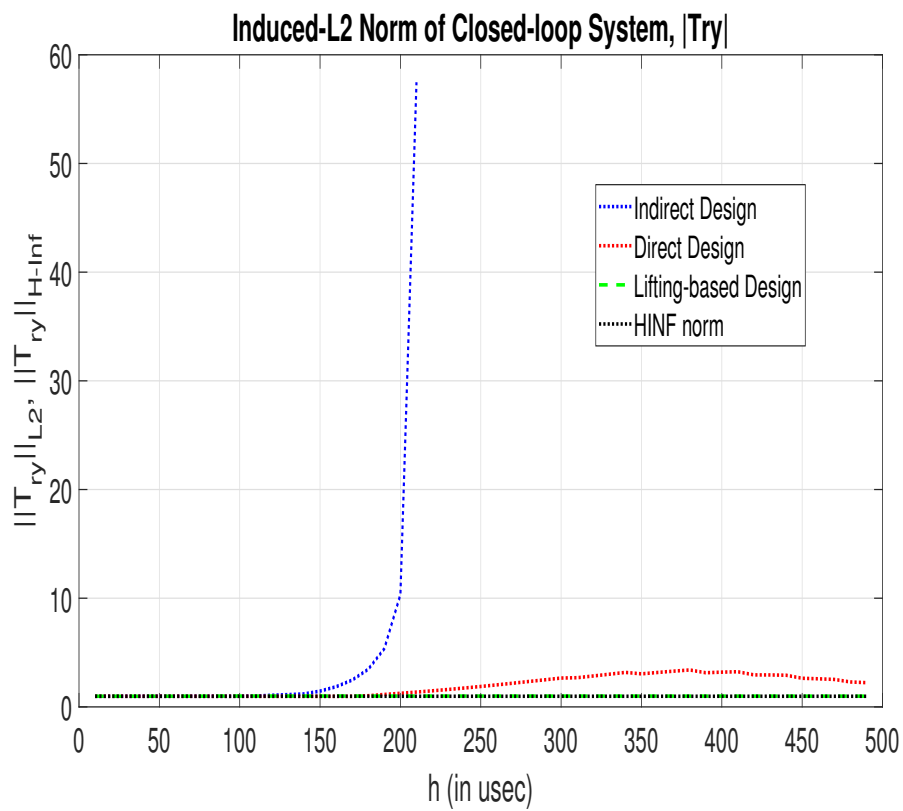


Figure 4.15: Effect of Variation in h ($= T_s$) on $\|T_{ry}\|_{\mathcal{L}_2}$

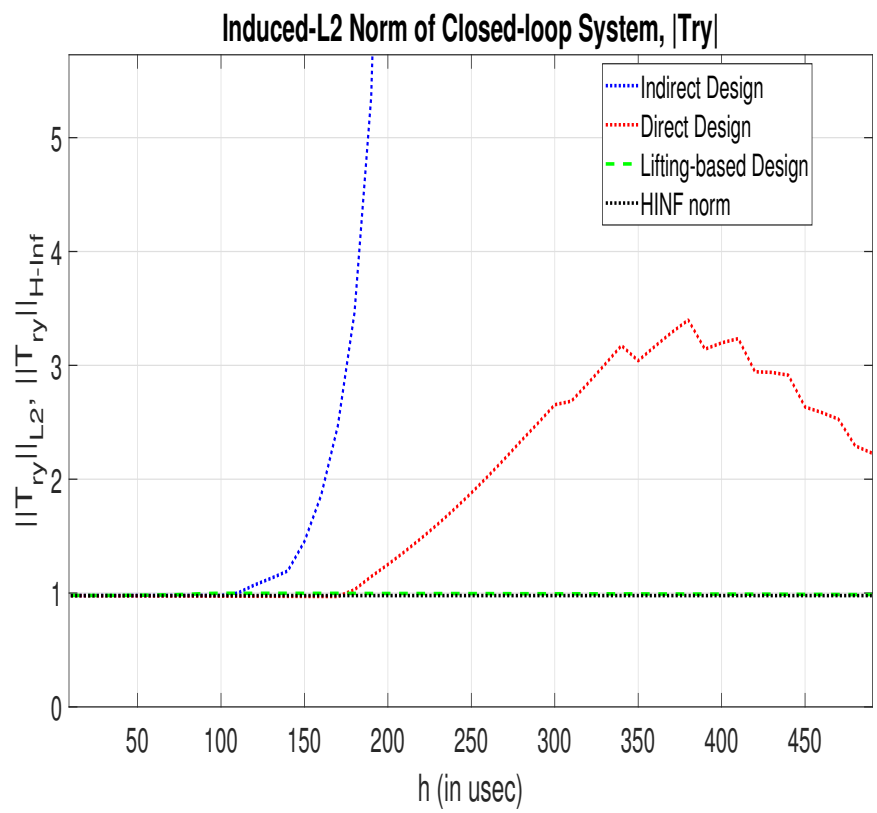


Figure 4.16: Effect of Variation in h ($= T_s$) on $\|T_{ry}\|_{\mathcal{L}_2}$ (Zoomed In)

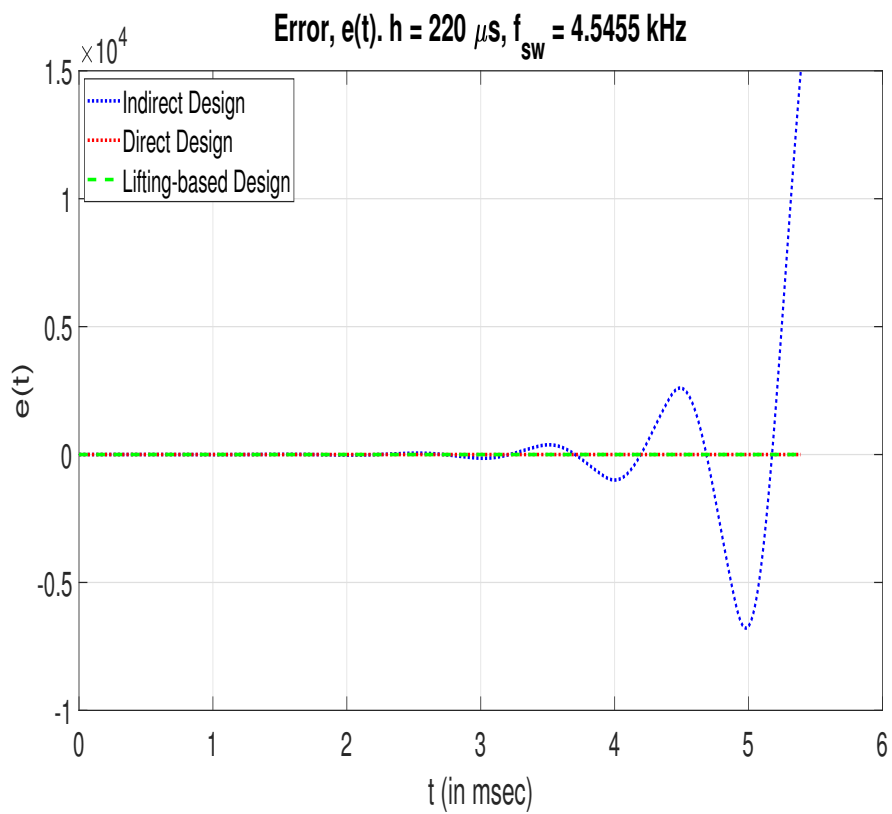


Figure 4.17: Error Time Response ($f_s = 4.5 \text{ kHz}$, $T_s = h = 220 \mu\text{s}$)

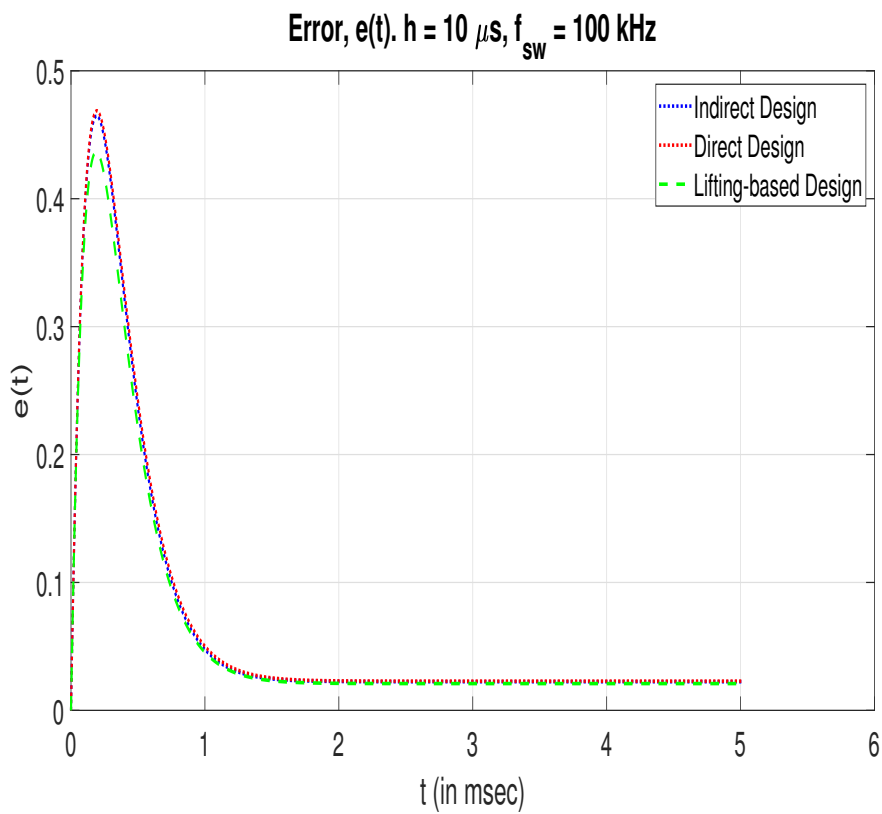


Figure 4.18: Error Time Response ($f_s = 100 \text{ kHz}$, $T_s = h = 10 \mu\text{s}$)

MODELING AND CONTROL OF DC-AC INVERTER WITH LCL FILTER

5.1 3- ϕ DC-AC Inverter Averaged Model

A balanced 3- ϕ four-wire inverter (Fig. 5.1) is equivalent to three single phase (1- ϕ) inverters. For simplicity, we study the 1- ϕ model of the inverter using an LCL filter (per phase) in Fig. 5.1. An average model can be used since we have a suitably high switching frequency. R_1 , R_2 and R_f are the negligible parasitic resistances in series with the two inductances (L_1 , L_2) and the capacitor (C_f), respectively (not shown in the figure). The currents passing through the inductors (I_1 and I_2) as well as the voltage across the capacitor (V_c) are taken as states of the model ($x = \begin{bmatrix} I_1 & I_2 & V_c \end{bmatrix}^T$), and output is $y = I_2$. We will consider the input u to be the control signal to the PWM connected to the switches, where $u \in [-1, 1]$. The state space representation will be: $\dot{x} = Ax + Bu$, $y = Cx + Du$, where

$$A = \begin{bmatrix} \left(-\frac{R_f+R_1}{L_1}\right) & \left(\frac{R_f}{L_1}\right) & \left(-\frac{1}{L_1}\right) \\ \left(\frac{R_f}{L_2}\right) & \left(-\frac{R_f+R_2}{L_2}\right) & \left(\frac{1}{L_2}\right) \\ \left(\frac{1}{C_f}\right) & \left(-\frac{1}{C_f}\right) & 0 \end{bmatrix} \quad (5.1)$$

$$B = \begin{bmatrix} \frac{1}{L_1} \\ 0 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, D = 0 \quad (5.2)$$

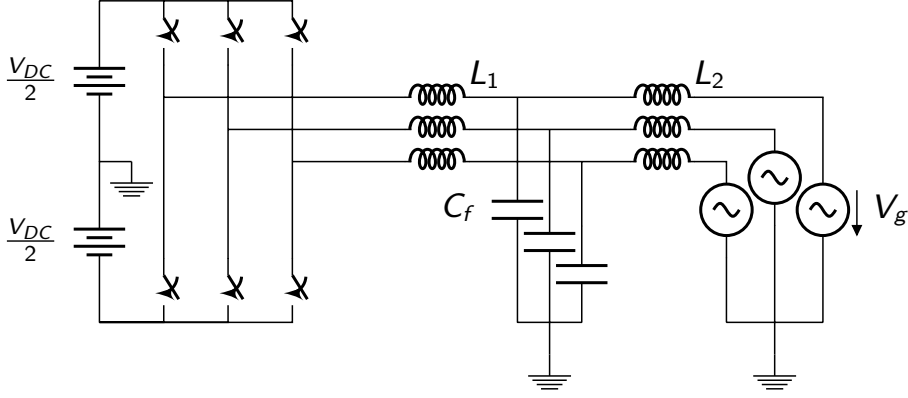


Figure 5.1: Grid-Tie Inverter (LCL Filter) Circuit

The corresponding transfer function is

$$H_{LCL} = \frac{sR_f C_f + 1}{s^3 L_2 L_1 C_f + s^2 C_f [L_2 (R_f + R_1) + L_1 (R_f + R_2)] + s [L_2 + L_1 + C_f (R_f R_2 + R_f R_1 + R_2 R_1)] + R_2 + R_1}$$

Modeling the inverter switching circuit as a gain $\frac{V_{DC}}{2}$ [64], plant transfer function will be $P = \frac{V_{dc}}{2} H_{LCL}$. The plant parameters used are Inverter-side inductor $L_1 = 0.56734mH$, Grid-side inductor $L_2 = 0.56734mH$, Capacitor $C_f = 15.351\mu F$, Parasitic Resistance $R_1 = R_2 = R_f = 1\mu\Omega$, DC Link Voltage $V_{DC} = 400V$, Switching frequency $f_{sw} = 15kHz$.

Here, a negligible amount of parasitic resistance is considered to be in series with the LCL filter components. This is the worst case condition without passive damping [33] as evident from the plant poles in Tab. 5.1, where one pole nearly at origin, and two very lightly damped poles. As such, this makes it a challenging control problem since the resonance has to be damped. An inner-outer control structure is used to address this problem [28, 29, 30, 31, 32, 33, 34, 35, 36, 37].

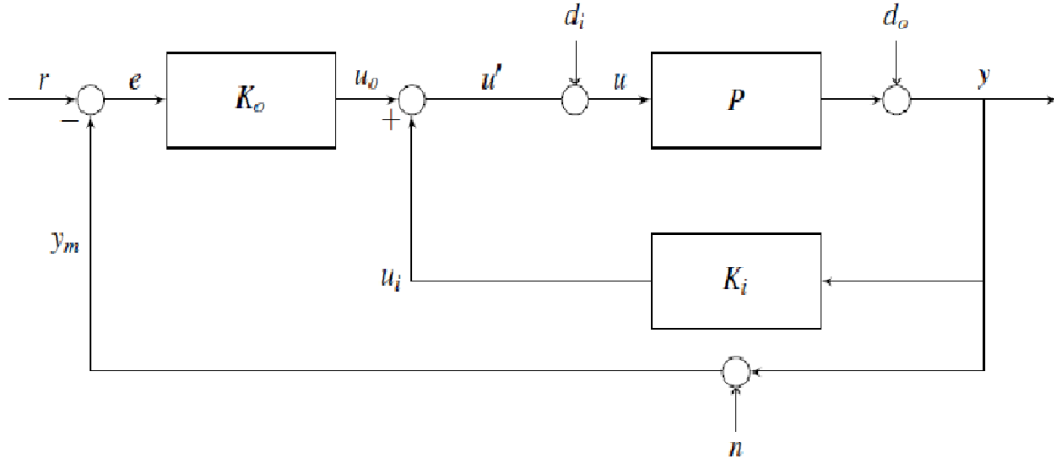


Figure 5.2: Inner-Outer Loop Control Structure

Table 5.1: Plant Poles and Zeros

Pole	Damping	Frequency
-1.76×10^{-3}	1	1.76×10^{-3}
$-0.0026 \pm j1.52 \times 10^4$	1.74×10^{-7}	1.52×10^4
Zero	Damping	Frequency
-6.51×10^{10}	1	6.51×10^{10}

5.2 Traditional Inner-Outer Control with a PR Controller

The hierarchical inner-outer loop control structure used in this paper is shown in Fig. 5.2. Here, P is the plant described in eq. (5.1-5.2), K_i and K_o are inner- and outer-loop controllers resp. r , d_i , d_o , n_i , n_o are exogenous input channels, and y is output channel. For our system, the output is grid current.

As seen in Tab. 5.1, the plant has one pole near the origin (that corresponds to approx. L-filter behavior, $\frac{1}{sL}$), and two very lightly damped poles ($\zeta_p = 1.74 \times 10^{-7}$). The inner-loop is used to move the high frequency, lightly damped poles to more

favorable locations without significantly affecting the low frequency dynamics. The outer-loop is designed for shaping low frequency behavior.

For this hierarchical structure, it is observed that 3 sets of properties are obtained by breaking the loop at the error (e), the control/plant input (u), and at inner-loop sensor noise (n_i). We analyse and design controllers for shaping these control maps/properties. It is to be noted that if same sensor is used for both inner as well as outer loop, then we only have two sets of distinct loop breaking points. These properties are discussed below.

Properties at error (e): Consider the loop broken at error signal (e). Let $P_{mod}(s) : u_p \rightarrow y = \frac{P}{1+PK_i}$. Then, open-loop transfer function is $L_e = P_{mod}K_o$, and the corresponding closed-loop sensitivity and complementary sensitivity are, $S_e : r \rightarrow e = \left[\frac{1}{1+L_e} \right]$, $T_e : r \rightarrow y = \left[\frac{L_e}{1+L_e} \right]$, $T_{ru} : r \rightarrow u = \left[\frac{-K_o}{1+PK_o+PK_i} \right]$.

5.3 Properties at the input and output

Consider the loop broken at controls (u). Let $K_{sum} = -(K_i + K_o)$. Then, open-loop transfer function is $L_c = K_{sum}P$, and the corresponding closed-loop sensitivity and complementary sensitivity are, $S_c : d_i \rightarrow u_p = \frac{1}{1-L_c}$, $T_c : d_i \rightarrow u = \frac{L_c}{1-L_c}$, $T_{d_iy} : d_i \rightarrow y = \frac{P}{1-L_c}$. We study the properties at error which is widely addressed in literature [33, 35], as well as the not so widely addressed properties at controls. We show how the controllers (esp. inner-loop) can be designed to obtain desirable properties at multiple loop-breaking points. Obtaining desirable properties at controls is of importance to be robust to disturbance (d_i) and/or modeling uncertainty at input. We note that though the plant is single-input single-output (SISO), studying the properties at multiple loop-breaking points becomes relevant mainly because of the following reasons: 1) the inner-outer structure we use inherently results in properties

at these points not being identical, 2) the map T_{diy} is not captured by breaking the loop at error.

Inner-Outer Loop Controller Structure: The structures of the controllers K_o and K_i in Fig. 5.2 used are:

$$K_o = g \left[\frac{s^2 + 2\zeta_{n_o}\omega_{n_o}s + \omega_{n_o}^2}{s^2 + 2\zeta_{d_o}\omega_{d_o}s + \omega_{d_o}^2} \right] \quad (5.3)$$

$$K_i = - \left[\frac{k_i s}{s + p_i} \right] \quad (5.4)$$

K_o is known as proportional-resonant (PR) controller [33, 35] with $\omega_{n_o} = \omega_{d_o} = 2\pi 60$, which helps with obtaining good low frequency (upto >60Hz) behavior.

Nominal Control Design: Using standard PR controller design techniques [33, 35], we choose $\zeta_{n_o} = 0.84$ and $\zeta_{d_o} = 1.33 \times 10^{-9}$ to draw the controller poles to favorable locations and $g = 0.0035$ to ensure sufficient loop gain at 59.3 Hz which is the worse case frequency dip as stated in the IEEE standard 1547 [81]. K_i is designed to obtain desirable high-frequency behavior [36]. Using standard design techniques, we obtain $k_i = 0.0579$ and $p_i = 18 \times 10^3$. Note that, this nominal design is done using standard techniques used in literature. As will be shown below, a major drawback with this is, though we achieve good properties at error, we see poor properties at controls/input. We show how we can address properties at both loop-breaking points.

Analysis of Nominal Controller: Using standard design procedures given in literature [33, 35, 36], we obtain the closed-loop properties shown in Tab. 4.1. The inner-loop controller K_i is designed based on desired damping of the high-frequency plant poles. But, from Fig. 5.3, zero of resulting open-loop transfer function L_c corresponding to this design is in right-half plane (RHP), close to origin. Note that, in Fig. 5.3, we plot the behavior of zeros of $L_c = K_{sum}P$ as we vary k_i . We also observe that for nominal design, the closed-loop properties at controls: S_c and T_c

have high peak values, though we have good properties at error: S_e and T_e .

Effect of Inner-Loop Controller on Closed-Loop Properties: Here we study the behavior of zeros of L_c in Fig. 5.3, following which we study the rationale behind selecting a value of k_i for obtaining reasonable properties simultaneously at both loop-breaking points.

$$K_{sum} = -(K_i + K_o) \quad (5.5)$$

$$= \frac{k_i s}{s + p_i} - g \left[\frac{s^2 + 2\zeta_{n_o} \omega_{n_o} s + \omega_{n_o}^2}{s^2 + 2\zeta_{d_o} \omega_{d_o} s + \omega_{d_o}^2} \right] \quad (5.6)$$

Eq. (5.6) shows that the zeros of K_{sum} (and hence L_c) are the roots of the equation, shown in Fig. 5.3.

$$\begin{aligned} & k_i s(s^2 + 2\zeta_{d_o} \omega_{d_o} s + \omega_{d_o}^2) - \\ & g(s^2 + 2\zeta_{n_o} \omega_{n_o} s + \omega_{n_o}^2)(s + p_i) = 0 \\ \implies & [-g(s^2 + 2\zeta_{n_o} \omega_{n_o} s + \omega_{n_o}^2)(s + p_i)] + \\ & k_i [s(s^2 + 2\zeta_{d_o} \omega_{d_o} s + \omega_{d_o}^2)] = 0 \end{aligned}$$

By plotting the root locus of $H_z(s)$ below, we can study the movement of zeros of L_i as k_i is varied.

$$H_z(s) = \frac{s(s^2 + 2\zeta_{d_o} \omega_{d_o} s + \omega_{d_o}^2)}{-g(s^2 + 2\zeta_{n_o} \omega_{n_o} s + \omega_{n_o}^2)(s + p_i)} \quad (5.7)$$

When we choose a value of k_i that moves the RHP zero of L_c to less problematic region (away from origin in this case), the properties at controls improve, as seen in Tab. 4.1. The rationale behind selecting a value of k_i is now discussed. Typically in the literature [36] L_e is shaped by designing $K_i(s)$ to damp the plant poles at resonance frequency and then designing $K_o(s)$ to shape L_e near the grid frequency ($2\pi 60$ rad/s).

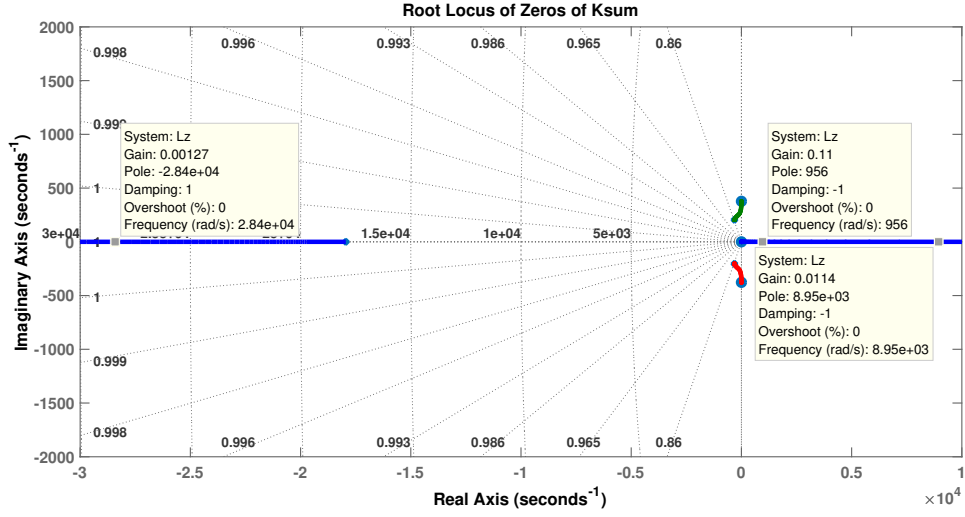


Figure 5.3: Root Locus of $H_z(s)$ (Eqn. 5.7)

This usually results in acceptable $|S_e|_\infty$ and $|T_e|_\infty$, but may result in unacceptable $|S_c|_\infty$ and $|T_c|_\infty$ as shown in Design 2 of Tab. 4.1 To obtain good properties at both error and controls, we select k_i as mentioned below. We see that as we reduce k_i , the RHP zero of L_c moves away from the origin which makes it more favorable. However, there is a tradeoff. This root locus in Fig. 5.4 shows the movement poles of P_{mod} as k_i increases. Note that lightly damped pole moves towards the origin. At low values of k_i , the pole is nearly at the origin. This results in bad low frequency behavior. Hence, we pick a value of k_i that helps obtain reasonable properties at both loop-breaking points. Below, we show how the root-locus in Fig. 5.4 was obtained. Noting that the plant in Eqn. 5.3 with the assumed parameters may be written as

$$P(s) = \frac{621(s + 6.5 \times 10^{10})}{(s + 0.002)(s^2 + 0.005s + 2.3 \times 10^8)} \quad (5.8)$$

$$= \frac{g_p(s + z_\infty)}{(s + p_0)(s^2 + 2\zeta_p\omega_{n_p}s + \omega_{n_p}^2)} \quad (5.9)$$

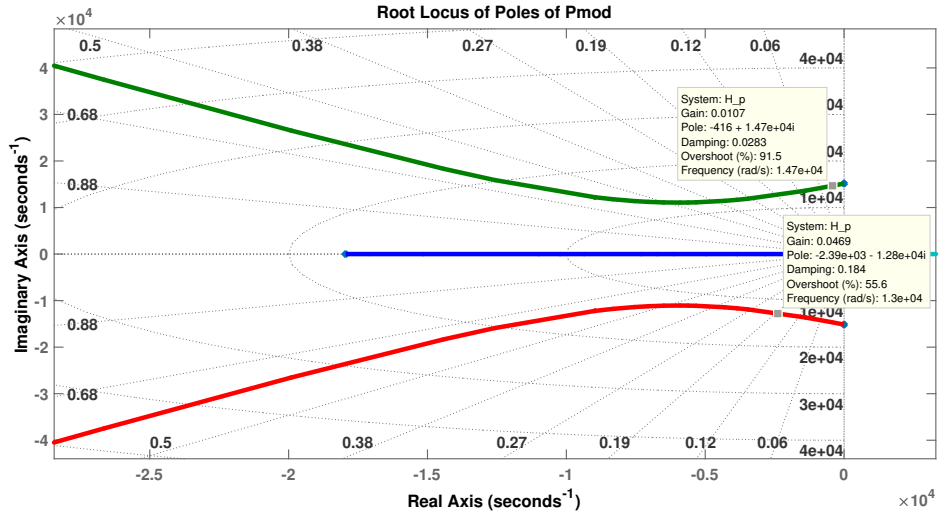


Figure 5.4: Root Locus of $H_p(s)$ (Eqn. 5.12)

The modified plant obtained is

$$\begin{aligned}
 P_{mod} &= \frac{P}{1 + PK_i} \\
 &= \frac{g_p(s + z_\infty)(s + p_i)}{(s + p_0)(s^2 + 2\zeta_p\omega_{np}s + \omega_{np}^2)(s + p_i) - g_p(s + z_\infty)(k_i s)}
 \end{aligned} \tag{5.10}$$

The poles of P_{mod} are the solution of

$$k_i [-g_p s(s + z_\infty)] + [(s + p_0)(s + p_i)(s^2 + 2\zeta_p\omega_{gp}s + \omega_{gp}^2)] = 0 \tag{5.11}$$

Behavior of the poles of P_{mod} with k_i can be studied by plotting the root locus of the following transfer function

$$H_p(s) = \frac{-g_p s(s + z_\infty)}{(s + p_0)(s + p_i)(s^2 + 2\zeta_p\omega_{gp}s + \omega_{gp}^2)} \tag{5.12}$$

To choose a value of k_i that results in reasonable properties at both loop-breaking points, we make the following observation: The following relations relate the margins of L_e (L_c) to $|S_e|_\infty$ ($|S_c|_\infty$) and $|T_e|_\infty$ ($|T_c|_\infty$) [82]:

$$|S|_\infty \geq \max \left\{ \frac{\downarrow GM}{1 - \downarrow GM}, \frac{1}{2 \sin(\frac{PM}{2})}, \frac{\uparrow GM}{\uparrow GM - 1} \right\} \quad (5.13)$$

Similarly, $|T|_\infty \geq \max \left\{ \frac{1}{1 - \downarrow GM}, \frac{1}{2 \sin(\frac{PM}{2})}, \frac{1}{\uparrow GM - 1} \right\}$. From Fig. 5.6, we see that for some value of k_i , the quantity $\frac{1}{2 \sin(\frac{PM}{2})}$ is approximately a tight bound. Note that, here we have plotted the quantity corresponding to third (high frequency) phase margin of L_c . Based on this, we pick a new value (0.0194) for k_i . We can observe from Tab. 5.2 that the new design results in improved properties at controls, while trading off properties at error.

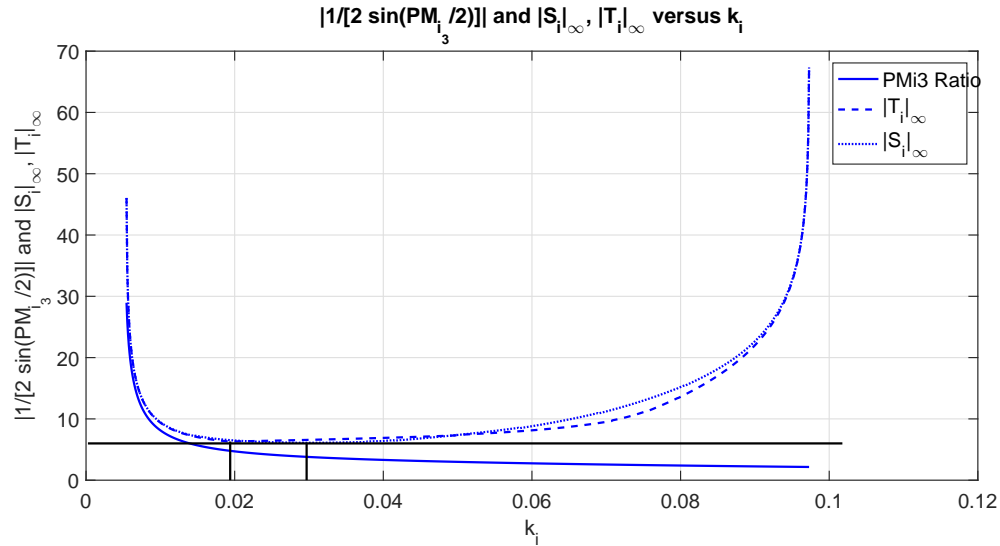


Figure 5.5: $|S_c|_\infty, |T_c|_\infty, \frac{1}{2 \sin \frac{PM_{c3}}{2}}$ v.S. k_i

Table 5.2: Critical Control-Relevant Properties (in dB)

k_i	Design	$ S_i _\infty$	$ T_i _\infty$	$ S_o _\infty$	$ T_o _\infty$
0.0579	1	9.9	8.9	1.1	3.4
0.0194	2	8.6	8.4	5.6	3.7

5.4 Inner-Outer Control with Lag Network

In order to further improve the closed-loop properties obtained earlier, we consider improving $\frac{1}{2\sin(\frac{PM}{2})}$. The idea here is to improve the bound $\frac{1}{2\sin(\frac{PM}{2})}$ (corresponding to third PM of L_c), to obtain better $|S_c|_\infty$ and $|T_c|_\infty$. This is done by adding phase lag [74] in series with L_c . The new structures of the controllers are:

$$K_o = g \left[\frac{s^2 + 2\zeta_{n_o}\omega_{n_o}s + \omega_{n_o}^2}{s^2 + 2\zeta_{d_o}\omega_{d_o}s + \omega_{d_o}^2} \right] \sqrt{\frac{z}{p}} \left[\frac{s+p}{s+z} \right] \quad (5.14)$$

$$K_i = - \left[\frac{k_i s}{s+p_i} \right] \sqrt{\frac{z}{p}} \left[\frac{s+p}{s+z} \right] \quad (5.15)$$

The properties obtained using above technique is shown in Tab. 5.3. Corresponding L_c is shown in Fig. 5.6. It can be seen that by introducing lag term to improve the third phase margin of L_c , the properties were improved, esp. compared to those in Tab. 4.1.

Table 5.3: Critical Control-Relevant Properties

Lag	$ S_o _\infty$	$ S_i _\infty$	$ T_o _\infty$	$ T_i _\infty$	BW L_o
15°	3.82	4.64	3.05	4.26	1241.2
20°	3.47	3.77	2.87	3.96	1365.1
25°	3.18	3.98	2.71	3.86	1514.2

Figs. 5.7-?? show the closed-loop properties corresponding to following three (3) designs.

- Design 1: Nominal Design [33, 36].

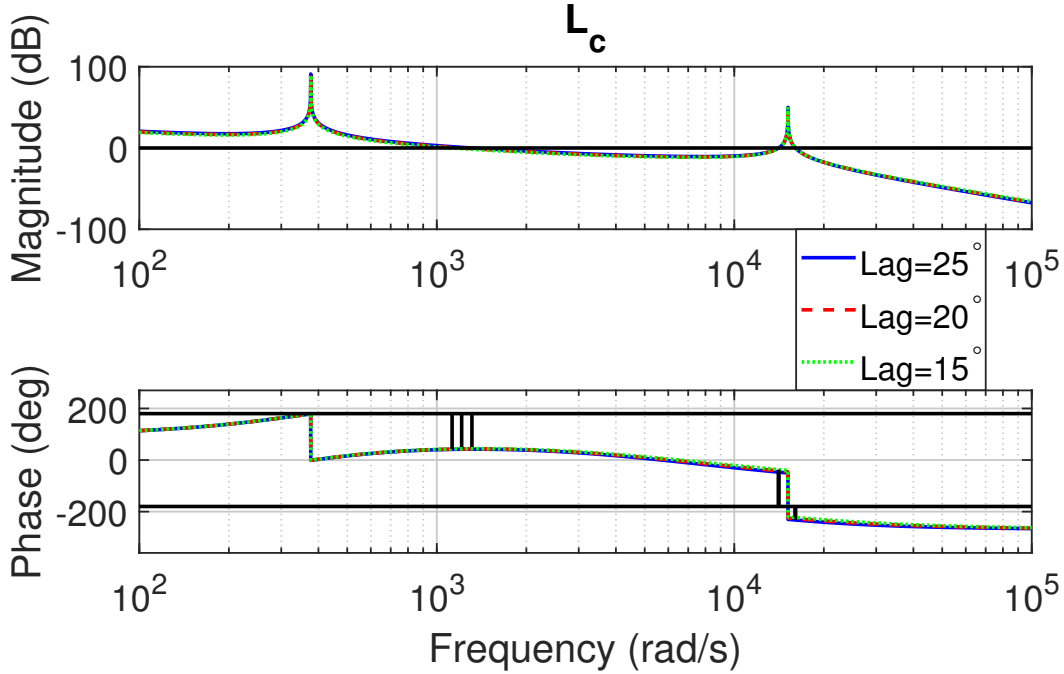


Figure 5.6: Open Loop (Broken at the Input)

- Design 2: Same structure as design 1, but k_i is chosen based on the minimum $|T_c|_\infty$ obtained in Fig. 5.5
- Design 3: With additional 25° lag in series with design 2 to improve third PM of L_c .

We see that Designs 2 and 3 result in an improvement over design 1 for nearly all the frequencies except near the resonant frequency of the plant.

5.5 Summary

To summarize, the tradeoffs involved in designing a hierarchical inner-outer control of active damping of LCL filter resonance of a grid-tied inverter was studied. A novel inner-outer control design technique can help obtain reasonable properties simultaneously at the error and plant input/controls was provided.

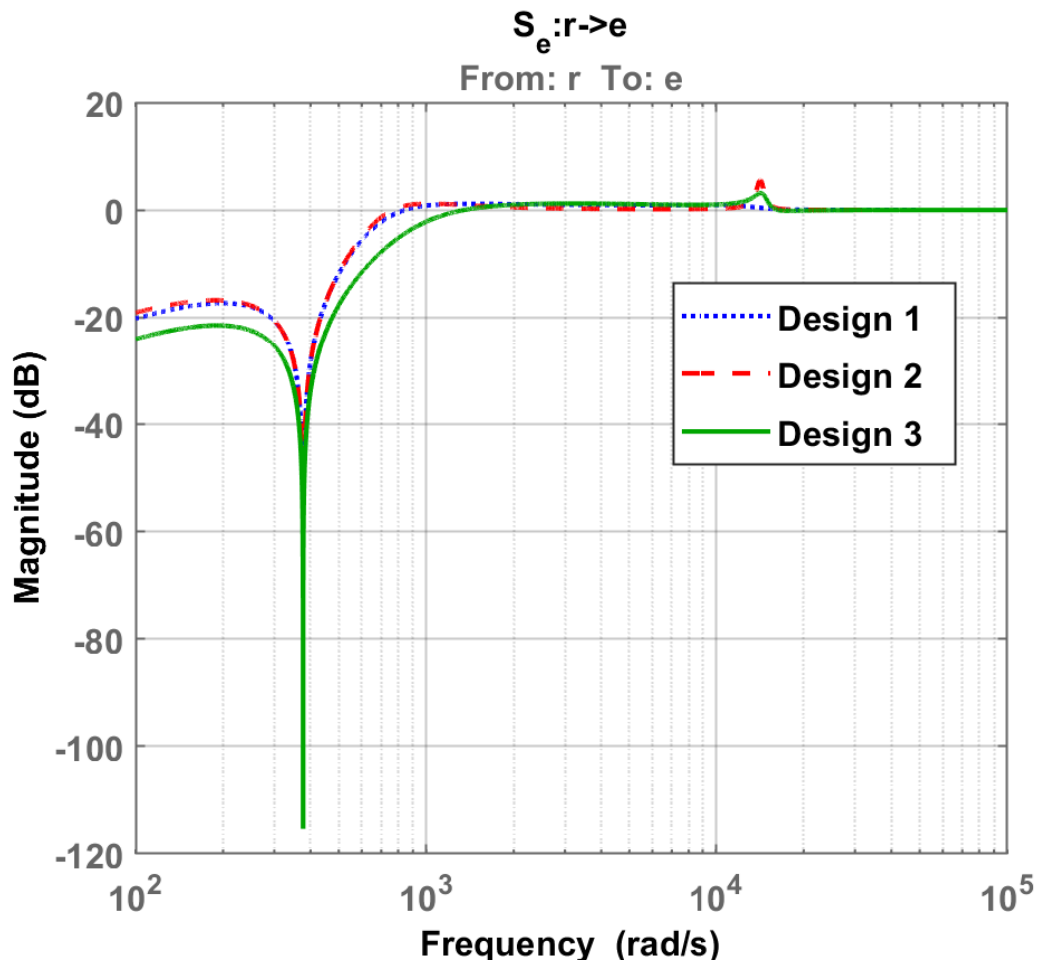


Figure 5.7: $S_e : r \rightarrow e$

SUMMARY and FUTURE DIRECTIONS

This dissertation has studied plant and controller design for 2 classes of power converters which have widespread applications in the field of power electronics. The focus has been on quantifying the performance of design methods for both the systems/plants and the controllers. This quantification is used to ascertain when a particular method is acceptable, and propose alternative methods when it isn't. Frequency responses of 2nd, 4th and 6th order filters are compared to determine when each of them is useful. An alternative to the traditional filter design equations is presented using Fourier analysis and the filter state-space model. This novel method does not rely on the *small ripple approximation* like the traditional one. 3 techniques for discrete-time digital controller design were compared for the buck converter and their performance was evaluated using the induced- \mathcal{L}_2 norm of the closed loop system. Finally, the inner-outer loop (active damping) control of DC-AC inverters with LCL filters was considered and novel controllers that can improve closed-loop sensitivities for the loop broken at both the error and the control were presented.

In summary, it was shown that precisely quantifying design objectives can allow us to compare methods and determine which technique is suitable for a given situation.

Future work will examine these design constraints for a wider variety of topologies including DC-DC boost and buck-boost converters, and also examine more complicated types of load like RL, RC and RLC loads for the buck converter.

REFERENCES

- [1] C. Zhang, J. Wang, S. Li, B. Wu, and C. Qian. “Robust control for PWM-based DC–DC buck power converters with uncertainty via sampled-data output feedback.” In *IEEE Transactions on Power Electronics* 30, no. 1 (2014): 504-515.
- [2] J. López, S. I. Seleme Jr, P. F. Donoso, L. M. F. Morais, P. C. Cortizo, and M. A. Severo. “Digital control strategy for a buck converter operating as a battery charger for stand-alone photovoltaic systems.” *Solar Energy* 140, 2016, pp.171-187.
- [3] M. Fu, C. Fei, Y. Yang, Q. Li, and F. Lee. “A GaN-Based DC/DC Module for Railway Applications: Design Consideration and High-Frequency Digital Control.” *IEEE Transactions on Industrial Electronics*, 2019.
- [4] J. G. Kassakian, M. F. Schlecht, and G. C. Verghese. *Principles of power electronics*. Graphis, 2000.
- [5] O. Cifdaloz, S. Konasani, A. A. Rodriguez, M. Islam, and D. Allee. “A sampled-data approach to dc-dc buck converter design.” In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC’05. 44th IEEE Conference on*, pp. 4779-4784. IEEE, 2005.
- [6] O. Cifdaloz, S. Konasani, A. A. Rodriguez, M. Islam, and D. Allee. “DC-DC buck converter design: a sampled-data approach.” In *American Control Conference, 2006*, pp. 6-pp. IEEE, 2006.
- [7] P. N. Paraskevopoulos. *Digital control systems*. London, 1996.
- [8] J. Ackermann. *Sampled Data Control Systems*. Springer Verlag, New York, 1985.
- [9] G. F. Franklin, J. D. Powell, and M. L. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley, London, 1990 (Second Edition).
- [10] C. H. Houpis, and G. B. Lamont. *Digital Control Systems*. McGraw-Hill, New York, 1985.
- [11] B. C. Kuo. *Automatic Control Systems*. Prentice Hall, Englewood Cliffs, New Jersey, 1975 (Third Edition).
- [12] K. Ogata. *Discrete-Time Control Systems*. Prentice Hall, Englewood Cliffs, New Jersey, 1987.

- [13] C. L. Phillips, and H. T. Nagle Jr. *Digital Control System Analysis and Design*. Prentice Hall, Englewood Cliffs, New Jersey, 1984.
- [14] Y. Yamamoto. “A retrospective view on sampled-data control systems.” *CWI Quarterly* 9, no. 3 (1996): 261-276.
- [15] D. Pan, X. Ruan, C. Bao, W. Li and X. Wang, “Optimized controller design for LCL-type grid-connected inverter to achieve high robustness against grid-impedance variation,” *IEEE Trans. Ind. Electron.* 62, no. 3, pp. 1537-1547, 2015.
- [16] F. Blaabjerg, R. Teodorescu, M. Liserre and A.V. Timbus, “Overview of control and grid synchronization for distributed power generation systems,” *IEEE Trans. Ind. Electron.* 53, no. 5, pp. 1398-1409, 2006.
- [17] J. Spector, “Here’s Everything New About Tesla’s Updated Powerwall 2.0,” 2017, Retrieved from <https://www.greentechmedia.com/articles/read/what-really-matters-from-tesla-battery-powerwall-powerpack>
- [18] R. Peña-Alzola, M. Liserre, F. Blaabjerg, M. Ordonez, and Y. Yang, “LCL-filter design for robust active damping in grid-connected converters,” *IEEE Trans. Ind. Informat.* 10, no. 4, pp. 2192-2203, 2014.
- [19] J. He and Y.W. Li, “Hybrid voltage and current control approach For DGgrid interfacing converters with LCL filters,” *IEEE Trans. Ind. Electron.*, vol. 60, no. 5, pp. 1797-1809, Mar. 2013.
- [20] J. Yin, S. Duan, and B. Liu, “Stability analysis of grid-connected inverter with LCL filter adopting a digital single-loop controller with inherent damping characteristic,” *IEEE Trans. Ind. Informat.*, vol. 9, no. 2, pp 1104-1112, May 2013.
- [21] M. Liserre, A. Dell’Aquila, and F. Blaabjerg. Stability improvements of an LCL-filter based three-phase active rectifier characteristic. *Proc. IEEE 33rd Annu. Power Electron. Spec. Conf. (PESC’02)*, vol. 3, page 1195-1201, May 2002.
- [22] W. Wu et al., “A new design method for the passive damped LCL and LLCL filter-based single-phase grid-tied inverter,” *IEEE Trans. Ind. Electron.*, vol. 60, no. 10, pp 4339-4350, Sep. 2013.
- [23] J.S. Freudenberg and D.P. Looze, “Relations between properties of multivariable feedback systems at different loop-breaking points: Part i,” *Proc. 24th IEEE CDC, Florida*, page 250, 1986.

- [24] J.S. Freudenberg and D.P. Looze, "Relations between properties of multivariable feedback systems at different loop-breaking points: Part ii," *American Control Conf. IEEE, Seattle*, pages 771-777, 1986.
- [25] A. Julean, *Active damping of LCL filter resonance in grid connected applications,* MS Thesis, Aalborg Universitet, 2009.
- [26] P.A. Dahono, "A control method to damp oscillation in the input lc filter," *Power Electronics Specialists Conference*, pp 1630-1635, 2002.
- [27] C. Wessels et al., "Active damping of lcl-filter resonance based on virtual resistor for pwm rectifiers & stability analysis with different filter parameters," *IEEE Power Electronics Specialists Conference*, pp 3532-3538, 2008.
- [28] J. Dannehl et al., "Investigation of active damping approaches for pi-based current control of grid-connected pulse width modulation converters with LCL filters," *IEEE Trans. Ind. Appl.*, vol. 46, no. 4, pp 1509-1517, 2010.
- [29] C. Bao et al., "Step-by-step controller design for LCL-type grid-connected inverter with capacitor-current-feedback active-damping," *IEEE Trans. Power Electron.* 29, no. 3, pp 1239-1253, 2014.
- [30] R. Peña-Alzola et al., "Systematic design of the lead-lag network method for active damping in LCL-filter based three phase converters," *IEEE Trans. Ind. Informat.* 10, no. 1, pp 43-52, 2014.
- [31] V. Blasko and V. Kaura, "A novel control to actively damp resonance in input LC filter of a three-phase voltage source converter," *IEEE Trans. Ind. Appl.*, vol. 33, no. 2, page 542-550, 1997.
- [32] A. Hava, T. Lipo, and W. Erdman, "Utility interface issues for line connected PWM voltage source converters: A comparative study," *Conf. Proc. 10th Annu. Appl. Power Electron. Conf. Expo. (APEC'95)*, vol. 1, page 125-132, 1995.
- [33] X. Wang, F. Blaabjerg, and P.C. Loh, "Grid-current-feedback active damping for LCL resonance in grid-connected voltage-source converters," *IEEE Trans. Power Electron.* 31, no. 1, pp 213-223, 2016.
- [34] C. Dick et al., "Active damping of LCL resonance with minimum sensor effort by means of a digital infinite impulse response filter," *Proc. Eur. Conf. Power Electron.*, pp 1-8, 2007.
- [35] M. Hanif et al., "Two degrees of freedom active damping technique for LCL filter based grid connected PV systems," *IEEE Trans. Ind. Electron.*, vol. 61, no. 6, pp 2795-2803, Jun. 2014.

- [36] J. Xu et al., “Active damping-based control for grid connected LCL-filtered inverter with injected grid current feedback only,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 9, page 4746-4758, Sep. 2014.
- [37] B. Bahrani, M. Vasiladiotis, and A. Rufer, “High-order vector control of grid-connected voltage-source converters with LCL-filters,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 6, pp 2767-2775, 2014.
- [38] K. Puttannaiah, et al., “A Generalized Mixed-Sensitivity Convex Approach to Hierarchical Multivariable Inner-Outer Loop Control Design Subject to Simultaneous Input and Output Loop Breaking Specifications,” *IEEE American Control Conf.*, pp 5632-5637, 2016.
- [39] K. Puttannaiah, J.A. Echols, and A.A. Rodriguez, “A Generalized \mathcal{H}_∞ Control Design Framework for Stable Multivariable Plants subject to Simultaneous Output and Input Loop Breaking Specifications,” *IEEE American Control Conf.*, pp 6617-6622, 2015.
- [40] K. Puttannaiah, et al., “Analysis and Use of Several Generalized H-Infinity Mixed Sensitivity Framework for Stable Multivariable Plants Subject to Simultaneous Output and Input Loop Breaking Specifications,” *IEEE Conf. on Decision and Control*, pp 3310-3315, 2015.
- [41] A. A. Rodriguez et al., “Modeling, design and control of low-cost differential-drive robotic ground vehicles: Part I-Single vehicle study,” *IEEE Conf. on Control Tech. & Applications*, pp. 155-160, 2017.
- [42] D. G. Cartagena, et al., “Modeling of a multi-core processor thermal dynamics for development of Dynamic Thermal Management controllers,” *IEEE American Control Conf.*, pp. 6917-6922, 2016.
- [43] S.P. Boyd, C.H. Barratt. *Linear controller design: limits of performance*. Prentice Hall Englewood Cliffs, NJ, 1991.
- [44] A.S. Thyagaturu, et al. “R-fft: Function split at ifft/fft in unified lte cran and cable access network,” *IEEE Trans. Broadcasting*, 2018
- [45] A.S. Thyagaturu, et al. “SDN-based smart gateways (Sm-GWs) for multi-operator small cell network management,” *IEEE Trans. Network and Service Management*, 13(4):740–753, 2016.
- [46] Naeim Safori, *Design of a DC-DC Buck converter for ultra-low power applications in 65 nm CMOS process*. Master Thesis.
- [47] Damien Prêle, *Advanced Electronic Systems*. Master Thesis, Hanoi, Vietnam, (2017) pp.153 (cel - 00843641v6).

- [48] *Maxim Integrated Products, Inc.*, “An Introduction to Switch-mode Power Supplies”, Application Note 4087, Sept 27, (2007).
- [49] P. T. Krein. “Elements of power electronics.” *Oxford University Press*, New York City, 1998.
- [50] Marian K. Kazimierczuk *Pulse-Width-Modulated DC-DC Power Converters*, 2nd ed., Wiley (2015).
- [51] R. Kuzi, *Passive Power Filters*, Proceedings of the CAS-CERN Accelerator School: Power Converters, Baden, Switzerland, ed. R. Bailey, CERN-2015-003 (2013) CERN, Geneva,
- [52] W. Janke, *Averaged models of pulse-modulated DC-DC power converters: Part-I: Discussion of standard methods*. Vol 61 (4) (2012) pp.609-631.
- [53] R.W. Erickson, *Introduction to Power Electronics*, ECEN 4797/5797, Lecture 21, Department of Electrical, Computers and Energy Engineering, University of Colorado, Boulder. October 14 (2013).
- [54] C.K. Tse, *Complex behaviour in switching power converters*, Proceedings of the IEEE, Vol. 90, No.5, May (2002).
- [55] R.W. Erickson and D.Maksimovic, *Fundamentals of power electronics*, 2nd ed., Kluwer Academic Publishing, (2002).
- [56] R.W. Erickson, *Introduction to Power Electronics*, ECEN 4797/5797, Lecture 22, (2018), University of Colorado, Boulder.
- [57] EE462L, Spring 2014 DC-DC Buck Converter.
- [58] Mucunguzi Rwagasove, *24V-DC/DC Step-down converter*, Aalborg University, March 14, 2017.
<https://projectr.aau.dk/projectr/files/254738211/Bachelorprojekt.pdf>
- [59] N. Mohan, and T. M. Undeland. *Power electronics: converters, applications, and design*. John Wiley & Sons, 2007.
- [60] Tang, Nghia, Bai Nguyen, Reza Molavi, Shahriar Mirabbasi, Yangyang Tang, Philipp Zhang, Jonghoon Kim, Partha Pratim Pande, and Deukhyoun Heo. *Fully integrated buck converter with fourth-order low-pass filter*. IEEE Transactions on Power Electronics 32, no. 5 (2016): 3700-3707.
- [61] W. Kim, M. S. Gupta, G.-Y. Wei, and D. Brooks. *System level analysis of fast, per-core DVFS using on-chip switching regulators*. in Proc. IEEE 14th Int. Symp. High Perform. Comput. Archit., 2008, pp. 123–134.

- [62] W. Kim, D. Brooks, and G. Y. Wei. *A fully-integrated 3-level DC–DC converter for nanosecond-scale DVFS*. IEEE J. Solid-State Circuits, vol. 47, no. 1, pp. 206–219, Jan. 2012.
- [63] R. W. Erickson and D. Maksimovic. *Fundamentals of Power Electronics*. 2nd ed. Norwell, MA, USA: Kluwer, 2001.
- [64] S.G. Parker, B.P. McGrath, and D.G. Holmes, “Regions of active damping control for LCL filters,” *IEEE Transactions on Industry Applications* 50, no. 1, pp 424-432, 2014.
- [65] P. Hazucha et al. *A 233-MHz 80%-87% efficient four-phase DC–DC converter utilizing air-core inductors on package*. IEEE J. Solid-State Circuits, vol. 47, no. 1, pp. 206–219, Jan. 2012.
- [66] M. Alimadadi et al. *A fully integrated 660 MHz low-swing energy recycling DC–DC converter*. IEEE Trans. Power Electron., vol. 24, no. 6, pp. 1475–1485, Jun. 2009.
- [67] C. Huang and P. K. T. Mok. *A 100 MHz 82.4% efficiency package bond-wire based four-phase fully-integrated buck converter with flying capacitor for area reduction*. IEEE J. Solid-State Circuits, vol. 48, no. 2, pp. 2977–2988, Dec. 2013.
- [68] J. Wibben and R. Harjani. *A high-efficiency DC–DC converter using 2 nH integrated inductors*. IEEE J. Solid-State Circuits, vol. 43, no. 4, pp. 844–854, Apr. 2008.
- [69] S. S. Mohan, M. D. Hershenson, S. P. Boyd, and T. H. Lee. *Simple accurate expressions for planar spiral inductances*. IEEE J. Solid-State Circuits, vol. 34, no. 10, pp. 1419–1424, Oct. 1999.
- [70] Texas Instruments, Application report, January 2014.
- [71] A. A. Rodriguez, D. Allee, S. Konasani, M. Islam *Digital Compensation for a Class of High Speed Switching DC-DC Converters: Issues and Tradeoffs* Unpublished, December 20, 2003.
- [72] A. Sarkar, *Modeling and Control of a Three Phase Voltage Source Inverter with an LCL Filter*, MS Thesis, Arizona State Univ., 2015.
- [73] A. A. Rodriguez et al., “Modeling, design and control of low-cost differential-drive robotic ground vehicles: Part II-Multiple vehicle study,” *IEEE Conf. on Control Tech. & Applications*, pp. 161-166, 2017.

- [74] W.C. Messner, et al., “Lead and lag compensators with complex poles and zeros design formulas for modeling and loop shaping,” *IEEE control systems*, 27(1), pp 44-54, 2007.
- [75] J.C. Doyle, J. E. Wall, G. Stein, “Performance and robustness analysis for structured uncertainty,” *IEEE Conf. on Decision and Control*, vol. 21, pp 629-636, 1982.
- [76] IEEE Standards Association, “IEEE 1547 standard for interconnecting distributed resources with electric power systems,” IEEE Standards Association, Piscataway (2003).
- [77] G.C. Goodwin, S.F. Graebe and M.E. Salgado. *Control system design*, Vol. 240. *New Jersey: Prentice Hall*, 2001.
- [78] S. Skogestad and I. Postlethwaite. *Multivariable feedback control: analysis and design*, volume 2. Wiley New York, 2007.
- [79] J.L. Goffin and J.P. Vial, “Convex nondifferentiable optimization: A survey focused on the analytic center cutting plane method,” *Optimization Methods and Software*, 17(5):805–867, 2002.
- [80] D.C. Youla, H. Jabr, and J.J. Bongiorno, “Modern wiener-hopf design of optimal controllers—part ii: The multivariable case,” *IEEE Trans. Automatic Control*, 21(3):319–338, 1976.
- [81] IEEE Standards Association, “IEEE 1547 standard for interconnecting distributed resources with electric power systems,” IEEE Standards Association, Piscataway (2003).
- [82] A. A. Rodriguez. *Analysis and Design of Multivariable Feedback Control Systems*. Control3D, L.L.C., Tempe, AZ 85284, 2004.
- [83] R. W. Erickson, and D. M. Maksimovic. *Fundamentals of power electronics*. 2nd Edition, Kulwer Academic Publishers, 2002.
- [84] B.A. Bamieh, and J.B. Pearson. “A General Framework for Linear Periodic Systems with Applications to \mathcal{H}^∞ Sampled-Data Control.” In *IEEE Transactions on Automatic Control*, Vol. AC-37, 1992, pp. 418-435.
- [85] S. Buso , and P. Mattavelli. “Digital control in power electronics.” *Lectures on power electronics 1*, no. 1 (2006): 1-158.
- [86] S. Skogestad, and I. Postlethwaite. *Multivariable feedback control: analysis and design*. Vol. 2. New York: Wiley, 2007.

- [87] Y. Yamamoto, and P. P. Khargonekar. "Frequency response of sampled-data systems." *IEEE Transactions on Automatic Control* 41, no. 2 (1996): 166-176.
- [88] B. A. Bamieh. "Intersample and Finite Wordlength Effects in Sampled-Data Problems." *IEEE Transactions on Automatic Control*, Vol. 48, No. 4, April, 2003, pp. 639–642.
- [89] T. Chen, and B. Francis. "Optimal Sampled-Data Control Systems." *Springer Verlag*, New York, 1995.
- [90] B. A. Bamieh, J. B. Pearson, B. A. Francis, and A. R. Tannenbaum. "A Lifting Technique for Linear Periodic Systems with Applications to Sampled-Data Control." *Systems and Control Letters*, 17, 1991, pp. 79–88.
- [91] W. Sun, K. M. Nagpal, and P. P. Khargonekar. " H_∞ control and filtering for sampled-data systems." *IEEE Transactions on Automatic Control* 38, no. 8 (1993): 1162-1175.
- [92] M. di Bernardo, F. Garofalo, L. Glielmo, and F. Vasca. "Nonlinear phenomena in pulse width modulated feedback controlled systems." In *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*, vol. 2, pp. 2161-2166. IEEE, 1996.
- [93] D. Raviv, and E. W. Djaja. "Technique for Enhancing the Performance of Discretized Controllers." *IEEE Control Systems*, pp. 52–57, 1999.
- [94] K. Zhou with J.C. Doyle. "Essentials of Robust Control." *Prentice Hall*, 1998.
- [95] M. Saberi, R. Lotfi, K. Mafinezhad, and W.A. Serdijn. "Analysis of power consumption and linearity in capacitive digital-to-analog converters used in successive approximation ADCs." *IEEE Transactions on Circuits and Systems I: Regular Papers* 58, no. 8 (2011): 1736-1748.
- [96] H. Hong, and G. Lee. "A 65-fJ/conversion-step 0.9-V 200-kS/s rail-to-rail 8-bit successive approximation ADC." *IEEE Journal of Solid-State Circuits* 42, no. 10 (2007): 2161-2168.
- [97] A. V. Peterchev, and S.R. Sanders. "Quantization resolution and limit cycling in digitally controlled PWM converters." *IEEE Transactions on Power Electronics* 18, no. 1 (2003): 301-308.
- [98] S. Choudhury. "Designing a TMS320F280x based digitally controlled DC-DC switching power supply." *Texas Instruments Application Report SPRAAB3* (2005).
- [99] P. N. Paraskevopoulos. "Digital control systems." *Prentice Hall*, London, 1996.

- [100] A. Sarkar, K. Puttannaiah, and A.A. Rodriguez. “Inner-outer loop based robust active damping for lcl resonance in grid-connected inverters using grid current feedback.” In *2018 Annual American Control Conference (ACC)*, pp. 6766–6771. IEEE, 2018.
- [101] Philips Semiconductors N-channel TrenchMOS transistor Logic level FET PHP21N06LT, PHB21N06LT, PHD21N06LT datasheet.
- [102] P. A. Iglesias, and K. Glover. “State-space approach to discrete-time, \mathcal{H}^∞ control.” *International Journal of Control*, vol. 54, no. 5 (1991): 1031–1073.
- [103] H. Fujioka, C.-Y. Kao, S. Almér, and U. Jönsson. “Sampled-data H^∞ control design for a class of PWM systems.” *Proceedings of the 44th IEEE Conference on Decision and Control*, IEEE (2005): 4499–4504.
- [104] H. Fujioka, C.-Y. Kao, S. Almér, and U. Jönsson. “Robust tracking with H^∞ performance for PWM systems.” *Automatica*, vol. 45, no. 8 (2009): 1808–1818.
- [105] S. Mariethoz, S. Almer, M. Baja, A. G. Beccuti, D. Patino, A. Wernrud, J. Buisson, H. Cormerais, T. Geyer, H. Fujioka, U. T. Jonsson, C. Kao, M. Morari, G. Papafotiou, A. Rantzer, and P. Riedinger. “Comparison of Hybrid Control Techniques for Buck and Boost DC-DC Converters.” *IEEE Transactions on Control Systems Technology*, vol. 18, no. 5 (2010): 1126–1145.
- [106] C.-C. Fang. “Sampled Data Analysis and Control of DC-DC Switching Converters.” *Ph.D. Thesis*, 1997.
- [107] Y. Duan, and H. Jin. “Digital controller design for switchmode power converters.” *Fourteenth Annual Applied Power Electronics Conference and Exposition, APEC 1999*, Vol. 2, March 1999, pp. 967–973.
- [108] J.S. Freudenberg, and D.P. Looze. ”Sensitivity reduction, nonminimum phase zeros, and design tradeoffs in single loop feedback systems.” In *Decision and Control*, 1983. The 22nd IEEE Conference, pp. 625-630. IEEE, 1983.
- [109] B. Shang, J. Liu, Y. Zhang, C. Wu, and Y. Q. Chen. “Fractional-order flight control of quadrotor UAS on vision-based precision hovering with larger sampling period.” *Nonlinear Dynamics*, vol. 97, no. 2, (2019): 1735–1746.
- [110] P. Song, C. Cui, and Y. Bai. “Robust output voltage regulation for DC–DC buck converters under load variations via sampled-data sensorless control.” *IEEE Access*, vol. 6, (2018): 10688–10698.
- [111] R. C. N. Pilawa-Podgurski, D. M. Giuliano, and D. J. Perreault. “Merged two-stage power converter architecture with softcharging switched-capacitor energy transfer.” *2008 IEEE Power Electronics Specialists Conference*, (2008): 4008–4015.

- [112] R. C. N. Pilawa-Podgurski, and D. J. Perreault. “Merged two-stage power converter with soft charging switched-capacitor stage in 180 nm CMOS.” *IEEE Journal of Solid-State Circuits*, vol. 47, no. 7, (2012): 1557–1567.
- [113] G. E. Dullerud, and F. Paganini. “A course in robust control theory: a convex approach.” *Springer Science & Business Media*, vol. 36, 2013.
- [114] O. Cifdaloz. “H-infinity mixed-sensitivity optimization for infinite dimensional plants subject to convex constraints.” *Arizona State University*, 2007.
- [115] K. Gopalakrishna, Frequency Characterization of Si, SiC, and GaN MOSFETs Using Buck Converter In CCM as an Application. 2013.

APPENDIX A

A MATLAB CODE: LIFTING

This m file designs all the 3 types of digital controllers for the buck converter. A controller is designed for each value of switching period $h = T_s$ in a range of T_s . The range is $T_s \in (10, 490) \mu s$.

```

1 % m File: buck_Geqd_vs_h.m
2 %
3 % *****
4 % *****
5 % *****
6 %
7 % H-INFINITY EQUIVALENT GENERALIZED PLANT
   Geqd CHARACTERISTICS AS A
8 % FUNCTION OF SAMPLING PERIOD h
9 %
10 %
11 %
12 % This program examines how key
   characteristics of the H-infinity
   equivalent
13 % discrete time generalized plant  $G_{\{eq,d\}}$ 
   vary with sampling period h for
14 % a DC-DC "buck converter" Sampled Data (
   SD) H-infinity
15 % Mixed Sensitivity (MS) problem.
16 %
17 % *****

```

```

18 % 1. START: SD_sim.slx
19 % 2. ADD THE SD FILES TO THE MATLAB PATH
20 % 3. THEN RUN THIS CODE
21 % *****
22 %
23 % References:
24 %
25 %   Chen and Francis's "Optimal Sampled
      Data Control Systems"
26 %   Cifdaloz, Oguzhan, Siva Konasani,
      Armando A. Rodriguez, Murshidul Islam,
      and David Allee. "A sampled-data
      approach to dc-dc buck converter
      design." In Proceedings of the 44th
      IEEE Conference on Decision and
      Control, pp. 4779-4784. IEEE, 2005.
27 %
28 % The following characteristics are
      examined:
29 %
30 %   General open/closed loop frequency
      responses and singular values vs. h
31 %   Number of input (exogeneous) signals
      w of  $G_{eq,d}$  vs. h
32 %   Number of output (regulated) signals

```

```

    z of  $G_{\{eq,d\}}$  vs. h
33 % Optimal performance  $\gamma_{opt}$  vs. h
34 % NOTE: The MS framework provides a
    suboptimal solution within a
35 % prescribed tolerance.
36 % Order of  $G_{\{eq,d\}}$  vs. h
37 % Order of minimal realization of  $G_{\{eq$ 
    ,d\} vs. h
38 %
39 % *****
40 % *****
41 % *****
42
43 % *****
44 %
45 % CLEAR COMMAND WINDOW, CLOSE FIGURES
46 %
47 close all;
48 clc;
49 clear all;
50
51 % *****
52 %
53 % SAVE FIGURES CONTROL
54 %

```

```

55 savefigs = 0;
56
57
58 % *****
59 %
60 % FIGURE RELATIVE FILE PATH
61 %
62 relpath = 'figures\';
63
64
65 % *****
66 %
67 % CREATE TIMESTAMP (FILE SAVING PURPOSES)
68 %
69
70 if savefigs
71
72     time = fix(clock);
73     timestamp = '';
74     for i = 1:length(time)
75         timestamp = [timestamp , num2str(
76             time(i))];
77         if i < length(time)
78             timestamp = [timestamp , '_'];
79         end

```

```

79     end
80     timestamp = [timestamp, '\'];
81     relpath = [relpath, timestamp];
           % Update relative path
82     mkdir(relpath);
                                           % Create
           directory for relative path
83
84 end
85
86 % *****
87 %
88 % ADD SDToolbox
89 %
90 addpath(' ../.. /SDToolbox ');
91
92 % *****
93 %
94 % INITIALIZE FIGURE COUNTER
95 %
96 figcount = 1;
97
98 % *****
99 %
100 % DEFINE COMPLEX VARIABLE s AS LAPLACE

```

```

                                TRANSFORM VARIABLE
101 %
102 s = tf('s');
103
104 % *****
105 %
106 % FREQUENCY VECTOR FOR BODE PLOTS/
                                FREQUENCY RESPONSES
107 %
108 wvec = logspace(-3,4,1000);
109
110 % *****
111 %
112 % SWEPT PARAMETERS
113 %
114
115 % *****
116 %
117 % SAMPLING PERIOD h
118 %
119
120 hvec = (10:10:490) * 1e-6;                                %
                                RANGE CHOSEN CONSIDERING h_crit =
                                212.06 us
121 % 5e-6 gives an error !!!

```



```

122
123 % hvec = [10e-6 100e-6 1e-3]; % gives an
      error !!!
124 % hvec = [ 1];
125 % hvec = logspace(-5,log10(0.75),200);
126 length_hvec = length(hvec);
127
128 % *****
129 %
130 % TARGET CLOSED LOOP BANDWIDTH wb
131 %
132 % wbvec = [1];
133 wbvec = 1e3;
134 length_wbvec = length(wbvec);
135
136
137
138 % *****
139 %
140 % FIXED STATE SPACE REPRESENTATIONS
141 %
142
143
144 % *****
145 %

```

```

146 % REGULARIZATION GAIN epsilon_1
147 %
148 % epsilon_1 is set to 0.01.
149 %
150 % epsilon_1 is set nonzero to satisfy
      condition (A2) (Chen and Francis pg.
151 % 26) is satisfied (so that the direct
      feedthrough matrix D_12 from
152 % control signal u to regulated signals z
      of the mixed sensitivity
153 % problem is full column rank); i.e., so
      the control signal u is weighted
154 % in both z_1 and z_2. As is, D_12 = [0 ;
      epsilon_1], so nonzero epsilon_1
155 % is needed to give D_12 full column rank
      .
156 %
157
158 eps1 = 0.1;
159
160 %%
161
162 % *****
163 %
164 % STORAGE

```

```

165 %
166 Geqd_cell = cell(length_hvec ,
                    length_wbvec);    % Holds H-inf
                    equivalent DT generalized plants
167
168 Kc_cell = cell(length_hvec , length_wbvec)
        ;    % Holds H-inf optimal CT
            controllers
169 Kd_cell = cell(length_hvec , length_wbvec)
        ;    % Holds H-inf optimal DT
            controllers
170
171 Wc_cell = cell(length_hvec , length_wbvec)
        ;    % Holds weighting functions W
            for CT designs
172 W_cell = cell(length_hvec , length_wbvec);
            % Holds weighting functions W
            for DT designs
173
174 F_cell = cell(length_hvec , length_wbvec);
            % Holds AAFs for designs
175 P_cell = cell(length_hvec , length_wbvec);
            % Holds P for designs
176
177 gamma_Wcinv_cell = cell(length_hvec ,

```

```

length_wbvec);      % Holds  $\gamma_c * W_c^{-1}$ 
178 gamma_Winv_cell = cell(length_hvec ,
length_wbvec);      % Holds  $\gamma_d * W^{-1}$ 
179
180 gamma_c_mat = zeros(length_hvec ,
length_wbvec);      % Holds performances
of controllers for CT designs
181 gamma_d_mat = zeros(length_hvec ,
length_wbvec);      % Holds performances
of controllers for DT designs
182
183 hmat = zeros(length_hvec , length_wbvec);
184
185 Geqd_numin_mat = zeros(length_hvec ,
length_wbvec);      % Holds number of
inputs of  $G_{\{eq,d\}}$ 
186 Geqd_numout_mat = zeros(length_hvec ,
length_wbvec);      % Holds number of
outputs of  $G_{\{eq,d\}}$ 
187 Geqd_order_mat = zeros(length_hvec ,
length_wbvec);      % Holds order of  $G_{\{eq,d\}}$ 
188 Geqd_minreal_order_mat = zeros(

```

```

length_hvec , length_wbvec);      % Holds
    order of minreal(G_{eq,d})

189
190 %%
191
192 % *****
193 % *****
194 %
195 % BEGIN MAIN LOOP
196 %
197 % Sweep W bandwidth wb, sampling period h
198 %
199 % *****
200 % *****
201
202 for wbcount = 1:length_wbvec
203
204     % *****
205     %
206     % CURRENT BANDWIDTH wg
207     %
208     wb = wbvec(wbcount);
209
210
211     for hcount = 1:length_hvec

```

```

212
213     % *****
214     %
215     % CURRENT SAMPLING PERIOD h
216     %
217     h = hvec(hcount);
218
219     % *****
220     %
221     % CALCULATE NYQUIST FREQUENCY wn
222     %
223     % Sampling frequency ws = 2*pi/h
224     % Nyquist frequency wn = ws/2 =
225         pi/h
226     %
227     wn = pi / h;
228
229     % *****
230     %
231     % PLANT STATE SPACE
232         REPRESENTATION
233     %
234     % Plant is a DC-DC buck converter

```

```

        and the plant model is given
        in
235     % the 2005 CDC paper referenced
        above.

236
237     % IMPORTANT: P: up → yp, for the
        TF from di → yp and do → yp
        ,
238     % we'll need W_di and Zout (refer
        to the paper for exact
        details)
239     load( 'BUCKDATA.mat' );
240     [Ap, Bp, Cp, Dp] = ssdata(P);
241
242
243     P_cell{hcount, wbcount} = P;
244
245
246     % PLANT DIMENSIONS:
247     nx_p = size(Ap,1);      % Number
        of states
248     nu_p = size(Bp,2);      % Number
        of inputs
249     ny_p = size(Cp,1);      % Number
        of outputs

```

```

250
251      % *****
252      %
253      % ANTI-ALIASING FILTER STATE
          SPACE REPRESENTATION
254      %
255      %  $F = (af/(s+af))$ 
256      %
257
258      % af = 100;          % Filter pole
          location (in LHP)
259 %      af = 0.5*wn;      % Filter
          pole location (in LHP)
260      af = 1* wb;        % Filter pole
          location (in LHP)
261
262      Af = -af;
263      Bf = af;
264      Cf = 1;
265      Df = 0;
266
267      F = ss (Af, Bf, Cf, Df);
268
269      F_cell {hcount, wbcnt} = F;
270

```



```

271     % F = tf(1);
272
273
274     % FILTER DIMENSIONS:
275     nx_f = size(Af,1);      % Number
                             % of states
276     nu_f = size(Bf,2);     % Number
                             % of inputs
277     ny_f = size(Cf,1);     % Number
                             % of outputs
278
279
280     % *****
281     %
282     % SENSITIVITY AT ERROR WEIGHTING
283     % FUNCTION STATE SPACE
284     % REPRESENTATION
285     %
286     %  $W = 1/Ms * (s+Ms*wb) / (s+eps*wb) * (aw) / (s+aw)$ 
287     % Where  $wb > 0$ ,  $0 < eps \ll 1 < Ms$ 
288     % .  $Ms*wb$  is the dominant zero
289     % location in
290     % the LHP (nominally, we want  $W \sim$ 

```

```

    (s+1)/s, so Ms = 1 + eps, wb
    = 1). The
288 % pole at awr is a HFRO pole
    needed for regularization (D11
    = 0) — see
289 % Chen and Francis, pp. 314.
290 %
291
292 eps = 1e-1;
293 Ms = 5;
294
295 % wb = wb;
296
297 aw = 10*Ms*wb; % HFRO
    regularization pole
298 % aw = 10*wn; % HFRO
    regularization pole
299
300 Aw = [ -aw aw
301         0 -eps*wb
302         ];
303
304 Bw = [ aw * 1/Ms
305        wb*(Ms-eps)/Ms ];

```

```

306         Cw = [ 1 0 ];
307
308         Dw = 0;
309
310         %         Aw = -bw*wn;
311         %         Bw = bw*wn * 10;
312         %         Cw = 1;
313         %         Dw = 0;
314
315         W = ss(Aw,Bw,Cw,Dw);
316
317         W_cell{hcount , wbcount} = W;
318
319         % For continous design (no
320         %         weighting function rolloff)
321         Wc = ss(-eps*wb, wb*(Ms-eps)/Ms
322         , 1, 1/Ms);
323         Wc_cell{hcount , wbcount} = Wc;
324
325         % W = ss(-eps*wb, wb*(Ms-eps)/Ms,
326         %         1, 1/Ms);
327         % W = 1/Ms * (s+Ms*wb)/(s+eps*wb)
328         %         ;
329         % W = (1/( (5/wn)*s + 1))^2;
330         %

```

```

327     % Wss=ss (W) ;
328     % Aw=Wss.a; Bw=Wss.b; Cw=Wss.c ;
        Dw=Wss.d;

329
330
331     % WEIGHTING FUNCTION DIMENSIONS:
332     nx_w = size(Aw,1) ;      % Number
        of states
333     nu_w = size(Bw,2) ;      % Number
        of inputs
334     ny_w = size(Cw,1) ;      % Number
        of outputs

335
336
337
338     % *****
339     %
340     % GENERALIZED PLANT STATE SPACE
        REPRESENTATION:
341     %
342     %           | A | B1 B2 |
        | A | B1 B2 |
343     %   G(s) = |-----|
        = |-----|
344     %           | C1 | D11 D12 |

```

```

345         | C1 | 0   D12 |
%           | C2 | D21 D22 |
           | C2 | 0   0   |
346 %
347 % For discussion of structural
           assumptions (i.e., D11=0, D21
           =0, D22=0), see
348 % Chen and Francis, pp. 314
349 %
350 % States:   [xw, xf, xp]
351 % Inputs:   [w, u]
352 % Outputs:  [z1, z2, y]
353 %
354
355 % GENERALIZED PLANT DIMENSIONS:
356 nx = nx_p + nx_f + nx_w;           %
           Number of states
357 nw = 1;                             %
           Number of exogeneous signals
           w
358 nu = 1;                             %
           Number of control signals u
359 nz = 2;                             %
           Number of regulated signals z
360 ny = 1;                             %

```

Number of measured signals y

```

361
362     A = [   Aw           zeros
           (nx_w , nx_f)   -Bw*Cp
363           zeros (nx_f , nx_w)   Af
                                           -Bf*
                                           Cp
364           zeros (nx_p , nx_w)   zeros
           (nx_p , nx_f)   Ap
                                           ];
365
366     B1 = [   Bw
              Bf
367           zeros (nx_p , nw)   ];
368
369
370     B2 = [   -Bw*Dp
              -Bf*Dp
371           Bp           ];
372
373
374     C1 = [   Cw           zeros
              (ny_w , nx_f)   -Dw*Cp
375           zeros (nz-1 , nx_w)   zeros
              (nz-1 , nx_f)   zeros (
              nz-1 , nx_p)   ];
376

```

```

377         C2 = [ zeros (ny ,nx_w)      Cf
                -Df*Cp    ];
378
379         D11 = [ Dw
380                zeros (nz-1,nw)    ];
381
382         D12 = [ -Dw*Dp
383                eps1    ];
384
385         D21 = Df;
386
387         D22 = -Df*Dp;
388
389         B = [ B1 B2 ];
390
391         C = [ C1
392                C2 ];
393
394         D = [ D11 D12
395                D21 D22 ];
396
397         G = ss (A,B,C,D);
398
399
400         % *****

```

```

401      %
402      % PERFORM CONTINUOUS-TIME H-INF
          DESIGN
403      %
404      % H-infinity controller synthesis
          using "hinfosyn"
405      % [K,CL,gamma] = hinfosyn(P,nmeas,
          ncont)
406      % See:      https://www.mathworks.com/help/robust/ref/hinfosyn.html
407      %
408      % The regularization condition
          D21 = 0 for SD H-infintiy
          design conflicts
409      % with the regularization
          condition that D21 have full
          row rank for
410      % continuous-time H-infinity
          design. Chen and francis
          introduce a second
411      % exogeneous signal w2 which is
          injected after the anti-
          aialiasing filter F
412      % after passing through a small

```



```

    gain epsilon_2 in a similar H-
    infinity
413 % example (see Example 2.3.1, pp.
    31). We will do the same here
    for the
414 % continuous design.
415 %
416 eps2 = 0.00001;
417
418 B1c = [B1 zeros(nx,1)];
419 D11c = [D11 zeros(nz,1)];
420 D21c = [D21 eps2];           % Now
    D21 has full row rank
421
422 Gc = [ Wc      0      -series(P
    ,Wc)
423      0      0      eps1
424      F      eps2      -series(P
    ,F)    ];
425
426 [Kc, cl , gamma_c] = hinfsyn(Gc, ny,
    nu, 'display', 'on');
427
428 % *****
429 %

```

```

430 % STORAGE
431 %
432 Kc_cell{hcount , wbcoun} = Kc;
                                % H-inf opt.
                                controller
433 gamma_c_mat(hcount , wbcoun) =
                                gamma_c; % Performance
                                achieved
434 gamma_Wcinv_cell{hcount , wbcoun}
                                = gamma_c * Wc^(-1); %
                                gamma * W^(-1)
435
436
437 % *****
438 %
439 % BEGIN H-INF OPTIMAL DESIGN VIA
                                H-INF EQUIVALENT DISCRETE
                                PLANT (CH 13 CHEN AND FRANCIS)
440 %
441 % For computational procedure of
                                H-inf synthesis , see Chen and
                                Francis ,
442 % sec . 13.8 , pp . 342-343.
443 %
444

```

```

445     tol = 1e-3;           % Tolerance
                           desired within optimal
                           performance
446
447     [Kd, Teqd, Geqd, gamma_d] = ms_sd
                           (G, nu, ny, h, tol);
448
449     minreal_Geqd = minreal(Geqd); %
                           Minimum realization of G_{eq,
                           d}
450
451     % Store generalized plant G_{eq,d
                           } characteristics
452     Geqd_numin_mat(hcount, wbcnt) =
                           size(Geqd.d, 2);
453     Geqd_numout_mat(hcount, wbcnt) =
                           size(Geqd.d, 1);
454     Geqd_order_mat(hcount, wbcnt) =
                           size(Geqd.a, 1);
455     Geqd_minreal_order_mat(hcount,
                           wbcnt) = size(minreal_Geqd.a
                           , 1);
456
457     % *****
458     %

```

```

459         % STORAGE
460         %
461         Geqd_cell{hcount , wbcount} = Geqd;
462         Kd_cell{hcount , wbcount} = Kd;
463         % H-inf opt.
464         controller
465         gamma_d_mat(hcount , wbcount) =
466         gamma_d;           % Performance
467         achieved
468         gamma_Winv_cell{hcount , wbcount} =
469         gamma_d * W^(-1);   % gamma
470         * W^(-1)
471
472     end           % END for bwfraccount
473     = 1:length_bwfracvec
474
475 end           % END for hcount = 1:
476     length_hvec
477
478 %%
479
480 % *****
481 % *****
482 %
483 % FORM CLOSED LOOP MAPS

```

```

476 %
477 % *****
478 % *****
479
480 % *****
481 %
482 % STORAGE
483 %
484 Sed_cell = cell(length_hvec , length_wbvec
                );      % Holds DT sensitivities at
                plant output
485 Ted_cell = cell(length_hvec , length_wbvec
                );      % Holds DT comp. sensitivities
                at plant output
486 Trud_cell = cell(length_hvec ,
                length_wbvec);      % Holds DT
                reference to control
487 Led_cell = cell(length_hvec , length_wbvec
                );      % Holds DT loop
488
489 Sec_cell = cell(length_hvec , length_wbvec
                );      % Holds CT sensitivities at
                plant output
490 Tec_cell = cell(length_hvec , length_wbvec
                );      % Holds CT comp. sensitivities

```

```

        at plant output
491 Truc_cell = cell(length_hvec ,
                    length_wbvec);    % Holds CT comp.
        reference to control
492 Lec_cell = cell(length_hvec , length_wbvec
                    );    % Holds CT loop
493
494
495 for wbcount = 1:length_wbvec
496
497     % *****
498     %
499     % CURRENT BANDWIDTH wb
500     %
501     wb = wvec(wbcount);
502
503
504     for hcount = 1:length_hvec
505
506         % *****
507         %
508         % CURRENT SAMPLING PERIOD h
509         %
510         h = hvec(hcount);
511

```

```

512      % *****
513      %
514      % CALCULATE NYQUIST FREQUENCY wn
515      %
516      % Sampling frequency ws = 2*pi/h
517      % Nyquist frequency wn = ws/2 =
          pi/h
518      %
519      wn = pi / h;
520
521      % *****
522      %
523      % EXTRACT PLANT
524      %
525      P = P_cell{hcount, wbcount};
526
527      % *****
528      %
529      % EXTRACT FILTER
530      %
531      F = F_cell{hcount, wbcount};
532
533      % *****
534      %
535      % ZOH-TRANSFORMATION OF FILTER

```

```

536      %
537      Fd = c2d(F,h, 'zoh ');
538
539      % *****
540      %
541      % ZOH-TRANSFORMATION OF FILTER
542      %
543      Pd = c2d(P,h, 'zoh ');
544
545      % *****
546      %
547      % ZOH-TRANSFORMATION OF PLANT AND
548          FILTER
549      %
550      PFd = c2d(series(F,P),h, 'zoh ');
551
552      % *****
553      %
554      % CONTROLLER (DISCRETE)
555      %
556      Kd = Kd_cell{hcount,wbcount};
557
558      % *****
559      %

```



```

560      % LOOP (DISCRETE)
561      %
562      Led = series (PFd,Kd);    % Loop
           broken at error
563      Led_cell{hcount ,wbcount} = Led;
564
565      % *****
566      %
567      % SENSITIVITY (DISCRETE)
568      %
569      Sed_cell{hcount ,wbcount} =
           feedback (1 ,Led) ;
570
571      % *****
572      %
573      % COMPLEMENTARY SENSITIVITY (
           DISCRETE)
574      %
575      Ted_cell{hcount ,wbcount} =
           feedback (Led ,1) ;
576
577      % *****
578      %
579      % Tru (DISCRETE)
580      %

```

```

581      Trud_cell{hcount , wbcoun} =
          feedback( series (Fd,Kd) ,Pd) ;

582

583      % *****

584      %

585      % CONTROLLER (CONTINUOUS)

586      %

587      Kc = Kc_cell{hcount , wbcoun} ;

588

589      % *****

590      %

591      % LOOP (CONTINUOUS)

592      %

593      Lec = series ( series (P,Kc) ,F) ;   %
          Loop broken at error

594      Lec_cell{hcount , wbcoun} = Lec ;

595

596      % *****

597      %

598      % SENSITIVITY (CONTINUOUS)

599      %

600      Sec_cell{hcount , wbcoun} =
          feedback(1 , series ( series (P,Kc)
          ,F)) ;

601

```

```

602     % *****
603     %
604     % COMPLEMENTARY SENSITIVITY (
        CONTINUOUS)
605     %
606     Tec_cell{hcount ,wbcount} =
        feedback( series( series(P,Kc) ,F
        ),1);
607
608     % *****
609     %
610     % Tru (CONTINUOUS)
611     %
612     Truc_cell{hcount ,wbcount} =
        feedback( series(F,Kc) ,P);
613
614     end                % END for bwfraccount
        = 1:length_bwfracvec
615
616     end                % END for hcount = 1:
        length_hvec
617
618     %%
619
620     % *****

```

```

621 % *****
622 %
623 % PLOT FREQUENCY RESPONSES
624 %
625 % *****
626 % *****
627
628
629 % *****
630 %
631 % LEGEND ENTRIES
632 %
633
634 lgd_cell = cell(length_hvec , length_wbvec
        );          % Holds legend entries
635
636 for wbcount = 1:length_wbvec
637
638     % *****
639     %
640     % CURRENT BANDWIDTH wb
641     %
642     wb = wvec(wbcount);
643
644     for hcount = 1:length_hvec

```

```

645
646     % *****
647     %
648     % CURRENT SAMPLING PERIOD h
649     %
650     h = hvec(hcount);
651
652     % *****
653     %
654     % FORM LEGEND ENTRY
655     %
656     lgd_cell{hcount,wbcount} = [ 'h =
        ' num2str(h) ];
657
658     end % END for bwfraccount
        = 1:length_bwfracvec
659
660 end % END for hcount = 1:
        length_hvec
661
662 % *****
663 %
664 % FREQUECNIES OF EVALUATION
665 %
666 % The frequency vector for each design

```

```

        depends on w_n, so each design needs
667 % its own frequency vector.
668 %
669
670 wvec_d_cell = cell(length_hvec ,
        length_wbvec);          % Holds
        frequency vector for each design
671
672 for wbcount = 1:length_wbvec
673
674     % *****
675     %
676     % CURRENT BANDWIDTH wb
677     %
678     wb = wbvec(wbcount);
679
680     for hcount = 1:length_hvec
681
682         % *****
683         %
684         % CURRENT SAMPLING PERIOD h
685         %
686         h = hvec(hcount);
687
688         % *****

```

```

689         %
690         % CALCULATE NYQUIST FREQUENEY wn
691         %
692         % Sampling frequency ws = 2*pi/h
693         % Nyquist frequeneqy wn = ws/2 =
           pi/h
694         %
695         wn = pi / h;
696
697         % *****
698         %
699         % FREQUENCIES OF EVALUATION
700         %
701         wvec_d_cell{hcount , wbcount} =
           logspace(-4, log10(0.999*wn)
           ,1000);
702
703
704     end           % END for bwfraccount
           = 1:length_bwfracvec
705
706 end
707
708 %%
709

```

```

710 % *****
711 %
712 % PLOTS — DISCRETE
713 %
714
715 plotfreqresp = 1;
716
717 if plotfreqresp
718
719 % *****
720 %
721 % BOOLEAN CONTROL TO PLOT ONE CT DESIGN
       AT BEGINNING OF EACH PLOT
722 %
723 % If include_ct = true, then the first
       continuous-time design corresponding
724 % to the current value of wb (i.e., the
       lowest h-value design for the
725 % current value of wb) will be plotted in
       front of each of the DT designs
726 %
727 include_ct = 1;
728
729
730 for wbcnt = 1:length_wbvec

```



```

731
732 % *****
733 %
734 % CURRENT BANDWIDTH wb
735 %
736 wb = wvvec(wbcount);
737
738 % *****
739 %
740 % FILE NAME/PLOT TITLE MODIFIERS
741 %
742 % Appended to the end of each file
       name to make names unique
743 %
744 filenamemod = [ ' — wb eq ' strrep(
       num2str(wb), '.', 'p') ' DT' ];
745 % titlemod = [ ' | h = ' num2str(h) '
sec '];
746 titlemod = '';
747
748 % *****
749 %
750 % DETERMINE X-AXIS LIMITS
751 %
752 tmp = wvec_d_cell{1,1};

```

```

753     wmin = tmp(1);
754     tmp = wvec_d_cell{1,end};
755     wmax = tmp(end);
756
757     % Frequency vector for the CT design
758     % (if it is included in these plots)
759     wvec_c = logspace(log10(wmin),log10(
760         wmax),1000);
761
762     % *****
763     %
764     % CONTROLLER FREQUENCY RESPONSE
765     %
766
767     axes_vec_mag = [wmin,wmax,-40,80];
768     % Window to be used for
769     % magnitude
770     axes_vec_ph = [wmin,wmax,-90,100];
771     % Window to be used for phase
772
773     % See plotbode.m under the folder "
774     % SDToolbox" for documentation
775
776     if include_ct
777         sys_cell = {Kc_cell{1,wbcount}
778             Kd_cell{: ,wbcount}};

```

```

771         lgd_text = { 'CT design' lgd_cell
                       {:, wbcnt}};
772         wvec_cell = [{ wvec_c};
                       wvec_d_cell(:, wbcnt)];
773     else
774         sys_cell = {Kd_cell {:, wbcnt}};
775         lgd_text = {lgd_cell {:, wbcnt}};
776         wvec_cell = wvec_d_cell(:, wbcnt
                                )
777     end
778     plottypes = [1;1];
779     ttl_cell = {[ 'Controller K Magnitude
                   Response' titlemod], [ 'Controller K
                   Phase Response' titlemod]};
780     axes_cell = {axes_vec_mag,
                   axes_vec_ph};
781
782     plotbode(sys_cell, wvec_cell,
              plottypes, ttl_cell, lgd_text,
              axes_cell, figcount);
783
784     % SAVE FIGURE
785     filename = [ 'K_mag' filenamemod];
786     if savefigs
787         savepdf(figcount, relpath, filename

```

```

);
788 end
789
790 figcount = figcount + 1;           %
      Increment figure counter
791
792 % SAVE FIGURE
793 filename = ['K_ph' filenamemod];
794 if savefigs
795     savepdf(figcount , relpath , filename
              );
796 end
797
798 figcount = figcount + 1;           %
      Increment figure counter
799
800 % *****
801 %
802 % LOOP FREQUENCY RESPONSE
803 %
804
805 axes_vec_mag = [wmin, wmax, -40, 40];
      % Window to be used for
      magnitude
806 axes_vec_ph = [wmin, wmax, -180, 0];

```

```

807                                     % Window to be used for phase
808                                     % See plotbode.m under the folder "
809                                     SDToolbox" for documentation
810     if include_ct
811         sys_cell = {Lec_cell{1,wbcount}
812                     Led_cell{: ,wbcount}};
813         lgd_text = {'CT design' lgd_cell
814                     {: ,wbcount}};
815         wvec_cell = [{wvec_c};
816                     wvec_d_cell{: ,wbcount}];
817     else
818         sys_cell = {Led_cell{: ,wbcount}};
819         lgd_text = {lgd_cell{: ,wbcount}};
820         wvec_cell = wvec_d_cell{: ,wbcount
821                                 )
822     end
823     plottypes = [1;1];
824     ttl_cell = {'Loop L Magnitude
825                 Response' titlemod],[ 'Loop L Phase
826                 Response' titlemod]};
827     axes_cell = {axes_vec_mag ,
828                 axes_vec_ph};
829
830     plotbode(sys_cell , wvec_cell ,

```

```

        plottypes , ttl_cell , lgd_text ,
        axes_cell , figcount);

823
824 % SAVE FIGURE
825 filename = ['L_mag' filenamemod];
826 if savefigs
827     savepdf(figcount , relpath , filename
828             );
829 end
830
831 figcount = figcount + 1;           %
832     Increment figure counter
833
834 % SAVE FIGURE
835 filename = ['L_ph' filenamemod];
836 if savefigs
837     savepdf(figcount , relpath , filename
838             );
839 end
840
841 figcount = figcount + 1;           %
842     Increment figure counter
843
844 % *****
845 %

```

```

842 % gamma * |W-1|
843 %
844
845 % See plotbode.m under the folder "
      SDToolbox" for documentation
846 sys_cell = {gamma_Winv_cell{: , wbcoun
      }};
847 plottypes = [1;0];
848 ttl_cell = {[ '\gamma |W-1|'
      titlemod]};
849 lgd_text = {lgd_cell{: , wbcoun}};
850 axes_cell = {};
851
852 plotbode(sys_cell , wvec_d_cell(: ,
      wbcoun) , plottypes , ttl_cell ,
      lgd_text , axes_cell , figcount);
853
854 figcount_sen = figcount;
855
856 % SAVE FIGURE
857 filename = [ '\gamma-W_inv' filenamemod
      ];
858 if savefigs
859     savepdf(figcount , relpath , filename
      );

```

```

860     end
861
862     figcount = figcount + 1;           %
           Increment figure counter
863
864     % *****
865     %
866     % SENSITIVITY FREQUENCY RESPONSE
867     %
868
869     axes_vec_mag = [wmin,wmax,-35,5];
           % Window to be used for
           magnitude
870
871     % See plotbode.m under the folder "
           SDToolbox" for documentation
872     if include_ct
873         sys_cell = {Sec_cell{1,wbcount}
           Sed_cell{: ,wbcount}};
874         lgd_text = {'CT design' lgd_cell
           {: ,wbcount}};
875         wvec_cell = [{wvec_c};
           wvec_d_cell{: ,wbcount}];
876     else
877         sys_cell = {Sed_cell{: ,wbcount}};

```



```

878         lgd_text = {lgd_cell{: , wbcount}};
879         wvec_cell = wvec_d_cell(:, wbcount
            )
880     end
881     plottypes = [1;0];
882     ttl_cell = {'Sensitivity S' titlemod
        ]};
883     axes_cell = {axes_vec_mag};
884
885     plotbode(sys_cell , wvec_cell ,
        plottypes , ttl_cell , lgd_text ,
        axes_cell , figcount);
886
887     % SAVE FIGURE
888     filename = ['S' filenamemod];
889     if savefigs
890         savepdf(figcount , relpath , filename
            );
891     end
892
893     figcount = figcount + 1;           %
        Increment figure counter
894
895     % *****
896     %

```

```

897 % COMPLEMENTARY SENSITIVITY FREQUENCY
      RESPONSE
898 %
899
900 axes_vec_mag = [wmin,wmax,-40,5];
      % Window to be used for
      magnitude
901
902 % See plotbode.m under the folder "
      SDToolbox" for documentation
903 if include_ct
904     sys_cell = {Tec_cell{1,wbcount}
      Ted_cell{: ,wbcount}};
905     lgd_text = {'CT design' lgd_cell
      {: ,wbcount}};
906     wvec_cell = [{wvec_c};
      wvec_d_cell{: ,wbcount}];
907 else
908     sys_cell = {Ted_cell{: ,wbcount}};
909     lgd_text = {lgd_cell{: ,wbcount}};
910     wvec_cell = wvec_d_cell{: ,wbcount
      )
911 end
912 plottypes = [1;0];
913 ttl_cell = {'Complementary

```

```

          Sensitivity T' titlemod]}};
914 axes_cell = {axes_vec_mag};
915
916 plotbode(sys_cell , wvec_cell ,
          plottypes , ttl_cell , lgd_text ,
          axes_cell , figcount);
917
918
919 % SAVE FIGURE
920 filename = ['T' filenamemod];
921 if savefigs
922     savepdf(figcount , relpath , filename
923            );
924
925 end
926
927
928 figcount = figcount + 1;          %
929     Increment figure counter
930
931
932 % *****
933 %
934 % REFERENCE TO CONTROL Tru
935 %
936
937 axes_vec_mag = [wmin , wmax , -40 , 15];

```

```

934         % Window to be used for
          magnitude
935
936 % See plotbode.m under the folder "
          SDToolbox" for documentation
937
938 if include_ct
          sys_cell = {Truc_cell{1,wbcount}
                    Trud_cell{: ,wbcount}};
          lgd_text = {'CT design' lgd_cell
                    {: ,wbcount}};
939          wvec_cell = [{wvec_c};
                      wvec_d_cell{: ,wbcount}];
940 else
          sys_cell = {Trud_cell{: ,wbcount}
                    };
          lgd_text = {lgd_cell{: ,wbcount}};
941          wvec_cell = wvec_d_cell{: ,wbcount
942                               )
943
944 end
945
946 plottypes = [1;0];
947 ttl_cell = {'Reference to Control:
948             T_{ru}' titlemod}];
949 axes_cell = {axes_vec_mag};
950
951 plotbode(sys_cell , wvec_cell ,

```

```

        plottypes , ttl_cell , lgd_text ,
        axes_cell , figcount);

950
951 % SAVE FIGURE
952 filename = ['Tru' filenamemod];
953 if savefigs
954     savepdf(figcount , relpath , filename
955            );
956
957 figcount = figcount + 1;      %
958     Increment figure counter
959
960 end                          % END for hcount = 1:
961
962     length_hvec
963
964 %%
965 %
966 % PLOTS — CONTINUOUS
967 %
968
969 % *****

```



```

993 %      titlemod = [ ' | h = ' num2str(h) '
      sec '];
994      titlemod = '';
995
996 % *****
997 %
998 % CONTROLLER FREQUENCY RESPONSE
999 %
1000
1001 axes_vec_mag = [wmin,wmax,-40,30];
      % Window to be used for
      magnitude
1002 axes_vec_ph = [wmin,wmax,-120,0];
      % Window to be used for phase
1003
1004 % See plotbode.m under the folder "
      SDToolbox" for documentation
1005 sys_cell = {Kc_cell{:},wbcount}};
1006 plottypes = [1;1];
1007 ttl_cell = {[ 'Controller K Magnitude
      Response' titlemod],[ 'Controller K
      Phase Response' titlemod]};
1008 lgd_text = {lgd_cell{:},wbcount}};
1009 axes_cell = {axes_vec_mag,
      axes_vec_ph};

```

```

1010
1011     plotbode(sys_cell , wvec , plottypes ,
           ttl_cell , lgd_text , axes_cell ,
           figcount);
1012
1013     % SAVE FIGURE
1014     filename = ['K_mag' filenamemod];
1015     if savefigs
1016         savepdf(figcount , relpath , filename
           );
1017     end
1018
1019     figcount = figcount + 1;           %
           Increment figure counter
1020
1021     % SAVE FIGURE
1022     filename = ['K_ph' filenamemod];
1023     if savefigs
1024         savepdf(figcount , relpath , filename
           );
1025     end
1026
1027     figcount = figcount + 1;           %
           Increment figure counter
1028

```



```

1029 % *****
1030 %
1031 % LOOP FREQUENCY RESPONSE
1032 %
1033
1034 axes_vec_mag = [wmin,wmax,-40,40];
           % Window to be used for
           magnitude
1035 axes_vec_ph = [wmin,wmax,-180,0];
           % Window to be used for phase
1036
1037 % See plotbode.m under the folder "
           SDToolbox" for documentation
1038 sys_cell = {Lec_cell{:},wbcount}};
1039 plottypes = [1;1];
1040 ttl_cell = {[ 'Loop L Magnitude
           Response' titlemod],[ 'Loop L Phase
           Response' titlemod]};
1041 lgd_text = {lgd_cell{:},wbcount}};
1042 axes_cell = {axes_vec_mag,
           axes_vec_ph};
1043
1044 plotbode(sys_cell, wvec, plottypes,
           ttl_cell, lgd_text, axes_cell,
           figcount);

```

```

1045
1046 % SAVE FIGURE
1047 filename = ['L_mag' filenamemod];
1048 if savefigs
1049     savepdf(figcount , relpath , filename
1050             );
1051 end
1052 figcount = figcount + 1; %
1053     Increment figure counter
1054 % SAVE FIGURE
1055 filename = ['L_ph' filenamemod];
1056 if savefigs
1057     savepdf(figcount , relpath , filename
1058             );
1059 end
1060 figcount = figcount + 1; %
1061     Increment figure counter
1062 % *****
1063 %
1064 % gamma * |W(-1)|
1065 %

```

```

1066
1067 % See plotbode.m under the folder "
      SDToolbox" for documentation
1068 sys_cell = {gamma_Winv_cell{: , wbcoun
      }};
1069 plottypes = [1;0];
1070 ttl_cell = {[ '\gamma |W^{-1}| '
      titlemod]};
1071 lgd_text = {lgd_cell{: , wbcoun}};
1072 axes_cell = {};
1073
1074 plotbode(sys_cell , wvec , plottypes ,
      ttl_cell , lgd_text , axes_cell ,
      figcount);
1075
1076 figcount_sen = figcount;
1077
1078 % SAVE FIGURE
1079 filename = [ '\gamma_W_inv' filenamemod
      ];
1080 if savefigs
1081     savepdf(figcount , relpath , filename
      );
1082 end
1083

```

```

1084     figcount = figcount + 1;           %
        Increment figure counter

1085
1086     % *****
1087     %
1088     % SENSITIVITY FREQUENCY RESPONSE
1089     %
1090
1091     axes_vec_mag = [wmin,wmax,-30,10];
        % Window to be used for
        magnitude

1092
1093     % See plotbode.m under the folder "
        SDToolbox" for documentation

1094     sys_cell = {Sec_cell{:},wbcount}};
1095     plottypes = [1;0];
1096     ttl_cell = {'Sensitivity S' titlemod
        ]};
1097     lgd_text = {lgd_cell{:},wbcount}};
1098     axes_cell = {axes_vec_mag};
1099
1100     plotbode(sys_cell , wvec , plottypes ,
        ttl_cell , lgd_text , axes_cell ,
        figcount);

1101

```

```

1102 % SAVE FIGURE
1103 filename = ['S' filenamemod];
1104 if savefigs
1105     savepdf(figcount , relpath , filename
1106             );
1107
1108     figcount = figcount + 1; %
1109     Increment figure counter
1110
1111 % *****
1112 % COMPLEMENTARY SENSITIVITY FREQUENCY
1113 % RESPONSE
1114
1115 axes_vec_mag = [wmin, wmax, -40, 10];
1116 % Window to be used for
1117 % magnitude
1118
1119 % See plotbode.m under the folder "
1120 % SDToolbox" for documentation
1121 sys_cell = {Tec_cell{:}, wbcnt};
1122 plottypes = [1;0];
1123 ttl_cell = {'Complementary

```

```

        Sensitivity T' titlemod]};
1121   lgd_text = {lgd_cell{: ,wbcount}}};
1122   axes_cell = {axes_vec_mag};
1123
1124   plotbode(sys_cell , wvec , plottypes ,
            ttl_cell , lgd_text , axes_cell ,
            figcount);
1125
1126
1127   % SAVE FIGURE
1128   filename = ['T' filenamemod];
1129   if savefigs
1130       savepdf(figcount , relpath , filename
            );
1131   end
1132
1133   figcount = figcount + 1;           %
            Increment figure counter
1134
1135
1136   % *****
1137   %
1138   % REFERENCE TO CONTROL Tru
1139   %
1140

```

```

1141 axes_vec_mag = [wmin, wmax, -40, 10];
           % Window to be used for
           magnitude
1142
1143 % See plotbode.m under the folder "
           SDToolbox" for documentation
1144 sys_cell = {Truc_cell{:}, wbcnt};
1145 plottypes = [1; 0];
1146 ttl_cell = {'Reference to Control:
           T_{ru}' titlemod};
1147 lgd_text = {lgd_cell{:}, wbcnt};
1148 axes_cell = {axes_vec_mag};
1149
1150 plotbode(sys_cell, wvec, plottypes,
           ttl_cell, lgd_text, axes_cell,
           figcount);
1151
1152 % SAVE FIGURE
1153 filename = ['Tru' filenamemod];
1154 if savefigs
1155     savepdf(figcount, relpath, filename
           );
1156 end
1157
1158 figcount = figcount + 1;           %

```

Increment figure counter

```
1159
1160
1161 end                                % END for hcount = 1:
    length_hvec
1162
1163 end
1164
1165
1166 %%
1167
1168 % *****
1169 % *****
1170 %
1171 % SD SIMULATIONS
1172 %
1173 % *****
1174 % *****
1175
1176 % *****
1177 %
1178 % BOOLEAN CONTROL TO DISABLE/ENABLE THIS
    SECTION
1179 %
1180 dosim = 1;
```



```

1181
1182 if dosim
1183
1184 % *****
1185 %
1186 % STORAGE
1187 %
1188 t_simulink_cell = cell(length_hvec ,
1189                        length_wbvec);    % Holds time data
1190
1191 y_simulink_cell = cell(length_hvec ,
1192                        length_wbvec);    % Holds output
1193                                         data
1194
1195 tu_simulink_cell = cell(length_hvec ,
1196                         length_wbvec);    % Holds time data
1197                                         for control signal
1198
1199 u_simulink_cell = cell(length_hvec ,
1200                        length_wbvec);    % Holds control
1201                                         signal data
1202
1203
1204
1205 % *****
1206 %
1207 % SIMULATIONS
1208 %

```

```

1199
1200 % *****
1201 %
1202 % MULTIPLIER OF RECIPROCAL OF BANDWIDTH
      TO STOP AT FOR SIMULATION TIME
1203 %
1204 % stop time = multiplier * 1 / wb
1205 %
1206 sim_stoptime_mult = 5; %
      changed from 5 to 0.1 on 11/28/2020,
      changed back to 5 on 12/5/2020
1207
1208 for wbcount = 1:length_wbvec
1209
1210 % *****
1211 %
1212 % CURRENT BANDWIDTH wb
1213 %
1214 wb = wbvec(wbcount);
1215
1216 % *****
1217 %
1218 % SET SIMULATION STOP TIME (sec)
1219 %
1220 sim_stoptime = sim_stoptime_mult * 1/

```

```

        wb;
1221 %     sim_stoptime = 5;
1222     set_param( 'SD_sim' , 'StopTime' ,
                num2str(sim_stoptime));
1223
1224     for hcount = 1:length_hvec
1225
1226         % *****
1227         %
1228         % EXTRACT CONTROLLER
1229         %
1230         Kd = Kd_cell{hcount , wbcount};
1231
1232         % *****
1233         %
1234         % RUN SIMULINK SIMULATION
1235         %
1236         h = hvec(hcount);
1237         simOut = sim( 'SD_sim' );           %
                Run simulation
1238
1239         % *****
1240         %
1241         % SAVE OUTPUT DATA
1242         %

```

```

1243         t_simulink_cell{hcount , wbcount} =
                simulink.y_simulink.Time;
1244         y_simulink_cell{hcount , wbcount} =
                simulink.y_simulink.Data;

1245
1246         tu_simulink_cell{hcount , wbcount}
                = simulink.u_simulink.Time;
1247         u_simulink_cell{hcount , wbcount} =
                simulink.u_simulink.Data;

1248
1249
1250     end                % END for bwfraccount
                = 1:length_bwfracvec

1251
1252 end                % END for hcount = 1:
                length_hvec

1253
1254
1255
1256 %%
1257
1258 % *****
1259 % *****
1260 %
1261 % CT SIMULATIONS

```

```

1262 %
1263 % *****
1264 % *****
1265
1266
1267 % *****
1268 %
1269 % STORAGE
1270 %
1271 t_cell = cell(length_hvec , length_wbvec);
           % Holds time data
1272 y_cell = cell(length_hvec , length_wbvec);
           % Holds output data
1273
1274 tu_cell = cell(length_hvec , length_wbvec)
           ;      % Holds time data for control
           signal
1275 u_cell = cell(length_hvec , length_wbvec);
           % Holds control signal data
1276
1277
1278 % *****
1279 %
1280 % SIMULATIONS
1281 %

```

```

1282
1283 for wbcnt = 1:length_wbvec
1284
1285     % *****
1286     %
1287     % CURRENT BANDWIDTH wb
1288     %
1289     wb = wbvec(wbcnt);
1290
1291     % *****
1292     %
1293     % SET SIMULATION STOP TIME (sec)
1294     %
1295     sim_stoptime = sim_stoptime_mult * 1/
1296     wb;
1297     %     sim_stoptime = 5;
1298
1299     for hcount = 1:length_hvec
1300
1301         % Output response
1302         [y_cell{hcount, wbcnt}, t_cell{
1303             hcount, wbcnt}] = step(
1304             Tec_cell{hcount, wbcnt},
1305             sim_stoptime);
1306
1307     end
1308 end

```

```

1303         % Control response
1304         [ u_cell{hcount , wbcount} , tu_cell{
                hcount , wbcount} ] = step(
                Truc_cell{hcount , wbcount} ,
                sim_stoptime);

1305
1306
1307
1308     end                % END for bwfraccount
                = 1:length_bwfracvec

1309
1310 end                % END for hcount = 1:
                length_hvec

1311
1312
1313 %%
1314
1315 % *****
1316 % *****
1317 %
1318 % PLOT TIME-DOMAIN RESPONSES — DT
1319 %
1320 % *****
1321 % *****
1322

```

```

1323 % *****
1324 %
1325 % BOOLEAN CONTROL TO PLOT ONE CT DESIGN
      AT BEGINNING OF EACH PLOT
1326 %
1327 % If include_ct = true, then the first
      continuous-time design corresponding
1328 % to the current value of wb (i.e., the
      lowest h-value design for the
1329 % current value of wb) will be plotted in
      front of each of the DT designs
1330 %
1331 include_ct = 1;
1332
1333
1334 for wbcnt = 1:length_wbvec
1335
1336     % *****
1337     %
1338     % CURRENT BANDWIDTH wb
1339     %
1340     wb = wbvec(wbcnt);
1341
1342     % *****
1343     %

```



```

1344 % FILE NAME/PLOT TITLE MODIFIERS
1345 %
1346 % Appended to the end of each file
      name to make names unique
1347 %
1348     filenamemod = [ ' — wb eq ' strrep(
          num2str(wb), '.', 'p') ' DT' ];
1349 %     titlemod = [ ' | h = ' num2str(h) '
sec '];
1350     titlemod = '';
1351
1352 % *****
1353 %
1354 % STEP RESPONSE — OUTPUT y
1355 %
1356
1357     figure(figcount)
1358
1359     if include_ct
1360
1361         t_plot = t_cell{1,wbcount};
1362         y_plot = y_cell{1,wbcount};
1363
1364         h_fig = plot(t_plot, y_plot);
1365         set(h_fig, 'LineWidth', 2);

```

```

1366         hold on
1367
1368     end
1369
1370     for hcount = 1:length_hvec
1371
1372         h_fig = plot(t_simulink_cell{
                    hcount ,wbcount} ,
                    y_simulink_cell{hcount ,wbcount
                    });
1373         set(h_fig , 'LineWidth' ,2);
1374
1375         if hcount == 1
1376             hold on;
1377         end
1378
1379     end % END for bwfraccount
        = 1:length_bwfracvec
1380
1381 % FORMATTING
1382 grid on;
1383 title(['Output Response y - SD
        Simulation' titlemod])
1384 if include_ct
1385     lgd_text = {'CT design' lgd_cell

```

```

        {:,wbcount}};
1386     else
1387         lgd_text = lgd_cell {:,wbcount};
1388     end
1389     lgd = legend(lgd_text);
1390     set(lgd, 'Location', 'Best');           %
        Put legend in empty spot
1391     xlabel('Time (s)')
1392     ylabel('y(t)')
1393
1394     % SAVE FIGURE
1395     filename = ['y_simulink' filenamemod
        ];
1396     if savefigs
1397         savepdf(figcount, relpath, filename
        );
1398     end
1399
1400     figcount = figcount + 1;           %
        Increment figure counter
1401
1402     % *****
1403     %
1404     % STEP RESPONSE — CONTROL u
1405     %

```

```

1406
1407     figure(figcount)
1408
1409     if include_ct
1410
1411         t_plot = tu_cell{1,wbcount};
1412         u_plot = u_cell{1,wbcount};
1413
1414         h_fig = plot(t_plot , u_plot);
1415         set(h_fig , 'LineWidth' ,2);
1416         hold on
1417
1418     end
1419
1420     for hcount = 1:length_hvec
1421
1422         h_fig = stairs(tu_simulink_cell{
1423             hcount ,wbcount} ,
1424             u_simulink_cell{hcount ,wbcount
1425             });
1426         set(h_fig , 'LineWidth' ,2);
1427
1428         if hcount == 1
1429             hold on;
1430         end

```

```

1428
1429     end                                % END for bwfraccount
        = 1:length_bwfracvec
1430
1431 % FORMATTING
1432 grid on;
1433 title(['Control Response u - SD
        Simulation' titlemod])
1434 lgd = legend(lgd_text);
1435 set(lgd, 'Location', 'Best');          %
        Put legend in empty spot
1436 xlabel('Time (s)')
1437 ylabel('u(t)')
1438
1439 % SAVE FIGURE
1440 filename = ['u_simulink' filenamemod
        ];
1441 if savefigs
1442     savepdf(figcount, relpath, filename
        );
1443 end
1444
1445 figcount = figcount + 1;              %
        Increment figure counter
1446

```

```

1447 end % END for hcount = 1:
        length_hvec
1448
1449
1450 %%
1451
1452 % *****
1453 % *****
1454 %
1455 % PLOT TIME-DOMAIN RESPONSES — CT
1456 %
1457 % *****
1458 % *****
1459
1460
1461 for wccount = 1:length_wbvec
1462
1463     % *****
1464     %
1465     % CURRENT BANDWIDTH wb
1466     %
1467     wb = wbvec(wccount);
1468
1469     % *****
1470     %

```

```

1471 % FILE NAME/PLOT TITLE MODIFIERS
1472 %
1473 % Appended to the end of each file
      name to make names unique
1474 %
1475 filenamemod = [ ' — wb eq ' strrep(
      num2str(wb), '.', 'p') ' DT' ];
1476 % titlemod = [ ' | h = ' num2str(h) '
sec '];
1477 titlemod = '';
1478
1479 % *****
1480 %
1481 % STEP RESPONSE — OUTPUT y
1482 %
1483
1484 figure(figcount)
1485
1486 for hcount = 1:length_hvec
1487
1488     t_plot = t_cell{hcount,wbcount};
1489     y_plot = y_cell{hcount,wbcount};
1490
1491     h_fig = plot(t_plot, y_plot);
1492     set(h_fig, 'LineWidth', 2);

```

```

1493
1494         if hcount == 1
1495             hold on;
1496         end
1497
1498     end                % END for bwfraccount
1499
1500     % FORMATTING
1501     grid on;
1502     title(['Output Response y' titlemod])
1503     lgd = legend(lgd_cell{:}, wbcnt);
1504     set(lgd, 'Location', 'Best');      %
1505         Put legend in empty spot
1506     xlabel('Time (s)')
1507     ylabel('y(t)')
1508     xlim([0, t_plot(end)])
1509
1510     % SAVE FIGURE
1511     filename = ['y_ct' filenamemod];
1512     if savefigs
1513         savepdf(figcount, relpath, filename
1514             );
1515     end
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600

```



```

1515     figcount = figcount + 1;           %
        Increment figure counter

1516

1517     % *****
1518     %
1519     % STEP RESPONSE — CONTROL u
1520     %
1521
1522     figure(figcount)
1523
1524     for hcount = 1:length_hvec
1525
1526         t_plot = tu_cell{hcount,wbcount};
1527         u_plot = u_cell{hcount,wbcount};
1528
1529         h_fig = plot(t_plot, u_plot);
1530         set(h_fig, 'LineWidth', 2);
1531
1532         if hcount == 1
1533             hold on;
1534         end
1535
1536     end           % END for bwfraccount
        = 1:length_bwfracvec
1537

```

```

1538 % FORMATTING
1539 grid on;
1540 title(['Control Response u' titlemod
        ])
1541 lgd = legend(lgd_cell{:}, wbcnt);
1542 set(lgd, 'Location', 'Best'); %
        Put legend in empty spot
1543 xlabel('Time (s)')
1544 ylabel('u(t)')
1545 xlim([0, t_plot(end)])
1546
1547 % SAVE FIGURE
1548 filename = ['u_ct' filenamemod];
1549 if savefigs
1550     savepdf(figcount, relpath, filename
            );
1551 end
1552
1553 figcount = figcount + 1; %
        Increment figure counter
1554
1555 end % END for hcount = 1:
        length_hvec
1556
1557 end

```

```

1558
1559
1560 %%
1561
1562
1563 % *****
1564 % *****
1565 %
1566 % H-INFINITY EQUIVALENT GENERALIZED PLANT
      G_{eq , d} FREQUENCY RESPONSES
1567 %
1568 % *****
1569 % *****
1570
1571 % *****
1572 %
1573 % H-INFINITY EQUIVALENT GENERALIZED PLANT
      G_{eq , d} MAGNITUDE RESPONSE
1574 %
1575
1576 plotbode_Geqd = 0;
1577
1578 if plotbode_Geqd
1579
1580 for i = 1:size(Geqd,1)

```

```

1581
1582     for j = 1:size(Geqd,2)
1583
1584         G_temp = Geqd(i,j);
1585
1586         figure(figcount);
1587
1588         systemname = [ 'G_{eq,d} ( '
1589                     num2str(i) ', ' num2str(j) ' ) '
1590                     ];
1591
1592         % See plotbode.m under the folder
1593         % "SDToolbox" for documentation
1594         sys_cell = {G_temp};
1595         plottypes = [1;0];
1596         ttl_cell = {[ '| ' systemname '(e^{
1597                     j\omega h) | ' ]};
1598         lgd_text = {};
1599         axes_cell = {};
1600
1601         plotbode(sys_cell, wvec_d,
1602                 plottypes, ttl_cell, lgd_text,
1603                 axes_cell, figcount);
1604
1605         xline(wn, '-.', '\omega_n', ...

```

```

1600         'LineWidth',1.25, '
           LabelOrientation','horizontal'
           ,...
1601         'FontWeight','bold','FontSize'
           ,15);
1602
1603     % SAVE FIGURE
1604     filename = [systemname '_mag'];
1605     if savefigs
1606         savepdf(figcount,relpath,
                 filename);
1607     end
1608
1609     figcount = figcount + 1;           %
           Increment figure counter
1610
1611     end
1612
1613 end
1614
1615
1616 % *****
1617 %
1618 % CLOSED LOOP SYSTEM MAGNITUDE RESPONSE
1619 %

```

```

1620
1621 for i = 1:size(Teqd,1)
1622
1623     for j = 1:size(Teqd,2)
1624
1625         T_temp = Teqd(i,j);
1626
1627         figure(figcount);
1628
1629         systemname = ['T_{eq,d w_-'
1630                     num2str(j) 'z_-' num2str(i) '}'
1631                     ];
1632
1633         % See plotbode.m under the folder
1634         % "SDToolbox" for documentation
1635
1636         sys_cell = {T_temp};
1637         plottypes = [1;0];
1638         ttl_cell = {'|' systemname '(e^{
1639                   j\omega h})|'}];
1640         lgd_text = {};
1641         axes_cell = {};
1642
1643         plotbode(sys_cell, wvec_d,
1644                 plottypes, ttl_cell, lgd_text,
1645                 axes_cell, figcount);

```

```

1639
1640     xline(wn, '-', '\omega_n', ...
1641         'LineWidth', 1.25, '
            LabelOrientation', 'horizontal'
            , ...
1642         'FontWeight', 'bold', 'FontSize'
            , 15);

1643
1644     % SAVE FIGURE
1645     filename = [systemname '_mag'];
1646     if savefigs
1647         savepdf(figcount, relpath,
            filename);
1648     end

1649
1650     figcount = figcount + 1;           %
            Increment figure counter

1651
1652     end

1653
1654     end

1655
1656     end

1657
1658     % *****

```

```

1659 %
1660 % H-INFINITY EQUIVALENT GENERALIZED PLANT
      G_{eq,d} SINGULAR VALUES
1661 %
1662
1663 plotsvs = 1;
1664
1665 if plotsvs
1666
1667     for wbcnt = 1:length_wbvec
1668
1669         % *****
1670         %
1671         % CURRENT BANDWIDTH wb
1672         %
1673         wb = wbvec(wbcnt);
1674
1675         % *****
1676         %
1677         % FILE NAME/PLOT TITLE MODIFIERS
1678         %
1679         % Appended to the end of each file
            name to make names unique
1680         %
1681         filenamemod = [ ' ' ];

```



```

1682 %      titlemod = [ ' | h = ' num2str(h) '
      sec '];
1683      titlemod = '';
1684
1685 % *****
1686 %
1687 % DETERMINE X-AXIS LIMITS
1688 %
1689 tmp = wvec_d_cell{1,1};
1690 wmin = tmp(1);
1691 tmp = wvec_d_cell{1,end};
1692 wmax = tmp(end);
1693
1694 % *****
1695 %
1696 % LOOP ON h
1697 %
1698 figure(figcount)
1699
1700 for hcount = 1:length_hvec
1701
1702     % Extract frequency vector
1703     wvec_d = wvec_d_cell{hcount,
      wbcnt};
1704

```

```

1705     % Calculate singular values
1706     sv_s = sigma(Geqd_cell{hcount,
                    wbcount}, wvec_d);
1707     sv_max = max(sv_s);    % Max sv_s
1708     sv_min = min(sv_s);    % Min sv_s
1709
1710     % Append data together for one
                    plot entry
1711     wvec_d_ap = [wvec_d nan wvec_d];
1712     sv_s_ap = [sv_min nan sv_max];
1713
1714     h_fig = semilogx(wvec_d_ap , 20*
                    log10(sv_s_ap));
1715     set(h_fig , 'LineWidth' ,2);
1716
1717     if hcount == 1
1718         hold on;
1719     end
1720
1721     end                % END for bwfraccount
                    = 1:length_bwfracvec
1722
1723     % FORMATTING
1724     grid on;
1725     title('Max. and Min. Singular Values

```

```

        of G_{eq,d}', 'fontsize',20)
1726   lgd = legend(lgd_cell{:},wbcount});
1727   set(lgd, 'Location', 'Best');           %
        Put legend in empty spot
1728   xlabel('Frequency (rad/s)', 'fontsize'
        ,20)
1729   ylabel('Singular Values (dB)', '
        fontsize',20)
1730   xlim([wmin,wmax])
1731   set(gca, 'fontsize',20);
1732
1733   % SAVE FIGURE
1734   filename = ['Geqd_svs_vs_h'
        filenamemod];
1735   if savefigs
1736       savepdf(figcount, relpath, filename
        );
1737   end
1738
1739   figcount = figcount + 1;           %
        Increment figure counter
1740
1741
1742 end                                     % END for hcount = 1:
        length_hvec

```

```

1743
1744
1745 end
1746
1747
1748 %%
1749
1750 % *****
1751 % *****
1752 %
1753 % H-INFINITY EQUIVALENT GENERALIZED PLANT
      G_{eq , d} CHARACTERISTICS
1754 %
1755 % *****
1756 % *****
1757
1758
1759 for wbcnt = 1:length_wbvec
1760
1761     % *****
1762     %
1763     % CURRENT BANDWIDTH wb
1764     %
1765     wb = wbvec(wbcnt);
1766

```

```

1767 % *****
1768 %
1769 % FILE NAME/PLOT TITLE MODIFIERS
1770 %
1771 % Appended to the end of each file
      name to make names unique
1772 %
1773     filenamemod = [ '' ];
1774 %     titlemod = [ ' | h = ' num2str(h) '
      sec '];
1775     titlemod = '';
1776
1777 % *****
1778 %
1779 % NUMBER OF INPUTS VS. h
1780 %
1781     figure(figcount);
1782 %     h_fig = semilogx(hvec,
      Geqd_numin_mat(:,wbcount), 'bo',...
1783 %         hvec, Geqd_numin_mat(:,wbcount)
      , 'b--');
1784 %     set(h_fig(1), 'LineWidth', 2);
1785 %     set(h_fig(2), 'LineWidth', 2);
1786     h_fig = semilogx(hvec, Geqd_numin_mat
      (:,wbcount));

```

```

1787     set(h_fig , 'LineWidth' ,2);
1788     hold on;
1789
1790     grid on;
1791     title('Number of Inputs of G-{eq,d}
           vs. h')
1792 %     lgd = legend(lgd_cell{:},wbcount);
1793 %     set(lgd , 'Location' , 'Best ');
           % Put legend in empty spot
1794     xlabel('h (sec)')
1795     ylabel('Number of Inputs')
1796     ylim([3,5])
1797
1798
1799 % SAVE FIGURE
1800     filename = ['Geqd_numin_vs_h'
                 filenamemod];
1801     if savefigs
1802         savepdf(figcount , relpath , filename
                 );
1803     end
1804
1805     figcount = figcount + 1;           %
           Increment figure counter
1806

```

```

1807
1808 % *****
1809 %
1810 % NUMBER OF OUTPUTS VS. h
1811 %
1812 figure(figcount);
1813 %     h_fig = semilogx(hvec,
1814 %           Geqd_numout_mat(:,wbcount), 'bo',...
1815 %           hvec, Geqd_numout_mat(:,wbcount
1816 %           ), 'b--');
1817 %     set(h_fig(1), 'LineWidth',2);
1818 %     set(h_fig(2), 'LineWidth',2);
1819 h_fig = semilogx(hvec,
1820 %           Geqd_numout_mat(:,wbcount));
1821 set(h_fig, 'LineWidth',2);
1822 hold on;
1823
1824 grid on;
1825 title('Number of Outputs of G_{eq,d}
1826 %           vs. h')
1827 %     lgd = legend(lgd_cell(:,wbcount));
1828 %     set(lgd, 'Location', 'Best');
1829 %     % Put legend in empty spot
1830 xlabel('h (sec)')
1831 ylabel('Number of Outputs')

```

```

1827     ylim([3,5])
1828
1829
1830     % SAVE FIGURE
1831     filename = ['Geqd_numout_vs_h'
1832                filenamemod];
1833     if savefigs
1834         savepdf(figcount , relpath , filename
1835                );
1836     end
1837
1838     figcount = figcount + 1;           %
1839     Increment figure counter
1840
1841     % *****
1842     %
1843     % NUMBER OF INPUTS AND OUTPUTS VS. h
1844     %
1845     figure(figcount);
1846     h_fig = semilogx(hvec , Geqd_numin_mat
1847                     (: , wbcnt));
1848     set(h_fig , 'LineWidth' , 2);
1849     hold on;
1850     h_fig = semilogx(hvec ,

```



```

        Geqd_numout_mat (: , wbcount));
1848 set(h_fig , 'LineWidth' ,2);
1849
1850 grid on;
1851 title('Number of Inputs and Outputs
        of G_{eq,d} vs. h')
1852 lgd = legend('Inputs' , 'Outputs');
1853 set(lgd , 'Location' , 'Best');           %
        Put legend in empty spot
1854 xlabel('h (sec)')
1855 ylabel('Number of Inputs/Outputs')
1856 ylim([3 ,5])
1857
1858
1859 % SAVE FIGURE
1860 filename = ['Geqd_numin_numout_vs_h'
        filenamemod];
1861 if savefigs
1862     savepdf(figcount , relpath , filename
        );
1863 end
1864
1865 figcount = figcount + 1;           %
        Increment figure counter
1866

```

```

1867
1868 % *****
1869 %
1870 % ORDER OF GENERALIZED PLANT  $G_{\{eq,d\}}$ 
      VS.  $h$ 
1871 %
1872 figure(figcount);
1873 h_fig = semilogx(hvec, Geqd_order_mat
      (:,wbcount));
1874 set(h_fig, 'LineWidth',2);
1875 hold on;
1876
1877 grid on;
1878 title('Order of  $G_{\{eq,d\}}$  vs.  $h$ ')
1879 xlabel('h (sec)')
1880 ylabel('Order of  $G_{\{eq,d\}}$ ')
1881
1882
1883 % SAVE FIGURE
1884 filename = ['Geqd_order_vs_h'
      filenamemod];
1885 if savefigs
1886     savepdf(figcount, relpath, filename
      );
1887 end

```

```

1888
1889     figcount = figcount + 1;           %
        Increment figure counter

1890
1891
1892     % *****
1893     %
1894     % ORDER OF MINIMUM REALIZATION OF
        GENERALIZED PLANT minreal(G_{eq,d
        }) VS. h
1895     %
1896     figure(figcount);
1897     h_fig = semilogx(hvec,
        Geqd_minreal_order_mat(:,wbcount))
        ;
1898     set(h_fig, 'LineWidth',2);
1899     hold on;
1900
1901     grid on;
1902     title('Order of minreal(G_{eq,d}) vs.
        h')
1903     xlabel('h (sec)')
1904     ylabel('Order of minreal(G_{eq,d})')
1905
1906

```

```

1907 % SAVE FIGURE
1908 filename = [ 'Geqd_minreal_order_vs_h'
               filenamemod ];
1909 if savefigs
1910     savepdf(figcount , relpath , filename
             );
1911 end
1912
1913 figcount = figcount + 1; %
               Increment figure counter
1914
1915
1916
1917 % *****
1918 %
1919 % OPTIMAL PERFORMANCE gamma_opt VS. h
1920 %
1921 figure(figcount);
1922 %     h_fig = semilogx(hvec , gamma_d_mat
               (: , wbcount) , 'bo' , ...
1923 %         hvec , gamma_d_mat (: , wbcount) , '
               b—');
1924 %     set(h_fig(1) , 'LineWidth' , 2);
1925 %     set(h_fig(2) , 'LineWidth' , 2);
1926 h_fig = semilogx(hvec , gamma_d_mat (: ,

```

```

        wbcount));
1927 set(h_fig, 'LineWidth', 2);
1928 hold on;
1929
1930 grid on;
1931 title('Optimal Performance  $\gamma_{\text{opt}}$  vs. h')
1932 % lgd = legend(lgd_cell(:, wbcount));
1933 % set(lgd, 'Location', 'Best');
        % Put legend in empty spot
1934 xlabel('h (sec)')
1935 ylabel('\gamma_{opt}')
1936 % ylim([2, 6])
1937
1938
1939 % SAVE FIGURE
1940 filename = ['gamma_opt_vs_h'
        filenamemod];
1941 if savefigs
1942     savepdf(figcount, relpath, filename
        );
1943 end
1944
1945 figcount = figcount + 1; %
        Increment figure counter

```

```

1946
1947
1948
1949
1950
1951
1952 end                                % END for hcount = 1:
        length_hvec
1953
1954 %% Generate discrete controllers using
        the Bilinear approximation (c2d with "
        tustin")
1955
1956 Kz_cell = cell(length_hvec , length_wbvec)
        ;      % Holds Discretized controllers
        (Bilinear approximation)
1957 Kz2_cell = cell(length_hvec , length_wbvec
        );      % Holds Discretized controllers
        (Direct Design)
1958
1959 % *****
1960 % *****
1961 %
1962 % H-INFINITY CONTROLLERS DISCRETIZED
        USING THE BILINEAR APPROXIMATION

```

```

1963 %
1964 % *****
1965 % *****
1966
1967 for wbcount = 1:length_wbvec
1968
1969     % *****
1970     %
1971     % CURRENT BANDWIDTH wg
1972     %
1973     wb = wbvec(wbcount);
1974
1975
1976     for hcount = 1:length_hvec
1977
1978         % *****
1979         %
1980         % CURRENT SAMPLING PERIOD h
1981         %
1982         h = hvec(hcount);
1983
1984
1985         % *****
1986         %
1987         % DISCRETIZE THE ANALOG

```

```

                                CONTROLLER FOR THIS h USING
                                THE BILINEAR
1988    % APPROXIMATION
1989    %
1990    [A, B, C, D] = ssdata(Kc);
1991    [Abt, Bbt, Cbt, Dbt] = f_bilin(A,
                                B, C, D, h);
1992    %           Kz_cell{hcount,wbcount} = c2d(
                                Kc, h, 'tustin');
1993    Kz_cell{hcount,wbcount} = ss(Abt,
                                Bbt, Cbt, Dbt, h);
1994    Kz2_cell{hcount,wbcount} =
                                direct_design(Gc, h);
1995
1996    end
1997
1998
1999    end
2000    %% SD Time Domain Simulations using "
                                sdsim" and Induced-L2 Norm
2001
2002    %
2003    % CREATE VECTORS FOR THE NORMS
2004    %
2005

```



```

2006
2007
2008 % close all
2009 figcount = figcount + 1;          %
      Increment figure counter
2010 %
2011 % "sdlsim": Time response of sampled-data
      feedback system
2012 %
2013
2014 % Create a string to be used as the
      legend
2015 str = [];
2016 for hcount = 1:length_hvec
2017     temp = [ 'h = ', num2str(hvec(hcount)
      .*1e6), ' us ' ];
2018     str{hcount} = temp;
2019 end
2020
2021 close all;
2022
2023 H1 = [];
2024 H2 = [];
2025 H3 = [];
2026 for hcount = 1:length_hvec

```

```

2027
2028     fprintf( '\n\n ITERATION: %3.2d OF
           %3.2d \n\n', hcount, length_hvec);
2029 % Store the hcountth controller
2030 Kd_temp = Kd_cell{hcount};
2031 Kz_temp = Kz_cell{hcount};
2032 Kz2_temp = Kz2_cell{hcount};
2033
2034 % Store the hcountth time vector and
           final time
2035 t_temp = t_simulink_cell{hcount};
2036 t_final = t_temp(end);
2037
2038 % Reference signal
2039 ref = ones(length(t_temp), 1);
2040
2041 % Generalized plant: r → y
2042 Gry = [ 0,      P      ;
          F,      -F*P   ] ;
2043
2044 % -----
2045 [n_xp, ~] = size(Ap);
2046 [n_xf, ~] = size(Af);
2047 nw = 1;
2048 nz = 1;
2049 nu = 1;

```

```

2050 % -----
2051 A_ry = [ Ap,      zeros(n_xp, n_xf)
          ;
          -Bf*Cp, Af      ] ;
2052
2053 % -----
2054 B1_ry = [ zeros(n_xp, nw) ;
          Bf      ] ;
2055
2056 B2_ry = [ Bp      ;
          -Bf*Dp   ] ;
2057
2058 B_ry = [ B1_ry, B2_ry ] ;
2059
2060 % -----
2061 C1_ry = [ Cp,      zeros(nz, n_xf)
          ];
2062
2063 C2_ry = [ -Df*Cp, Cf
          ];
2064
2065 % -----
2066 D11_ry = [ zeros(nz, nw) ] ;
2067
2068 D12_ry = [ Dp      ] ;
2069
2070 D21_ry = [ Df      ] ;
2071
2072 D22_ry = [ -Df*Dp   ] ;
2073
2074 D_ry = [ D11_ry, D12_ry
          ;
          D21_ry, D22_ry ] ;

```

```

                ;
2072                D21_ry ,      D22_ry  ]
                ;
2073 % -----
2074 G_ry = ss(A_ry , B_ry , C_ry , D_ry);
2075
2076 %      G_ry IS CORRECT !!!
2077 %      disp('CHECK THE MODELS:');
2078 %      disp('CORRECT MODEL FOR COMPARISON
:');
2079 %      zpk(Gry)
2080 %      disp('MODEL DERIVED:');
2081 %      zpk(G_ry)
2082 %
2083 %      load handel
2084 %      sound(y,Fs)
2085
2086 % Generalized plant: r -> u
2087 one = 6e3/(s+6e3);
2088 Wr = ss(one);
2089 [Ar, Br, Cr, Dr] = ssdata(Wr);
2090 [n_xr, ~] = size(Ar);
2091 % -----
2092 Gru = [ 0,      one      ;
2093         F,      -F*P    ] ;

```

```

2094 % -----
2095 A_ru = [ Ap, zeros
           (n_xp, n_xf), zeros(n_xp, n_xr)
           ;
           -Bf*Cp, Af,
           zeros(
           n_xf, n_xr) ;
2097 zeros(n_xr, n_xp), zeros
           (n_xr, n_xf), Ar
           ] ;

2098 % -----
2099 B1_ru = [ zeros(n_xp, nw) ;
2100           Bf ;
2101           zeros(n_xr, nw) ] ;
2102 B2_ru = [ Bp ;
2103           -Bf*Dp ;
2104           Br ] ;
2105 B_ru = [ B1_ru, B2_ru ] ;
2106 % -----
2107 C1_ru = [ zeros(nz, n_xp),
           zeros(nz, n_xf), Cr
           ];
2108 C2_ru = [ -Df*Cp,
           Cf, zeros(ny, n_xr)
           ];

```

```

2109
2110 C_ru = [ C1_ru ;
2111           C2_ru ] ;
2112 % -----
2113
2114 D11_ru = [ zeros(nz, nw) ] ;
2115 D12_ru = [ zeros(nz, nu) ]
           ;
2116 D21_ru = [ Df ] ;
2117 D22_ru = [ -Df*Dp ] ;
2118
2119 D_ru = [ D11_ru, D12_ru
           ;
2120         D21_ru, D22_ru ]
           ;
2121 % -----
2122 G_ru = ss(A_ru, B_ru, C_ru, D_ru);
2123
2124 % G_ru IS CORRECT !!!
2125 % disp('CHECK THE MODELS:');
2126 % disp('CORRECT MODEL FOR COMPARISON
:');
2127 % zpk(G_ru)
2128 % disp('MODEL DERIVED:');
2129 % zpk(G_ru)

```

```

2130 %
2131 %     load handel
2132 %     sound(y,Fs)
2133
2134 % Generalized plant: r -> e
2135 Gre = [ Wr,      -Wr*P      ;
2136         F,      -F*P      ] ;
2137 % -----
2138 A_re = [  Ap,          zeros
2139         (n_xp, n_xf),  zeros(n_xp, n_xr)
2140         ;
2141         -Bf*Cp,       Af,
2142         zeros(
2143             n_xf, n_xr)      ;
2144         zeros(n_xr, n_xp),  zeros
2145         (n_xr, n_xf),  Ar
2146         ] ;
2147 % -----
2148 B1_re = [  zeros(n_xp, nw)      ;
2149           Bf                    ;
2150           Br                    ] ;
2151 B2_re = [  Bp                    ;
2152           -Bf*Dp                 ;
2153           zeros(n_xr, nu)       ] ;
2154 B_re = [ B1_re,  B2_re  ] ;

```

```

2149 % -----
2150 C1_re = [ zeros(nz, n_xp),
           zeros(nz, n_xf), Cr
           ];
2151 C2_re = [ -Df*Cp,
           Cf, zeros(ny, n_xr)
           ];
2152
2153 C_re = [ C1_re ;
2154          C2_re ] ;
2155 % -----
2156
2157 D11_re = [ zeros(nz, nw) ] ;
2158 D12_re = [ zeros(nz, nu) ]
           ;
2159 D21_re = [ Df ] ;
2160 D22_re = [ -Df*Dp ] ;
2161
2162 D_re = [ D11_re, D12_re
           ;
           D21_re, D22_re ]
           ;
2163
2164 % -----
2165 G_re = ss(A_re, B_re, C_re, D_re);
2166

```



```

2167 %      G_re IS CORRECT !!!
2168 %      disp('CHECK THE MODELS:');
2169 %      disp('CORRECT MODELS POLES FOR
          COMPARISON:');
2170 %      POLES_orig = eig(minreal(G_re))
2171 %      disp('MODEL DERIVED, POLES:');
2172 %      POLES_derived = eig(minreal(G_re))
2173 %      disp('ERROR:');
2174 %      ERROR(1) = POLES_orig(1) -
          POLES_derived(2);
2175 %      ERROR(2) = POLES_orig(2) -
          POLES_derived(3);
2176 %      ERROR(3) = POLES_orig(3) -
          POLES_derived(1);
2177 %      ERROR(4) = POLES_orig(4) -
          POLES_derived(4);
2178 %      disp(ERROR)
2179 %
2180 %      load handel
2181 %      sound(y, Fs)
2182 %      return
2183
2184 disp(['For h = ', num2str( hvec(hcount
          )*1e6 ), ' us, the induced L2 Norms
          are: ']);

```

```

2185 % COMPUTE THE INDUCED L2 NORM FOR Try
2186 fprintf( '\n Try: \n' );
2187 [GAML,GAMU] = sdhinfnorm(G_ry ,
      Kz_temp);
2188 norm = (GAML+GAMU)/2;
2189 disp(['(Bilinear) Induced L2 Norm of
      T: w → yp is ', num2str( norm )]);
2190 Try_norm_bilin(hcount) = norm;
2191
2192 [GAML,GAMU] = sdhinfnorm(G_ry ,
      Kz2_temp);
2193 norm = (GAML+GAMU)/2;
2194 disp(['(Bilinear) Induced L2 Norm of
      T: w → yp is ', num2str( norm )]);
2195 Try_norm_dd(hcount) = norm;
2196
2197 [GAML,GAMU] = sdhinfnorm(G_ry ,
      Kd_temp);
2198 norm = (GAML+GAMU)/2;
2199 disp(['(SD) Induced L2 Norm of T: w
      → yp is ', num2str( norm )]);
2200 Try_norm_sd(hcount) = norm;
2201
2202 fprintf( '\n Tru: \n' );
2203 [GAML,GAMU] = sdhinfnorm(G_ru ,

```

```

    Kz_temp);
2204 norm = (GAML+GAMU)/2;
2205 disp(['(Bilinear) Induced L2 Norm of
    T: w → u is ', num2str( norm )]);
2206 Tru_norm_bilin(hcount) = norm;
2207
2208 [GAML,GAMU] = sdhinfnorm(G_ru,
    Kd_temp);
2209 norm = (GAML+GAMU)/2;
2210 disp(['(SD) Induced L2 Norm of T: w
    → u is ', num2str( norm )]);
2211 Tru_norm_sd(hcount) = norm;
2212
2213 fprintf('\n Tre: \n');
2214 [GAML,GAMU] = sdhinfnorm(G_re,
    Kz_temp);
2215 norm = (GAML+GAMU)/2;
2216 disp(['(Bilinear) Induced L2 Norm of
    T: w → e is ', num2str( norm )]);
2217 Tre_norm_bilin(hcount) = norm;
2218
2219 [GAML,GAMU] = sdhinfnorm(G_re,
    Kd_temp);
2220 norm = (GAML+GAMU)/2;
2221 disp(['(SD) Induced L2 Norm of T: w

```

```

        -> e is ', num2str( norm )]);
2222 Tre_norm_sd(hcount) = norm;
2223
2224 % Time response of SD System from
        MATLAB (Plot the SD Time Response
2225 % (using "sdlsim") and store the data
        )
2226 [VT,~,~,~] = sdlsim(Gry, Kd_temp, ref
        , t_temp, t_final);
2227 t_ry = VT{1};
2228 y_t = VT{2};
2229
2230 [VT,~,~,~] = sdlsim(Gre, Kd_temp, ref
        , t_temp, t_final);
2231 t_re = VT{1};
2232 e_t = VT{2};
2233
2234 % [VT,~,~,~] = sdlsim(Gru, Kd_temp,
        ref, t_temp, t_final);
2235 % t_ru = VT{1};
2236 % u_t = VT{2};
2237
2238
2239 % Plot the time domain responses
2240 figure(figcount);

```

```

2241 h1 = plot(t_re , e_t , ':' , 'linewidth'
           , 2);
2242 grid on; hold on;
2243 xlabel('Time (sec)', 'fontsize', 20);
2244 ylabel('Error: e(t)', 'fontsize', 20);
2245 title('SD Time Response: Error (
           sdsim)', 'fontsize', 20);
2246 set(gca, 'fontsize', 20);
2247 H1 = [H1, h1];
2248
2249 figure(figcount+1);
2250 h2 = plot(t_ry , y_t , ':' , 'linewidth'
           , 2);
2251 grid on; hold on;
2252 xlabel('Time (sec)', 'fontsize', 20);
2253 ylabel('Output: y(t)', 'fontsize', 20)
           ;
2254 title('SD Time Response: Output (
           sdsim)', 'fontsize', 20);
2255 set(gca, 'fontsize', 20);
2256 legend(h2, str{hcount});
2257 H2 = [H2, h2];
2258
2259 % figure(figcount+2);
2260 % h3 = plot(t_ru , u_t , ':' , '

```

```

        linewidth ', 2);
2261 %     grid on; hold on;
2262 %     xlabel('Time (sec)', 'fontsize ', 20)
        ;
2263 %     ylabel('Control: u(t)', 'fontsize ',
        20);
2264 %     title('SD Time Response: Control (
        sdsim)', 'fontsize ', 20);
2265 %     set(gca, 'fontsize ', 20);
2266 %     legend(h3, str{hcount});
2267 %     H3 = [H3, h3];
2268
2269
2270 end
2271 legend(H1, str);
2272 legend(H2, str);
2273 legend(H3, str);
2274
2275 %% Continuous-time System and H-infinity
        Norms
2276
2277 npts = length(Tre_norm_bilin);           %
        NUMBER OF POINTS
2278
2279 % CONTINUOUS-TIME CLOSED-LOOP MAPS

```

```

2280 Try = lft (Gry, Kc);
2281 Tre = lft (Gre, Kc);
2282 Tru = lft (Gru, Kc);
2283
2284 % H-INFINITY NORMS
2285 Try_hinf_norm = hinfnorm(Try);
2286 Try_hinf_norm = Try_hinf_norm * ones(1,
      npts);
2287 Tre_hinf_norm = hinfnorm(Tre);
2288 Tre_hinf_norm = Tre_hinf_norm * ones(1,
      npts);
2289 Tru_hinf_norm = hinfnorm(Tru);
2290 Tru_hinf_norm = Tru_hinf_norm * ones(1,
      npts);
2291 %% Plot the induced L2 Norms
2292
2293 hvec_us = hvec * 1e6;
2294
2295 figure ( figcount+2);
2296 plot (hvec_us , Tre_norm_bilin , 'b:' , '
      linewidth' , 2);
2297 grid on; hold on;
2298 plot (hvec_us , Tre_hinf_norm , 'g-' , '
      linewidth' , 2);
2299 grid on; hold on;

```

```

2300 plot(hvec_us, Tre_norm_sd, 'r—', '
        linewidth', 2);
2301 xlabel('h (in usec)', 'fontsize', 19);
2302 ylabel('||T_{re}||_{L2}', 'fontsize', 19)
        ;
2303 title('Induced-L2 Norm of Sensitivity, |
        Tre|', 'fontsize', 19);
2304 set(gca, 'fontsize', 19);
2305 legend('bilin', 'HINF norm', 'sd');
2306
2307 figure(figcount+3);
2308 plot(hvec_us, Tru_norm_bilin, 'b:', '
        linewidth', 2);
2309 grid on; hold on;
2310 plot(hvec_us, Tru_hinf_norm, 'g-', '
        linewidth', 2);
2311 grid on; hold on;
2312 plot(hvec_us, Tru_norm_sd, 'r—', '
        linewidth', 2);
2313 xlabel('h (in usec)', 'fontsize', 19);
2314 ylabel('||T_{ru}||_{L2}', 'fontsize', 19)
        ;
2315 title('Induced-L2 Norm of Sensitivity, |
        Tru|', 'fontsize', 19);
2316 set(gca, 'fontsize', 19);

```



```

2317 legend('bilin', 'HINF norm', 'sd');
2318
2319 figure(figcount+4);
2320 plot(hvec_us, Try_norm_bilin, 'b:', '
        linewidth', 2);
2321 grid on; hold on;
2322 plot(hvec_us, Try_norm_dd, 'r:', '
        linewidth', 2);
2323 grid on; hold on;
2324 plot(hvec_us, Try_norm_sd, 'g—', '
        linewidth', 2);
2325 grid on; hold on;
2326 plot(hvec_us, Try_hinf_norm, 'k:', '
        linewidth', 2);
2327 xlabel('h (in usec)', 'fontsize', 19);
2328 ylabel('||T- $\{ry\}$ ||- $\{L2\}$ , ||T- $\{ry\}$ ||- $\{H-$ 
        Inf}', 'fontsize', 19);
2329 title('Induced-L2 Norm of Closed-loop
        System, |Try|', 'fontsize', 19);
2330 set(gca, 'fontsize', 19);
2331 legend('Indirect Design', 'Direct Design'
        , 'Lifting-based Design', 'HINF norm')
        ;
2332
2333 load handel

```

```

2334     sound(y, Fs)
2335
2336 % save('Data to Compare - Aug 9 - 2021',
        'Tre_norm_bilin');
2337
2338 %% Closed-loop SD Time Responses
2339
2340 % CHECK Try TOO !!!
2341
2342 % MEANING OF THE SUBSCRIPTS:
2343 % z -> bilinear transformation of the
        cont.-time controller
2344 % d -> direct-sd controller design
2345
2346 % Switching Frequency: 4.5455 kHz, h =
        220 us (unstable)
2347 h_unstable = hvec(end);
2348 Kz_unstable = Kz_cell{end};
2349 Kz2_unstable = Kz2_cell{end};
2350 Kd_unstable = Kd_cell{end};
2351
2352 [ET, ~, ~, ~] = sdlsim(Gre, Kz_unstable, ref
        , t_temp, t_final);
2353 t_z_unstable = ET{1};
2354 error_z_unstable_t = ET{2};

```

```

2355
2356 [ET,~,~,~] = sdlsim(Gre, Kz2_unstable,
        ref, t_temp, t_final);
2357 t_z2_unstable = ET{1};
2358 error_z2_unstable_t = ET{2};
2359
2360 [ET,~,~,~] = sdlsim(Gre, Kd_unstable, ref
        , t_temp, t_final);
2361 t_d_unstable = ET{1};
2362 error_d_unstable_t = ET{2};
2363
2364 figure(figcount+6);
2365 plot(t_z_unstable.*1e3,
        error_z_unstable_t, 'b:', 'linewidth',
        2);
2366 grid on; hold on;
2367 plot(t_z2_unstable.*1e3,
        error_z2_unstable_t, 'r:', 'linewidth'
        , 2);
2368 grid on; hold on;
2369 plot(t_d_unstable.*1e3,
        error_d_unstable_t, 'g—', 'linewidth'
        , 2);
2370 xlabel('t (in msec)', 'fontsize', 19);
2371 ylabel('e(t)', 'fontsize', 19);

```

```

2372 title('Error, e(t). h = 220 \mus, f_{sw}
        = 4.5455 kHz', 'fontsize', 19);
2373 set(gca, 'fontsize', 19);
2374 legend('Indirect Design', 'Direct Design'
        , 'Lifting-based Design');
2375
2376 % Switching Frequency: 100 kHz, h = 10 us
        (stable)
2377 h_stable = hvec(1);
2378 Kz_stable = Kz_cell{1};
2379 Kz2_stable = Kz2_cell{1};
2380 Kd_stable = Kd_cell{1};
2381
2382 [ET,~,~,~] = sdlsim(Gre, Kz_stable, ref,
        t_temp, t_final);
2383 t_z_stable = ET{1};
2384 error_z_stable_t = ET{2};
2385
2386 [ET,~,~,~] = sdlsim(Gre, Kz2_stable, ref,
        t_temp, t_final);
2387 t_z2_stable = ET{1};
2388 error_z2_stable_t = ET{2};
2389
2390 [ET,~,~,~] = sdlsim(Gre, Kd_stable, ref,
        t_temp, t_final);

```

```

2391 t_d_stable = ET{1};
2392 error_d_stable_t = ET{2};
2393
2394 figure(figcount+7);
2395 plot(t_z_stable.*1e3, error_z_stable_t, '
        b:', 'linewidth', 2);
2396 grid on; hold on;
2397 plot(t_z2_stable.*1e3, error_z2_stable_t,
        'r:', 'linewidth', 2);
2398 grid on; hold on;
2399 plot(t_d_stable.*1e3, error_d_stable_t, '
        g—', 'linewidth', 2);
2400 xlabel('t (in msec)', 'fontsize', 19);
2401 ylabel('e(t)', 'fontsize', 19);
2402 title('Error, e(t). h = 10 \mus, f_{sw} =
        100 kHz', 'fontsize', 19);
2403 set(gca, 'fontsize', 19);
2404 legend('Indirect Design', 'Direct Design'
        , 'Lifting-based Design');
2405
2406 %% COMPUTE % DECREASE IN POWER LOSS
2407
2408 P_start = 716.9; % W,
        see ACC 2022 paper draft for
        computation

```

```

2409 P_final = 32.3; % W,
      see ACC 2022 paper draft for
      computation

2410

2411 percent_dec = 100 * (P_start - P_final)/
      abs(P_start);

2412

2413 fprintf('\n\n There was a %3.2d %%
      decrease in the power dissipation \n\n
      ', percent_dec);

2414

2415 %% ANALYZE THE OPEN LOOP OF THE
      CONTINUOUS-TIME SYSTEM

2416

2417 Ls = series(Kc,P); %
      Open loop (continuous-time)

2418 margin = allmargin(Ls); %
      Margins (including UGC)

2419 BW_w = margin.PMFrequency; % UGC
      (in rad/s)

2420 BW_f = BW_w/(2* pi); % UGC
      (in Hz)

2421

2422 fprintf('\n
      _____\

```

```

        n');
2423 fprintf( '\n OPEN LOOP BANDWIDTH (
        CONTINUOUS-TIME) = %3.2f kHz \n', BW_f
        *1e-3);
2424 fprintf( '\n OPEN LOOP PHASE MARGIN (
        CONTINUOUS-TIME) = %3f deg \n', margin
        .PhaseMargin);

2425
2426 w_vec = logspace(1, 8, 20e3);

2427
2428 [mag,ph] = bode(Ls,w_vec);
2429 mag = squeeze(mag(1,1,:));
2430 mag = transpose(mag);
2431 mag = mag2db(mag);
2432 ph = squeeze(ph(1,1,:));
2433 ph = transpose(ph);

2434
2435 figure(figcount+8);
2436 Prepfig(15);
2437 subplot(2,1,1);
2438 semilogx(w_vec,mag); grid on; ylabel('
        Magnitude (in dB)', 'fontsize', 19);
        grid on;
2439 set(gca,'fontsize',19);set(gca,'
        GridLineStyle','-', 'linewidth',2);

```

```

2440 axis([1e1 1e8 -50 50])
2441 hold on;
2442 title('Open Loop (Lo)');
2443 subplot(2,1,2);
2444 semilogx(w_vec,ph); grid on; ylabel('
        Phase (in deg)', 'fontsize', 19); grid
        on;
2445 set(gca, 'fontsize', 19); set(gca, '
        GridLineStyle', '-', 'linewidth', 2);
2446 h1 = findobj(gcf, 'type', 'line');
2447 set(h1, 'linewidth', 3);
2448 xlabel('Frequency (rad/s)', 'fontsize',
        19);
2449
2450 w_vec = logspace(0, 6, 20e3);
2451
2452 [mag, ~] = bode(W, w_vec);
2453 mag = squeeze(mag(1,1,:));
2454 mag = transpose(mag);
2455 mag = mag2db(mag);
2456 [mag2, ~] = bode(inv(W), w_vec);
2457 mag2 = squeeze(mag2(1,1,:));
2458 mag2 = transpose(mag2);
2459 mag2 = mag2db(mag2);
2460

```



```

2461 figure(figcount+9);
2462 Prepfig(15);
2463 semilogx(w_vec,mag); grid on; ylabel('
    Magnitude (in dB)', 'fontsize', 19);
    grid on;
2464 set(gca, 'fontsize',19);set(gca, '
    GridLineStyle', '- ', 'linewidth', 2);
2465 hold on;
2466 semilogx(w_vec,mag2); grid on; ylabel('
    Magnitude (in dB)', 'fontsize', 19);
    grid on;
2467 set(gca, 'fontsize',19);set(gca, '
    GridLineStyle', '- ', 'linewidth', 2);
2468 title('Weight');
2469 set(gca, 'fontsize',19);set(gca, '
    GridLineStyle', '- ', 'linewidth', 2);
2470 h1 = findobj(gcf, 'type', 'line');
2471 set(h1, 'linewidth', 3);
2472 xlabel('Frequency (rad/s)', 'fontsize',
    19);
2473 legend('W(s)', 'W^{-1}(s)');

1 % m File: example13_7_2.m
2
3 clc
4 clear

```

```

5 close all
6 format short
7
8 % see Section 13.7 and Example 13.7.2
9 % | A | B1 B2 |
10 % |-----|
11 % | C1 | D11 D12 |
12 % | C2 | D21 D22 |
13
14 A = 0;
15 B1 = 1;
16 B2 = -1;
17 C1 = 1;
18 C2 = 1;
19 D11 = 0;
20 D12 = 0;
21 D21 = 0;
22 D22 = 0;
23
24 B = [B1 B2];
25 C = [C1; C2];
26 D = [D11 D12; D21 D22];
27
28 G = ss(A,B,C,D);
29

```

```

30 h = 1.5;
31 % Compute the norm of the Hilbert–Schmidt
      operator as a lower bound of the
32 % gamma vector
33 %
34 gamma_min = 1e-5; %
      Initial lower-bound estimate of ||
      D11_||
35 gamma_max = 5; %
      Initial upper-bound estimate of ||
      D11_||
36 tol_D11 = 1e-3; %
      Desired tolerance of calculation of
      ||D11_||
37
38 normD11_operator = norm_D11_(A, B1, C1, h
      , gamma_min, gamma_max, tol_D11);
39 disp(['Norm of the Hilbert–Schmidt
      operator (from the function) is, ||D11
      || = ', num2str(normD11_operator)]);
40 % pause
41
42 h = 1:0.01:1.89;
43
44 l2norm = NaN.* ones(1, length(h));

```

```

45 L2_Norm = NaN.*ones(1,length(h));
46
47 disp('USING THE L2 NORM FUNCTION:');
48 for ii = 1:length(h)
49     disp(' ');
50     disp(['h = ',num2str(h(ii))','
           'iteration number ',num2str(ii),'
           'of ',num2str(length(h))',' i.e. ',
           num2str(100*ii./length(h)),'%'])
           ;
51
52     Kd = tf(1,1,h(ii));
           %
           Controller for given value of h
53
54     % Compute the induced L2 norm using
           our function
55     [l2norm(ii),~] = sdl2norm ( G, Kd, h(
           ii) );
56
57     % Compute the induced L2 norm using
           MATLAB's function
58     [L2_Norm(ii),~] = sdhinfnorm ( G, Kd
           );
59 end

```

```

60
61 save('Data - Ex 13_7_2');
62 %%
63 %
64 % DATA FROM THE BOOK
65 %
66 % (1, 1), (1.1, 1), (1.2, 1), (1.3, 1.05)
        , (1.4, 1.4), (1.5, 1.6),
67 % (1.6, 2.1), (1.7, 3), (1.8, 5.1), (1.9,
        11)
68 Twz = [1.*ones(1,10), 1.*ones(1,10),
        1.05.*ones(1,10), 1.4.*ones(1,10),
        1.6.*ones(1,10), ...
69         2.1*ones(1,10), 3*ones(1,10),
        5.1*ones(1,10), 11*ones
        (1,10) ];
70
71 %
        *****
72 %
73 % PLOT THE DATA
74 %
75 figure(1);
76 plot(h, l2norm, 'b:', 'linewidth', 2);

```

```

77 hold on;
78 plot(h, L2_Norm, 'r—', 'linewidth', 2);
79 hold on;
80 % plot(h, Twz, 'k:', 'linewidth', 2);
81 plot(h, ones(1, length(h)), 'k—', '
      linewidth', 2);
82 ylabel('||T_{zw}||_{L2}', 'fontsize', 20)
      ;
83 xlabel('T_s', 'fontsize', 20);
84 title('Figure 13.11 on page 340', '
      fontsize', 20);
85 grid on;
86 set(gca, 'fontsize', 20);
87 lgd_text = {'Chen', 'Bamieh', 'Hinf Norm'};
88 lgd = legend(lgd_text);
89 set(lgd, 'Location', 'Best');           % Put
      legend in empty spot
90
91
92 indices = find(l2norm == -1);
93 h_check = h(indices);

```

APPENDIX B

B MATLAB CODE: BUCK CONVERTER RIPPLE

This m file computes the output voltage ripple for a vector of capacitor values. For each plant design, we estimate the plant ripple using a variety of methods as described in chapter 3.

```
1 % m File :  
    Compare_Our_Theory_VoltRipple_n_Terms .  
    m  
2  
3 clear ;  
4 close all ; clc ;  
5  
6 % Time Vector  
7 tic ;  
8  
9 tau = 2*330*10-6*10;  
10 Ts = 1/100000;  
11 n_samples = 1000;           % The  
    switching interval is sampled  
    n_samples time  
12 t_final = 15*tau;          %  
    Limiting the time vector to 0.099 sec  
13 dT = Ts/1000;  
14 t = 0:dT:t_final ;  
15 %sim('pulse_gen_simelec '); % Takes  
    866.282543 seconds .  
16 %save('Pulses');
```



```

17 load('Pulses');
18 toc;
19
20
21 % Compute the index for t = 0.12 sec
    where we want to compute the ripple
22 tic;
23
24 % Compute the Voltage Ripple from Our
    Theory (Simplified plant model, 3
25 % Harmonics)
26
27 D = 0.24;           % Duty Ratio
28 A = 5;             % Amplitude of the
    Square Wave/ Input DC voltage
29 % DC Component
30 C0 = A*D;         % DC Component of the
    Fourier Series that we want to pass
    through the Filter
31 % Switching Frequency
32 w0 = 2*pi*100000;
33
34 L = 380 * 10^-6;
35 fs = 100000;
36 Ts = 1/fs;

```

```

37 Vin = 5;
38 Vo = 1.2;
39 D = Vo/Vin;
40 dI = D*(1-D)*[(Vin*Ts)/L];
41 Ro = 10;
42
43 toc;
44
45 % C = (0.2:(25-0.2)/400:25) * 10^-6;
46 C = (0.2:(25-0.2)/10:25) * 10^-6;
47 n_vec = [3, 9, 12];
48
49
50 for ii = 1:length(C)
51     for jj = 1:length(n_vec)
52         tic;
53         %disp(ii);
54         fprintf('\nITERATION NUMBER %3.2d OF
55                 %3.2d', ii, length(C));
56         fprintf('\nITERATION NUMBER %3.2d OF
57                 %3.2d', jj, length(n_vec));
58
59         dV = (dI/C(ii))/(8*fs);
60         s = tf('s');
61         Simpl_LPF_Vo = [1/(L*C(ii))]/s^2;

```

```

60 %Simpl_LPF_IL = [1/(L)]/s;
61
62 DPVR_Trad(ii , jj) = dV*100/Vo;
63
64 n = n_vec(jj);
65                                     %
        NUMBER OF HARMONICS
65 % Generate the Voltage Output for n
        harmonics
66 LPF_output = C0;
67 for kk = 1:n
68 C_ = C0*(sin(kk*D*pi)/(kk*pi*D))*exp(-1i*
        kk*D*pi);
69 C_r = abs(C_);
70 C_theta = angle(C_);
71 [mv, phv] = bode(Simpl_LPF_Vo, kk*w0);
72 LPF_output = LPF_output + 2*C_r*mv*cos(kk
        *w0*t+C_theta+phv*(pi/180));
73 end
74
75 % Compute the Desired % Voltage Ripple (
        DPVR) from the New Theory
76 Volt_Ripple = LPF_output - C0;
77 Volt_Ripple = Volt_Ripple(end-1000:end);
        % We only want the voltage

```

```

        ripple for the last switching interval
78
79 V_max = max(Volt_Ripple);
80 V_min = min(Volt_Ripple);
81
82 DPVRF_T(ii , jj) = ((V_max-V_min)/1.2)
        *100;
83
84
85
86 %% Full SS Model of the Plant (Rl = 0 and
        Rc = 0) with Rectangular Wave Input
87
88 A_ss = [0 (-1/L); (1/C(ii)) -1/(C(ii)*Ro)
        ];
89 B_ss = [(1/L); 0];
90 C_ss = [0 1];
91 D_ss = 0;
92
93 P = ss(A_ss , B_ss , C_ss , D_ss);
94
95 %load('u_pulse');
96 % Using the Latest pulses 8/6/2016
97 % load('CorrectInput20Tau');
98 u = u_pulse;

```

```

99 [Y,~,~] = lsim(P,u,t);
100
101 volt_ripple = Y - C0;
102 volt_ripple = volt_ripple(end-1000:end);
           % We only want the voltage
           ripple for the last switching interval
103 v_max = max(volt_ripple);
104 v_min = min(volt_ripple);
105
106 APVR(ii , jj) = ((v_max-v_min)/1.2)*100;
107
108 %% Output of the Full S.S. Model with n
           Harmonics at the Input
109
110 % Generate the Voltage Output for n
           harmonics
111 LPF_output = C0;
112 for kk = 1:n
113 C_ = C0*(sin(kk*D*pi)/(kk*pi*D))*exp(-1i*
           kk*D*pi);
114 C_r = abs(C_);
115 C_theta = angle(C_);
116 [mv,phv] = bode(P, kk*w0);
117 LPF_output = LPF_output + 2*C_r*mv*cos(kk
           *w0*t+C_theta+phv*(pi/180));

```

```

118 end
119
120 % Compute the Desired % Voltage Ripple (
      DPVR) from the New Theory
121 Volt_Ripple = LPF_output - C0;
122 Volt_Ripple = Volt_Ripple(end-1000:end);
      % We only want the voltage
      ripple for the last switching interval
123
124 V_max = max(Volt_Ripple);
125 V_min = min(Volt_Ripple);
126
127 PLANT_FT(ii , jj) = ((V_max-V_min)/1.2)
      *100;      % in %
128
129 %% Compute the Errors Between the Actual
      and the Theories
130
131 % DIVIDE EVERYTHING BY 100 TO CONVERT %
      TO FRACTION
132 %Error1(jj) = ( (DPVR(jj)-DPVR2(jj)) );
133 Theory1(ii , jj) = DPVR_Trad(ii , jj)/100;
134 Theory2(ii , jj) = DPVR_FT(ii , jj)/100;
135 Theory3(ii , jj) = PLANT_FT(ii , jj)/100;
136 Actual(ii , jj) = APVR(ii , jj)/100;

```

```

137
138 % COMPUTE THE ERRORS
139 Error_Trad(ii , jj) = abs(Actual(ii , jj)-
    Theory1(ii , jj))*100/Actual(ii , jj);
140 Error_FT(ii , jj) = abs(Actual(ii , jj)-
    Theory2(ii , jj))*100/Actual(ii , jj);
141 Error_SS_FT(ii , jj) = abs(Actual(ii , jj)-
    Theory3(ii , jj))*100/Actual(ii , jj);
142
143 %% Output of the Full Model with 3H at
    the Input
144
145 %Vin = C0;
146 %for jj = 1:n;
147 %    C = C0*(sin(jj*D*pi)/(jj*pi*D))*exp
    (-1i*jj*D*pi);
148 %    C_r = abs(C);
149 %    C_theta = angle(C);
150 %    Vin = Vin + 2*C_r*cos(jj*w0*t+
    C_theta);
151 %end
152
153 %[Y2,~,~] = lsim(P,Vin,t);
154
155 %voltage_ripple2 = Y2 - C0;

```

```

156 %volt_ripple2 = volt_ripple2(end-1000:end
    );          % We only want the voltage
              ripple for the last switching interval
157 %v_max2 = max(volt_ripple2);
158 %v_min2 = min(volt_ripple2);
159
160 %DPVR_Full_Model_3H(ii) = ((v_max2-v_min2
    )/1.2)*100;
161
162 %Theory3(ii) = DPVR_Full_Model_3H(ii)
    /100;
163 %Error_3(ii) = abs(Actual(ii)-Theory3(ii)
    )*100/Actual(ii);
164
165 % Cap(ii , jj) = C(ii , jj)/10^-6;
166
167 fprintf('\n');
168 toc;
169     end
170 end
171
172 %% Save the Data
173 % save('
    Volt_Ripple_12H_Theory_400_Points_L_380
    ', 'DPVREM', 'DPVR_FT', 'APVR', 'u', 'Cap

```



```

        ' , ' Error_1 ' , ' Error_2 ' , ' Theory1 ' , '
        Theory2 ' );
174 %% Plot the Data
175 close all ;
176
177 figure (1) ;
178 loglog (APVR (: , 1) , Error_Trad (: , 1) , '
        LineWidth ' , 2) ;
179 grid on ;
180 xlabel ( ' Actual % Volt Ripple ' , ' FontSize '
        , 20) ;
181 ylabel ( '% Error (Voltage Ripple) ' , '
        FontSize ' , 20) ;
182 title ( '%Error = |(Actual - Theory)| | x
        100 / Actual ' );
183
184 figure (2) ;
185 loglog (APVR (: , 1) , Error_Trad (: , 1) , 'k' ,
        ' LineWidth ' , 2) ;
186 hold on ;
187 loglog (APVR (: , 1) , Error_FT (: , 1) , 'b: ' , '
        LineWidth ' , 2) ;
188 hold on ;
189 loglog (APVR (: , 1) , Error_FT (: , 2) , 'r: ' , '
        LineWidth ' , 2) ;

```

```

190 hold on;
191 loglog(APVR(:, 1), Error_FT(:, 3), 'g:', '
    LineWidth', 2);
192 grid on;
193 legend('Traditional Small Ripple theory',
    'Fourier (3 harmonics)', 'Fourier (9
    harmonics)', 'Fourier (12 harmonics)');
194 xlabel('Actual % Volt Ripple', 'FontSize'
    , 20);
195 ylabel('% Error (Voltage Ripple)', '
    FontSize', 20);
196 title('%Error = |(Actual - Theory)| x
    100 / Actual; (Approx. T.F.)');
197
198 figure(3);
199 loglog(APVR(:, 1), Error_Trad(:, 1), 'k',
    'LineWidth', 2);
200 hold on;
201 loglog(APVR(:, 1), Error_SS_FT(:, 1), 'b:'
    , 'LineWidth', 2);
202 hold on;
203 loglog(APVR(:, 1), Error_SS_FT(:, 2), 'r:'
    , 'LineWidth', 2);
204 hold on;
205 loglog(APVR(:, 1), Error_SS_FT(:, 3), 'g:'

```

```

        , 'LineWidth', 2);
206 grid on;
207 legend('Traditional Small Ripple theory',
        'Fourier (3 harmonics)', 'Fourier (9
        harmonics)', 'Fourier (12 harmonics)');
208 xlabel('Actual % Volt Ripple', 'FontSize',
        ,20);
209 ylabel('% Error (Voltage Ripple)', '
        FontSize', 20);
210 title('%Error = |(Actual - Theory)| x
        100 / Actual; (Exact S.S.)');
211
212 load handel
213 sound(y, Fs)

```

APPENDIX C

C MATLAB CODE: HIGHER ORDER FILTERS

This m file plots a family of filter Bode plots. The filters are 2nd (LC), 4th (LCLC) and 6th (LCLCLC) order filters. The frequency responses are obtained from Simulink by building each circuit in Simulink and then using MATLAB's *Frequency Response Estimation* functionality. The data are then stored in .MAT files and plotted in the script shown next.

```
1 % m File : HIGHER_ORDER_FILTERS.m
2
3 clc ;
4 clear ;
5 close all ;
6 %
7 % MODELS OF 4th AND 6th ORDER FILTERS
8 % HAVE BEEN STORED
9 %
10 % THESE WERE OBTAINED FROM SIMULINK FORM
11 % THE CIRCUIT BY USING A
12 % SMALL-SIGNAL PERTURBATION AT THE INPUT
13 % AND MEASURING THE OUTPUT IN ORDER
14 % TO GET THE FREQUENCY RESPONSE
15 %
16 %% TOTAL INDUCTANCE AND CAPACITANCE. LOAD
17 % RESISTANCE
```

```

17 Ro = 100;
18 L = 6.8e-9;
19 C = 4e-9;
20
21 %% 2nd ORDER FILTER
22
23 s = tf('s');
24
25 b = 1/(L*C);
26 a = 1/(C*Ro);
27 P_2nd_Order = b / (s^2 + a*s + b) ;
28
29 %% 4th ORDER FILTER
30
31 load('Linear 4th Order Model from
      Simulink Circuit.mat')
32 P_4th_Order = linsys1;
33
34 disp('4th Order Plant:');
35 zpk(P_4th_Order)
36
37 %% 6th ORDER FILTER
38
39 load('Linear 6th Order Model from
      Simulink Circuit.mat')

```

```

40 P_6th_Order = linsys1;
41
42 disp('6th Order Plant:');
43 zpk(P_6th_Order)
44
45 %% FAMILY OF BODE PLOTS
46
47 wvec = logspace(7, 10, 100e3);
48
49 % GENERATE THE MAGNITUDE AND PHASE
      RESPONSES FOR ALL 3 PLANTS:
50
51 % 2nd ORDER
52 [mag2, ph2] = bode(P_2nd_Order, wvec);
53 mag2 = squeeze(mag2(1, 1, :));
54 mag2 = transpose(mag2);
55 mag2 = mag2db(mag2);
56 ph2 = squeeze(ph2(1, 1, :));
57 ph2 = transpose(ph2);
58
59 % 4th ORDER
60 [mag4, ph4] = bode(P_4th_Order, wvec);
61 mag4 = squeeze(mag4(1, 1, :));
62 mag4 = transpose(mag4);
63 mag4 = mag2db(mag4);

```

```

64 ph4 = squeeze(ph4(1,1,:));
65 ph4 = transpose(ph4);
66
67 % 6th ORDER
68 [mag6,ph6] = bode(P_6th_Order,wvec);
69 mag6 = squeeze(mag6(1,1,:));
70 mag6 = transpose(mag6);
71 mag6 = mag2db(mag6);
72 ph6 = squeeze(ph6(1,1,:));
73 ph6 = transpose(ph6);
74
75 figure(1);
76 Prefig(15);
77 subplot(2,1,1);
78 semilogx(wvec, mag2); hold on; semilogx(
    wvec, mag4); hold on; semilogx(wvec,
    mag6);
79 grid on; ylabel('Magnitude (in dB)',
    'FontSize',19); grid on;
80 set(gca,'fontsize',15);set(gca,'
    GridLineStyle','-', 'linewidth',2);
81 hold on;
82 title('Buck Converter (Plant)');
83 subplot(2,1,2);
84 semilogx(wvec,ph2); hold on; semilogx(

```



```

        wvec, ph4); hold on; semilogx(wvec,
        ph6);
85 grid on; ylabel('Phase (in deg)', '
        FontSize',19); grid on;
86 set(gca, 'fontsize',15); set(gca, '
        GridLineStyle', '- ', 'linewidth',2);
87 h1 = findobj(gcf, 'type', 'line');
88 set(h1, 'linewidth',3);
89
90 legend('2nd Order (LC)', '4th Order (LCLC
        )', '6th Order (LCLCLC)');
91
92 %
93 % WHEN IS THE 2nd ORDER BEST ?
94 %
95 Indices2ndOrder = find(mag2 < mag4);
96 Index2ndOrder_Best = max(Indices2ndOrder)
        ;
97 w_2nd = wvec(Index2ndOrder_Best);
98 Best_2nd_Order_Mag = mag2(
        Index2ndOrder_Best);           % mag2
        is already in dB
99
100 disp(['2nd Order LC filters are best till
        ', num2str(w_2nd*1e-6), ' Mrad/s, i.e.

```

```

        ', num2str(w_2nd/(2*pi*1e6)), ' MHz']];
101 disp(['This corresponds to an attenuation
        of ', num2str(Best_2nd_Order_Mag), ' dB
        ']);
102 fprintf('\n\n');
103
104 %
105 % WHEN IS THE 4th ORDER BEST ?
106 %
107 Indices4thOrder = find(mag4 < mag6);
108 Index4thOrder_Best = max(Indices4thOrder)
        ;
109 w_4th = wvec(Index4thOrder_Best);
110 Best_4th_Order_Mag = mag4(
        Index4thOrder_Best);           % mag4
        is already in dB
111 disp(['4th Order LCLC filters are best
        till ', num2str(w_4th*1e-6), ' Mrad/s, i
        .e. ', num2str(w_4th/(2*pi*1e6)), ' MHz'
        ']);
112 disp(['This corresponds to an attenuation
        of ', num2str(Best_4th_Order_Mag), ' dB
        ']);

```

APPENDIX D

D MATLAB CODE: INVERTER INNER-OUTER LOOP

These m files design the inner and outer controllers for the inverter as described in

Chapter 5.

```
1 % m File: Design_Methodology.m
2
3 %% Clear the workspace data, clear the
   screen and cloase all plots
4 clear all; clc; close all;
5
6
7 %% Plant
8
9 s=tf('s');
10 w1=2*pi*60;
11 %wvec=logspace(-2,5,10000);
12 wvec=logspace(2,5,10000);
13
14 rl = 1;
15 wg = 2*pi*60;
16 En = 120*sqrt(3);
17 Vg = 120;
18 Pn = 5000;
19 In = Pn./(3*Vg);
20 Zbase = En^2./Pn;
21 Lbase = Zbase/wg;
22 Cbase = 1/(wg*Zbase);
```

```

23 %L = Lt * Lb;
24 Vdc = 400;
25 wg = 2*pi*60;
26 wsw = 2*pi*15000;
27 x = 10 * wg;
28 y = 0.5 * wsw;
29
30
31 % Resistances
32 R1 = 0;%10^-6;
33 R2 = 0;%10^-6;
34
35 L1 = (0.04951/2)*Lbase;
36 L2 = (0.04951/2)*Lbase;
37 Cd = 0.05*Cbase;
38 Rd = 0;%10^-6;
39
40 A = [-(Rd+R1)/L1   (Rd/L1)   (-1/L1);
41       (Rd/L2)   -(Rd+R2)/L2   (1/L2);
42       (1/Cd)   (-1/Cd)   0];
43 B = [(1/L1);
44       0;
45       0];
46 Bdo = [0;
47         (-1/L2);

```

```

48     0];
49 C = [0  1  0];
50 D = 0;
51
52 P = ss(A,B,C,D);
53
54 Vdc = 400;
55
56 P = (Vdc/2) * P;
57
58 disp('Poles and zeros of the Plant');
59 damp(pole(P))
60 damp(zero(P))
61
62 disp('Transfer Function of the Plant');
63 zpk(P)
64
65 % We get an incorrect phase plot
66 figure;
67 bode(P,wvec);
68 grid on;
69 opts=bodeoptions;
70 opts.InputLabels.FontSize=12;
71 opts.OutputLabels.FontSize=12;
72 opts.XLabel.FontSize=14;

```

```

73 opts.YLabel.FontSize=14;
74 opts.Title.FontSize=20;
75 h_line = findobj(gcf, 'type', 'line');
76 set(h_line, 'LineWidth',2);
77 h_axes = findobj(gcf, 'type', 'axes');
78 set(h_axes, 'LineWidth',1, 'FontSize',14, '
    GridAlpha',0.25);
79 title('Nominal Plant');
80
81 [mag,ph] = bode(P,wvec);
82 mag = squeeze(mag(1,1,:));
83 mag = transpose(mag);
84 mag = mag2db(mag);
85 ph = ph + 360;
86 ph = squeeze(ph(1,1,:));
87 ph = transpose(ph);
88
89 figure;
90 Prepfig(15);
91 subplot(2,1,1);
92 semilogx(wvec,mag); grid on; ylabel('
    Magnitude (in dB)'); grid on;
93 set(gca, 'fontsize',15);set(gca, '
    GridLineStyle','-', 'linewidth',2);
94 hold on;

```

```

95 title('Plant (P)');
96 subplot(2,1,2);
97 semilogx(wvec,ph); grid on; ylabel('Phase
    (in deg)'); grid on;
98 set(gca,'fontsize',15);set(gca,'
    GridLineStyle','-', 'linewidth',2);
99 h1 = findobj(gcf,'type','line');
100 set(h1,'linewidth',3);
101
102 [wn,zeta] = damp(P);
103 wres = wn(end);
104
105 %% Inner loop Controller
106
107 m = 20.86
108 k = 0.942
109 z = sqrt([(1-k^2)/(2*k^2)]/m)
110
111 wh = [sqrt(m)+(2/sqrt(m))] * wres * sqrt((1-k
    ^2)/2)
112 wn = k*wres
113
114 wh_again = (m+2)*z*wn
115
116 kc = [(L1+L2)*wh - m*z*wn^3*L1*L2*Cd]/[

```



```

Vdc/2]
117
118 Ki = kc*s/(s+wh)
119
120 disp('Poles and zeros of the Inner
      Controller ');
121 pole(Ki)
122 zero(Ki)
123
124 disp('Transfer Function of the Inner
      Controller ');
125 zpk(Ki)
126
127 %% Modified Plant
128
129 Pmod = feedback(P,Ki,+1);
130
131 disp('Poles and zeros of the modified
      Plant ');
132 damp(pole(Pmod))
133 damp(zero(Pmod))
134
135 disp('Transfer Function of the modified
      Plant ');
136 zpk(Pmod)

```

```

137
138 figure;
139 rlocus(Pmod);
140 grid off;
141 h_line = findobj(gcf, 'type', 'line');
142 set(h_line, 'LineWidth',2);
143
144 figure;
145 bode(Pmod, wvec);
146 grid on;
147 opts=bodeoptions;
148 opts.InputLabels.FontSize=12;
149 opts.OutputLabels.FontSize=12;
150 opts.XLabel.FontSize=14;
151 opts.YLabel.FontSize=14;
152 opts.Title.FontSize=20;
153 h_line = findobj(gcf, 'type', 'line');
154 set(h_line, 'LineWidth',2);
155 h_axes = findobj(gcf, 'type', 'axes');
156 set(h_axes, 'LineWidth',1, 'FontSize',14, '
    GridAlpha',0.25);
157 title('Modified Plant');
158
159
160 figure;

```

```

161 rlocus(Pmod);
162 grid off;
163 h_line = findobj(gcf, 'type', 'line');
164 set(h_line, 'LineWidth',2);
165 h_axes = findobj(gcf, 'type', 'axes');
166 set(h_axes, 'LineWidth',1, 'FontSize',14, '
    GridAlpha',0.25);
167 % title('Modified Plant');
168
169 %% Checking the peak of Pmod with more
    points
170
171 wvec1 = logspace(2,4,20000);
172 wvec2 = logspace(4,4.1,20000);
173 wvec3 = logspace(4.1,5,20000);
174
175 w = [wvec1, wvec2, wvec3];
176
177 figure;
178 bode(Pmod,w);
179 grid on;
180 opts=bodeoptions;
181 opts.InputLabels.FontSize=12;
182 opts.OutputLabels.FontSize=12;
183 opts.XLabel.FontSize=14;

```

```

184 opts.YLabel.FontSize=14;
185 opts.Title.FontSize=20;
186 h_line = findobj(gcf, 'type', 'line');
187 set(h_line, 'LineWidth',2);
188 h_axes = findobj(gcf, 'type', 'axes');
189 set(h_axes, 'LineWidth',1, 'FontSize',14, '
    GridAlpha',0.25);
190 title('Modified Plant (more points)');
191
192 %%% Li = P*Ki
193 %
194 % Li = series(P,Ki);
195 %
196 % figure;
197 % rlocus(Li);
198 % h_line = findobj(gcf, 'type', 'line');
199 % set(h_line, 'LineWidth',2);
200 % title('L_i');
201 % grid off;
202 %
203 % figure;
204 % rlocus(1-Li);
205 % h_line = findobj(gcf, 'type', 'line');
206 % set(h_line, 'LineWidth',2);
207 % title('1 - L_i');

```

```

208 % grid off;
209 %% Designing a PR Controller (Ayyanar, R
      2015, Current controller design for dc
      -ac stage of single phase PV inverters
      , lecture notes, Renewable Electric
      Energy Systems EEE598 Arizona State
      University, delivered October 2015)
210 wc = 1000 * (2*pi); % 1 kHz
211 [m, ~] = bode(Pmod, wc);
212 kp = 1/m
213 wl = 59.3 * (2*pi);
214 [Gp, ~] = bode(Pmod, wl);
215 k_sys = s/(s^2 + wl^2);
216 [mag, ~] = bode(k_sys, wl);
217 ki = [(1000/Gp)-kp]/mag
218
219 K_pr = kp + (ki*s)/(s^2 + 10^-6*s + wl^2)
      ;
220
221 disp('Poles and zeros of the PR
      Controller');
222 damp(pole(K_pr))
223 damp(zero(K_pr))
224
225 disp('Transfer Function of the PR

```

```

        Controller ');
226 zpk(K_pr)
227
228 %% Outer Controller
229 % Ko (1st design)
230 Ko1 = 0.15*K_pr;
231
232 disp('Poles and zeros of the Outer
        Controller ');
233 pole(Ko1)
234 zero(Ko1)
235
236 disp('Transfer Function of the Outer
        Controller ');
237 zpk(Ko1)
238
239 w = logspace(2,3,200000);
240
241 figure;
242 bode(Ko1,w);
243 grid on;
244 opts=bodeoptions;
245 opts.InputLabels.FontSize=12;
246 opts.OutputLabels.FontSize=12;
247 opts.XLabel.FontSize=14;

```

```

248 opts.YLabel.FontSize=14;
249 opts.Title.FontSize=20;
250 h_line = findobj(gcf, 'type', 'line');
251 set(h_line, 'LineWidth',2);
252 h_axes = findobj(gcf, 'type', 'axes');
253 set(h_axes, 'LineWidth',1, 'FontSize',14, '
    GridAlpha',0.25);
254 title('Outer Controller');
255
256 zeta = [ki/kp] * [1/(2*w1)]
257 %% Modified Controller
258
259 Kmod1 = Ki - Ko1;
260
261 disp('Poles and zeros of the Modified
    Controller');
262 pole(Kmod1)
263 zero(Kmod1)
264
265 disp('Transfer Function of the Modified
    Controller');
266 zpk(Kmod1)
267
268 %% Closed Loop Maps
269

```

```

270 % Build the Closed Loop System using "
        connect". There are 3 systems; Ko, Ki
271 % and P
272
273 % Label the block I/Os
274 Ko1.u = 'e';      Ko1.y = 'uo';
275 Ki.u = 'y';      Ki.y = 'ui';
276 P.u = 'up';      P.y = 'y';
277 % Specify summing junctions
278 Sum1 = sumblk('e = r - y');
279 Sum2 = sumblk('u = uo + ui');
280 Sum3 = sumblk('up = di + u');
281 % Add analysis points
282 up = AnalysisPoint('up');
283 u = AnalysisPoint('u');
284 e = AnalysisPoint('e');
285 % Connect the blocks together
286 T0 = connect(P,Ko1,Ki,Sum1,Sum2,Sum3,{ 'r '
        , 'di '},{ 'e ', 'up ', 'y ', 'u '},{ 'up ', 'u ', 'e
        '});
287 % getLoopTransfer(T,Locations , sign ,
        openings)
288 Lo1 = getLoopTransfer(T0, 'e ', -1);
289 Lu1 = getLoopTransfer(T0, 'up ');
290

```



```

291 So1 = getIOTransfer(T0, 'r', 'e');
292 Su1 = getIOTransfer(T0, 'di', 'up');
293 To1 = getIOTransfer(T0, 'r', 'y');
294 Tu1 = getIOTransfer(T0, 'di', 'u');
295 KS1 = getIOTransfer(T0, 'r', 'u');
296 SP1 = getIOTransfer(T0, 'di', 'y');
297
298 disp('Closed Loop Stability');
299 isstable(To1)
300
301 figure;
302 margin(Lo1);
303 grid on;
304
305 disp('Peak So in dB');
306 mag2db(norm(So1, Inf))
307 disp('Peak Su in dB');
308 mag2db(norm(Su1, Inf))
309 disp('Peak To in dB');
310 mag2db(norm(To1, Inf))
311 disp('Peak Tu in dB');
312 mag2db(norm(Tu1, Inf))
313
314 figure;
315 margin(Lu1);

```

```

316 grid on;
317
318 w1 = 59.3 * (2*pi);
319 disp('The open loop magnitude at 59.3 Hz
        is ');
320 [Lo_w1, ~] = bode(Lo1, w1)
321
322
323 save('Check_Design_Methodology', 'Ko1', '
        Lo1', 'K_pr');
324
325 Ksum = Ki - Ko1;
326
327 disp('———');
328 disp('Real Zero of Ksum');
329 Ksum_zeros = zero(Ksum);
330 Indx = find(imag(Ksum_zeros)==0);
331 Ksum_zeros(Indx)
332 disp('———');
333 %% Check that PM3u is associated with Tu
        and Su
334
335 Mar = allmargin(Lu1);
336 disp('Third phase margin of Lu');
337 PMu3 = Mar.PhaseMargin(3)

```

```

338 disp('mag2db(1/abs(2*sin(PMu3/2)))');
339 mag2db(1/abs(2*sin(PMu3/2)))
340
341 %% Frequency at which we want to add lag
342
343 Mar = allmargin(Lu1);
344 w_lag = Mar.PMFrequency(3);
345
346 %% Add lag at the w_lag frequency
347
348 phi_m = 25;
349 z_p = [1 - sin(deg2rad(phi_m))]/[1 + sin(
        deg2rad(phi_m))];
350 p = w_lag/sqrt(z_p);
351 z = z_p*p;
352
353 C_lag = sqrt(z/p) * [(s+p) / (s+z)];
354
355 figure;
356 bode(C_lag, wvec);
357 grid on;
358 title('Lag network designed');
359 h_line = findobj(gcf, 'type', 'line');
360 set(h_line, 'LineWidth', 2);
361

```

```

362 %% 25 deg Lag in series with Kmod
363
364 Ko2 = Ko1 * C_lag;
365 Ki2 = Ki * C_lag;
366
367 % Build the Closed Loop System using "
           connect". There are 3 systems; Ko, Ki
368 % and P
369
370 % Label the block I/Os
371 Ko2.u = 'e';      Ko2.y = 'uo';
372 Ki2.u = 'y';      Ki2.y = 'ui';
373 P.u = 'up';      P.y = 'y';
374 % Specify summing junctions
375 Sum1 = sumblk('e = r - y');
376 Sum2 = sumblk('u = uo + ui');
377 Sum3 = sumblk('up = di + u');
378 % Add analysis points
379 up = AnalysisPoint('up');
380 u = AnalysisPoint('u');
381 e = AnalysisPoint('e');
382 % Connect the blocks together
383 T0 = connect(P, Ko2, Ki2, Sum1, Sum2, Sum3, { 'r
           ', 'di' }, { 'e', 'up', 'y', 'u' }, { 'up', 'u', '
           e' });

```

```

384 % getLoopTransfer (T, Locations , sign ,
        openings)
385 Lo2 = getLoopTransfer (T0, 'e', -1);
386 Lu2 = getLoopTransfer (T0, 'up');
387
388 So2 = getIOTransfer (T0, 'r', 'e');
389 Su2 = getIOTransfer (T0, 'di', 'up');
390 To2 = getIOTransfer (T0, 'r', 'y');
391 Tu2 = getIOTransfer (T0, 'di', 'u');
392 KS2 = getIOTransfer (T0, 'r', 'u');
393 SP2 = getIOTransfer (T0, 'di', 'y');
394
395 Mar = allmargin (Lo2);
396 BW_Lo = Mar.PMFrequency(1);
397
398 disp('---25deg Lag design---');
399 disp('Closed Loop Stability');
400 isstable (To2)
401 disp('Peak Su (in dB)');
402 mag2db(norm(Su2, Inf))
403 disp('Peak So (in dB)');
404 mag2db(norm(So2, Inf))
405 disp('Peak To (in dB)');
406 mag2db(norm(To2, Inf))
407 disp('Peak Tu (in dB)');

```

```

408 mag2db(norm(Tu2, Inf))
409 disp('Bandwidth of Lo');
410 disp(BW_Lo);
411
412 disp('Stability of all the closed loop
      maps for this design (25 deg lag)');
413 if( isstable(To2) & isstable(So2) &
      isstable(Su2) & isstable(Tu2) &
      isstable(KS2) & isstable(SP2))
414     disp('All stable');
415 else
416     disp('Some of the maps are not stable
      ');
417 end
418
419
420 figure;
421 margin(Lo2);
422 grid on;
423
424 figure;
425 margin(Lu2);
426 grid on;
427
428 Pmod = feedback(P, Ki2, +1);

```

```

429
430 [PMOD,~] = bode(Pmod,100)
431 [LO2,~] = bode(Lo2,100)
432
433 figure;
434 bode(Lo2,wvec,'b');
435 hold on;
436 bode(Pmod,wvec,'r—');
437 opts=bodeoptions;
438 opts.InputLabels.FontSize=12;
439 opts.OutputLabels.FontSize=12;
440 opts.XLabel.FontSize=14;
441 opts.YLabel.FontSize=14;
442 opts.Title.FontSize=20;
443 h_line = findobj(gcf,'type','line');
444 set(h_line,'LineWidth',2);
445 h_axes = findobj(gcf,'type','axes');
446 set(h_axes,'LineWidth',1,'FontSize',14,'
    GridAlpha',0.25);
447 legend('Lo','Pmod');
448
449 figure;
450 bode(Lo2,wvec,'b');
451 hold on;
452 bode(Pmod*(LO2/PMOD),wvec,'r—');

```

```

453 opts=bodeoptions;
454 opts.InputLabels.FontSize=12;
455 opts.OutputLabels.FontSize=12;
456 opts.XLabel.FontSize=14;
457 opts.YLabel.FontSize=14;
458 opts.Title.FontSize=20;
459 h_line = findobj(gcf, 'type', 'line');
460 set(h_line, 'LineWidth',2);
461 h_axes = findobj(gcf, 'type', 'axes');
462 set(h_axes, 'LineWidth',1, 'FontSize',14, '
      GridAlpha',0.25);
463 legend('Lo', 'Pmod*(15.5/2.5k)');
464
465 %% 20 deg of lag
466
467 phi_m = 20;
468 z_p = [1 - sin(deg2rad(phi_m))]/[1 + sin(
      deg2rad(phi_m))];
469 p = w_lag/sqrt(z_p);
470 z = z_p*p;
471
472 C_lag = sqrt(z/p) * [(s+p) / (s+z)];
473
474 Ko3 = Ko1 * C_lag;
475 Ki3 = Ki * C_lag;

```



```

476
477 % Build the Closed Loop System using "
           connect". There are 3 systems; Ko, Ki
478 % and P
479
480 % Label the block I/Os
481 Ko3.u = 'e';      Ko3.y = 'uo';
482 Ki3.u = 'y';      Ki3.y = 'ui';
483 P.u = 'up';      P.y = 'y';
484 % Specify summing junctions
485 Sum1 = sumblk('e = r - y');
486 Sum2 = sumblk('u = uo + ui');
487 Sum3 = sumblk('up = di + u');
488 % Add analysis points
489 up = AnalysisPoint('up');
490 u = AnalysisPoint('u');
491 e = AnalysisPoint('e');
492 % Connect the blocks together
493 T0 = connect(P,Ko3,Ki3,Sum1,Sum2,Sum3,{ 'r
           ', 'di' }, { 'e', 'up', 'y', 'u' }, { 'up', 'u', '
           e' });
494 % getLoopTransfer(T, Locations, sign,
           openings)
495 Lo3 = getLoopTransfer(T0, 'e', -1);
496 Lu3 = getLoopTransfer(T0, 'up');

```

```

497
498 So3 = getIOTransfer(T0, 'r', 'e');
499 Su3 = getIOTransfer(T0, 'di', 'up');
500 To3 = getIOTransfer(T0, 'r', 'y');
501 Tu3 = getIOTransfer(T0, 'di', 'u');
502 KS3 = getIOTransfer(T0, 'r', 'u');
503 SP3 = getIOTransfer(T0, 'di', 'y');
504
505 Mar = allmargin(Lo3);
506 BWLo = Mar.PMFrequency(1);
507
508 disp('---20deg Lag design---');
509 disp('Closed Loop Stability');
510 isstable(To3)
511 disp('Peak Su (in dB)');
512 mag2db(norm(Su3, Inf))
513 disp('Peak So (in dB)');
514 mag2db(norm(So3, Inf))
515 disp('Peak To (in dB)');
516 mag2db(norm(To3, Inf))
517 disp('Peak Tu (in dB)');
518 mag2db(norm(Tu3, Inf))
519 disp('Bandwidth of Lo');
520 disp(BWLo);
521

```

```

522 disp('Stability of all the closed loop
        maps for this design (20 deg lag)');
523 if( isstable(To3) & isstable(So3) &
        isstable(Su3) & isstable(Tu3) &
        isstable(KS3) & isstable(SP3))
524     disp('All stable');
525 else
526     disp('Some of the maps are not stable
            ');
527 end
528
529
530 %% 15 deg of lag
531
532 phi_m = 15;
533 z_p = [1 - sin(deg2rad(phi_m))]/[1 + sin(
        deg2rad(phi_m))];
534 p = w_lag/sqrt(z_p);
535 z = z_p*p;
536
537 C_lag = sqrt(z/p) * [(s+p) / (s+z)];
538
539 Ko4 = Ko1 * C_lag;
540 Ki4 = Ki * C_lag;
541

```

```

542 % Build the Closed Loop System using "
           connect". There are 3 systems; Ko, Ki
543 % and P
544
545 % Label the block I/Os
546 Ko4.u = 'e';      Ko4.y = 'uo';
547 Ki4.u = 'y';      Ki4.y = 'ui';
548 P.u = 'up';       P.y = 'y';
549 % Specify summing junctions
550 Sum1 = sumblk('e = r - y');
551 Sum2 = sumblk('u = uo + ui');
552 Sum3 = sumblk('up = di + u');
553 % Add analysis points
554 up = AnalysisPoint('up');
555 u = AnalysisPoint('u');
556 e = AnalysisPoint('e');
557 % Connect the blocks together
558 T0 = connect(P,Ko4,Ki4,Sum1,Sum2,Sum3,{ 'r
           ', 'di' }, { 'e', 'up', 'y', 'u' }, { 'up', 'u', '
           e' });
559 % getLoopTransfer(T, Locations, sign,
           openings)
560 Lo4 = getLoopTransfer(T0, 'e', -1);
561 Lu4 = getLoopTransfer(T0, 'up');
562

```

```

563 So4 = getIOTransfer(T0, 'r', 'e');
564 Su4 = getIOTransfer(T0, 'di', 'up');
565 To4 = getIOTransfer(T0, 'r', 'y');
566 Tu4 = getIOTransfer(T0, 'di', 'u');
567 KS4 = getIOTransfer(T0, 'r', 'u');
568 SP4 = getIOTransfer(T0, 'di', 'y');
569
570 Mar = allmargin(Lo4);
571 BWLo = Mar.PMFrequency(1);
572
573 disp('——15deg Lag design——');
574 disp('Closed Loop Stability');
575 isstable(To4)
576 disp('Peak Su (in dB)');
577 mag2db(norm(Su4, Inf))
578 disp('Peak So (in dB)');
579 mag2db(norm(So4, Inf))
580 disp('Peak To (in dB)');
581 mag2db(norm(To4, Inf))
582 disp('Peak Tu (in dB)');
583 mag2db(norm(Tu4, Inf))
584 disp('Bandwidth of Lo');
585 disp(BWLo);
586
587 disp('Stability of all the closed loop

```

```

        maps for this design (15 deg lag)');
588 if( isstable(To4) & isstable(So4) &
        isstable(Su4) & isstable(Tu4) &
        isstable(KS4) & isstable(SP4))
589     disp('All stable');
590 else
591     disp('Some of the maps are not stable
        ');
592 end
593
594 %% Family of Plots
595
596 Line1 = 10^7/(s+10^7);
597 Line2 = 2*Line1;
598
599 [mag,ph] = bode(Line2,wvec);
600 mag = squeeze(mag(1,1,:));
601 mag = transpose(mag);
602 mag = mag2db(mag);
603 %ph = ph + 360;
604 ph = ph;
605 ph = squeeze(ph(1,1,:));
606 ph = transpose(ph);
607
608 figure;

```

```

609 Prefig(15);
610 subplot(2,1,1);
611 h3 = semilogx(wvec,mag,'b'); grid on;
        ylabel('Magnitude (dB)'); grid on;
612 set(gca,'fontsize',15);set(gca,'
        GridLineStyle','-', 'linewidth',2);
613 hold on;
614 title('Open Loop');
615 subplot(2,1,2);
616 h4 = semilogx(wvec,ph,'b'); grid on;
        ylabel('Phase (deg)'); grid on;
617 xlabel('Frequency (rad/s)');
618 hold on;
619 set(gca,'fontsize',15);set(gca,'
        GridLineStyle','-', 'linewidth',2);
620 %h1 = findobj(gcf,'type','line');
621 set(h3,'linewidth',3);
622 set(h4,'linewidth',3);
623 % Draw a vertical line from the -180deg
        line to the phase
624 disp('Lo2');
625 Mar = allmargin(Lo2)
626 % x = Mar.PMFrequency(2);
627 % PM = Mar.PhaseMargin(2);
628 % semilogx([x x],[-540 -540+PM],'k',''

```

```

        linewidth ',2);
629 % for ii = [1,3];
630 % x = Mar.PMFrequency(ii);
631 % PM = Mar.PhaseMargin(ii);
632 % semilogx([x x],[-180 -180+PM], 'k', '
        linewidth ',2);
633 % hold on;
634 % end
635 % Prefig(15);
636 x = Mar.PMFrequency;
637 PM = Mar.PhaseMargin;
638 semilogx([x x],[-180 -180+PM], 'k', '
        linewidth ',2);
639
640 hold on;
641
642 [mag,ph] = bode(Lo3,wvec);
643 mag = squeeze(mag(1,1,:));
644 mag = transpose(mag);
645 mag = mag2db(mag);
646 ph = ph;
647 ph = squeeze(ph(1,1,:));
648 ph = transpose(ph);
649
650 Prefig(15);

```



```

651 subplot(2,1,1);
652 h7 = semilogx(wvec,mag,'r'); grid on;
        ylabel('Magnitude (dB)'); grid on;
653 set(gca,'fontsize',15);set(gca,'
        GridLineStyle','-', 'linewidth',2);
654 hold on;
655 title('Open Loop (L_o)');
656 subplot(2,1,2);
657 h8 = semilogx(wvec,ph,'r'); grid on;
        ylabel('Phase (deg)'); grid on;
658 xlabel('Frequency (rad/s)');
659 set(gca,'fontsize',15);set(gca,'
        GridLineStyle','-', 'linewidth',2);
660 %h1 = findobj(gcf,'type','line');
661 set(h7,'linewidth',3);
662 set(h8,'linewidth',3);
663 % Draw a vertical line from the -180deg
        line to the phase
664 disp('Lo3');
665 Mar = allmargin(Lo3)
666 % x = Mar.PMFrequency(1);
667 % PM = Mar.PhaseMargin(1);
668 % semilogx([x x],[180 180+PM], 'k', '
        linewidth',2);
669 % for ii = 2:length(Mar.PMFrequency)

```

```

670 % x = Mar.PMFrequency(ii);
671 % PM = Mar.PhaseMargin(ii);
672 % semilogx([x x],[-180 -180+PM], 'k', '
        linewidth',2);
673 % hold on;
674 % end
675 % Prefig(15);
676 x = Mar.PMFrequency;
677 PM = Mar.PhaseMargin;
678 semilogx([x x],[-180 -180+PM], 'k', '
        linewidth',2);
679
680 hold on;
681
682 [mag,ph] = bode(Lo4,wvec);
683 mag = squeeze(mag(1,1,:));
684 mag = transpose(mag);
685 mag = mag2db(mag);
686 ph = ph;
687 ph = squeeze(ph(1,1,:));
688 ph = transpose(ph);
689
690 Prefig(15);
691 subplot(2,1,1);
692 h11 = semilogx(wvec,mag,'y'); grid on;

```

```

        ylabel('Magnitude (dB)'); grid on;
693 set(gca,'fontsize',15);set(gca,'
        GridLineStyle','-', 'linewidth',2);
694 hold on;
695 semilogx([10^2 10^5],[0 0], 'k', 'linewidth
        ',2);
696 hold on;
697 title('Open Loop (Lo)');
698 subplot(2,1,2);
699 h12 = semilogx(wvec,ph, 'y'); grid on;
        ylabel('Phase (deg)'); grid on;
700 xlabel('Frequency (rad/s)');
701 set(gca,'fontsize',15);set(gca,'
        GridLineStyle','-', 'linewidth',2);
702 %h1 = findobj(gcf,'type','line');
703 set(h11,'linewidth',3);
704 set(h12,'linewidth',3);
705 % Draw a vertical line from the -180deg
        line to the phase
706 disp('Lu3');
707 disp('Lu3');
708 Mar = allmargin(Lo4)
709 % x = Mar.PMFrequency(1);
710 % PM = Mar.PhaseMargin(1);
711 % semilogx([x x],[180 180+PM], 'k', '

```

```

        linewidth ',2);
712 % for ii = 2:length(Mar.PMFrequency)
713 % x = Mar.PMFrequency(ii);
714 % PM = Mar.PhaseMargin(ii);
715 % semilogx([x x],[-180 -180+PM], 'k', '
        linewidth ',2);
716 % hold on;
717 % end
718 % hold on;
719 x = Mar.PMFrequency;
720 PM = Mar.PhaseMargin;
721 semilogx([x x],[-180 -180+PM], 'k', '
        linewidth ',2);
722
723 semilogx([1e2 1e5],[180 180], 'k', '
        linewidth ',2);
724 hold on;
725 semilogx([1e2 1e5],[-180 -180], 'k', '
        linewidth ',2);
726 hold on;
727 semilogx([1e2 1e5],[-540 -540], 'k', '
        linewidth ',2);
728 Prefig(15);
729
730 legend([h3 h7 h11], 'Lo1', 'Lo2', 'Lo3');

```

```

731
732 figure ;
733 bodemag(So2 , wvec) ;
734 hold on ;
735 bodemag(So3 , wvec) ;
736 hold on ;
737 bodemag(So4 , wvec) ;
738 hold on ;
739 bodemag(Line1 , 'k—' , wvec) ;
740 hold on ;
741 bodemag(Line2 , 'k—' , wvec) ;
742 grid on ;
743 opts=bodeoptions ;
744 opts.InputLabels.FontSize=12;
745 opts.OutputLabels.FontSize=12;
746 opts.XLabel.FontSize=14;
747 opts.YLabel.FontSize=14;
748 opts.Title.FontSize=20;
749 h_line = findobj(gcf , 'type' , 'line') ;
750 set(h_line , 'LineWidth' , 2) ;
751 h_axes = findobj(gcf , 'type' , 'axes') ;
752 set(h_axes , 'LineWidth' , 1 , 'FontSize' , 14 , '
    GridAlpha' , 0.25) ;
753 title('So') ;
754 [hL , hObj]=legend('So1' , 'So2' , 'So3') ; %

```

```

        return the handles array
755 set(hL, 'FontSize',14);
756 hTL=findobj(hObj, 'type', 'line'); % get
        the lines , not text
757 set(hTL, 'LineWidth',2) % set linewidth
758 hTL=findobj(hObj, 'type', 'Text'); % get
        the text
759 set(hTL, 'FontSize',14) % set fontsize
760
761 figure;
762 bodemag(Su2, wvec);
763 hold on;
764 bodemag(Su3, wvec);
765 hold on;
766 bodemag(Su4, wvec);
767 hold on;
768 bodemag(Line1, 'k—', wvec);
769 hold on;
770 bodemag(Line2, 'k—', wvec);
771 grid on;
772 opts=bodeoptions;
773 opts.InputLabels.FontSize=12;
774 opts.OutputLabels.FontSize=12;
775 opts.XLabel.FontSize=14;
776 opts.YLabel.FontSize=14;

```

```

777 opts.Title.FontSize=20;
778 h_line = findobj(gcf, 'type', 'line');
779 set(h_line, 'LineWidth',2);
780 h_axes = findobj(gcf, 'type', 'axes');
781 set(h_axes, 'LineWidth',1, 'FontSize',14, '
    GridAlpha',0.25);
782 title('Su');
783 [hL,hObj]=legend('Su1','Su2','Su3'); %
    return the handles array
784 set(hL, 'FontSize',14);
785 hTL=findobj(hObj, 'type', 'line'); % get
    the lines, not text
786 set(hTL, 'LineWidth',2) % set linewidth
787 hTL=findobj(hObj, 'type', 'Text'); % get
    the text
788 set(hTL, 'FontSize',14) % set fontsize
789
790 figure;
791 bodemag(To2,wvec);
792 hold on;
793 bodemag(To3,wvec);
794 hold on;
795 bodemag(To4,wvec);
796 hold on;
797 bodemag(Line1, 'k—',wvec);

```

```

798 hold on;
799 bodemag(Line2, 'k—', wvec);
800 grid on;
801 opts=bodeoptions;
802 opts.InputLabels.FontSize=12;
803 opts.OutputLabels.FontSize=12;
804 opts.XLabel.FontSize=14;
805 opts.YLabel.FontSize=14;
806 opts.Title.FontSize=20;
807 h_line = findobj(gcf, 'type', 'line');
808 set(h_line, 'LineWidth', 2);
809 h_axes = findobj(gcf, 'type', 'axes');
810 set(h_axes, 'LineWidth', 1, 'FontSize', 14, '
      GridAlpha', 0.25);
811 title('To');
812 [hL, hObj]=legend('To1', 'To2', 'To3'); %
      return the handles array
813 set(hL, 'FontSize', 14);
814 hTL=findobj(hObj, 'type', 'line'); % get
      the lines, not text
815 set(hTL, 'LineWidth', 2) % set linewidth
816 hTL=findobj(hObj, 'type', 'Text'); % get
      the text
817 set(hTL, 'FontSize', 14) % set fontsize
818

```



```

819 figure ;
820 bodemag(Tu2, wvec) ;
821 hold on ;
822 bodemag(Tu3, wvec) ;
823 hold on ;
824 bodemag(Tu4, wvec) ;
825 hold on ;
826 bodemag(Line1 , 'k—' , wvec) ;
827 hold on ;
828 bodemag(Line2 , 'k—' , wvec) ;
829 grid on ;
830 opts=bodeoptions ;
831 opts.InputLabels.FontSize=12;
832 opts.OutputLabels.FontSize=12;
833 opts.XLabel.FontSize=14;
834 opts.YLabel.FontSize=14;
835 opts.Title.FontSize=20;
836 h_line = findobj(gcf , 'type' , 'line') ;
837 set(h_line , 'LineWidth' , 2) ;
838 h_axes = findobj(gcf , 'type' , 'axes') ;
839 set(h_axes , 'LineWidth' , 1 , 'FontSize' , 14 , '
      GridAlpha' , 0.25) ;
840 title('Tu') ;
841 [hL , hObj]=legend('Tu1' , 'Tu2' , 'Tu3') ; %
      return the handles array

```

```

842 set(hL, 'FontSize',14);
843 hTL=findobj(hObj, 'type', 'line'); % get
      the lines , not text
844 set(hTL, 'LineWidth',2) % set linewidth
845 hTL=findobj(hObj, 'type', 'Text'); % get
      the text
846 set(hTL, 'FontSize',14) % set fontsize
847
848 disp('Check Tu');
849 mag2db(norm(Tu2, Inf))
850 mag2db(norm(Tu3, Inf))
851 mag2db(norm(Tu4, Inf))
852
853 %% Lu
854
855 [mag, ph] = bode(Lu2, wvec);
856 mag = squeeze(mag(1,1,:));
857 mag = transpose(mag);
858 mag = mag2db(mag);
859 %ph = ph + 360;
860 ph = ph;
861 ph = squeeze(ph(1,1,:));
862 ph = transpose(ph);
863
864 figure;

```

```

865 Prefig(15);
866 subplot(2,1,1);
867 h3 = semilogx(wvec,mag,'b'); grid on;
      ylabel('Magnitude (dB)'); grid on;
868 set(gca,'fontsize',15);set(gca,'
      GridLineStyle','-', 'linewidth',2);
869 hold on;
870 title('Open Loop');
871 subplot(2,1,2);
872 h4 = semilogx(wvec,ph,'b'); grid on;
      ylabel('Phase (deg)'); grid on;
873 xlabel('Frequency (rad/s)');
874 hold on;
875 set(gca,'fontsize',15);set(gca,'
      GridLineStyle','-', 'linewidth',2);
876 %h1 = findobj(gcf,'type','line');
877 set(h3,'linewidth',3);
878 set(h4,'linewidth',3);
879 % Draw a vertical line from the -180deg
      line to the phase
880 disp('Lu1');
881 Mar = allmargin(Lu2)
882 x = Mar.PMFrequency(2);
883 PM = Mar.PhaseMargin(2);
884 semilogx([x x],[-540 -540+PM], 'k', '

```

```

        linewidth',2);
885 for ii = [1,3];
886 x = Mar.PMFrequency(ii);
887 PM = Mar.PhaseMargin(ii);
888 semilogx([x x],[-180 -180+PM], 'k', '
        linewidth',2);
889 hold on;
890 end
891 Prefig(15);
892
893 hold on;
894
895 [mag,ph] = bode(Lu3,wvec);
896 mag = squeeze(mag(1,1,:));
897 mag = transpose(mag);
898 mag = mag2db(mag);
899 ph = ph;
900 ph = squeeze(ph(1,1,:));
901 ph = transpose(ph);
902
903 Prefig(15);
904 subplot(2,1,1);
905 h7 = semilogx(wvec,mag,'r'); grid on;
        ylabel('Magnitude (dB)'); grid on;
906 set(gca,'fontsize',15);set(gca,'

```

```

        GridLineStyle', '- ', 'linewidth', 2);
907 hold on;
908 title('Open Loop (L_u)');
909 subplot(2,1,2);
910 h8 = semilogx(wvec, ph, 'r'); grid on;
        ylabel('Phase (deg)'); grid on;
911 xlabel('Frequency (rad/s)');
912 set(gca, 'fontsize', 15); set(gca, '
        GridLineStyle', '- ', 'linewidth', 2);
913 %h1 = findobj(gcf, 'type', 'line');
914 set(h7, 'linewidth', 3);
915 set(h8, 'linewidth', 3);
916 % Draw a vertical line from the -180deg
        line to the phase
917 disp('Lu2');
918 Mar = allmargin(Lu3)
919 x = Mar.PMFrequency(1);
920 PM = Mar.PhaseMargin(1);
921 semilogx([x x], [180 180+PM], 'k', '
        linewidth', 2);
922 for ii = 2:length(Mar.PMFrequency)
923 x = Mar.PMFrequency(ii);
924 PM = Mar.PhaseMargin(ii);
925 semilogx([x x], [-180 -180+PM], 'k', '
        linewidth', 2);

```

```

926 hold on;
927 end
928 Prefig(15);
929
930 hold on;
931
932 [mag,ph] = bode(Lu4,wvec);
933 mag = squeeze(mag(1,1,:));
934 mag = transpose(mag);
935 mag = mag2db(mag);
936 ph = ph;
937 ph = squeeze(ph(1,1,:));
938 ph = transpose(ph);
939
940 Prefig(15);
941 subplot(2,1,1);
942 h11 = semilogx(wvec,mag,'y'); grid on;
          ylabel('Magnitude (dB)'); grid on;
943 set(gca,'fontsize',15);set(gca,'
          GridLineStyle','-', 'linewidth',2);
944 hold on;
945 semilogx([10^2 10^5],[0 0],'k','linewidth
          ',2);
946 hold on;
947 title('Open Loop (Lu)');

```

```

948 subplot(2,1,2);
949 h12 = semilogx(wvec,ph,'y'); grid on;
        ylabel('Phase (deg)'); grid on;
950 xlabel('Frequency (rad/s)');
951 set(gca,'fontsize',15);set(gca,'
        GridLineStyle','-', 'linewidth',2);
952 %h1 = findobj(gcf,'type','line');
953 set(h11,'linewidth',3);
954 set(h12,'linewidth',3);
955 % Draw a vertical line from the -180deg
        line to the phase
956 disp('Lu3');
957 disp('Lu3');
958 Mar = allmargin(Lu4)
959 x = Mar.PMFrequency(1);
960 PM = Mar.PhaseMargin(1);
961 semilogx([x x],[180 180+PM], 'k', '
        linewidth',2);
962 for ii = 2:length(Mar.PMFrequency)
963 x = Mar.PMFrequency(ii);
964 PM = Mar.PhaseMargin(ii);
965 semilogx([x x],[-180 -180+PM], 'k', '
        linewidth',2);
966 hold on;
967 end

```

```

968 hold on;
969 semilogx([1e2 1e5],[180 180], 'k', '
        linewidth',2);
970 hold on;
971 semilogx([1e2 1e5],[-180 -180], 'k', '
        linewidth',2);
972 hold on;
973 semilogx([1e2 1e5],[-540 -540], 'k', '
        linewidth',2);
974 Prefig(15);
975
976 legend([h3 h7 h11], 'Lu1', 'Lu2', 'Lu3');
977
978 %% Compare Designs 1 and 4
979
980 [mag,ph] = bode(Lo1,wvec);
981 mag = squeeze(mag(1,1,:));
982 mag = transpose(mag);
983 mag = mag2db(mag);
984 %ph = ph + 360;
985 ph = ph;
986 ph = squeeze(ph(1,1,:));
987 ph = transpose(ph);
988
989 figure;

```



```

990 Prefig(15);
991 subplot(2,1,1);
992 h3 = semilogx(wvec,mag,'b'); grid on;
          ylabel('Magnitude (dB)'); grid on;
993 set(gca,'fontsize',15);set(gca,'
          GridLineStyle','-', 'linewidth',2);
994 hold on;
995 title('Open Loop');
996 subplot(2,1,2);
997 h4 = semilogx(wvec,ph,'b'); grid on;
          ylabel('Phase (deg)'); grid on;
998 xlabel('Frequency (rad/s)');
999 hold on;
1000 set(gca,'fontsize',15);set(gca,'
          GridLineStyle','-', 'linewidth',2);
1001 %h1 = findobj(gcf,'type','line');
1002 set(h3,'linewidth',3);
1003 set(h4,'linewidth',3);
1004 % Draw a vertical line from the -180deg
          line to the phase
1005 disp('Lo2');
1006 Mar = allmargin(Lo2)
1007 x = Mar.PMFrequency;
1008 PM = Mar.PhaseMargin;
1009 semilogx([x x],[-180 -180+PM], 'k', '

```

```

        linewidth',2);
1010
1011 hold on;
1012
1013 [mag,ph] = bode(Lo4,wvec);
1014 mag = squeeze(mag(1,1,:));
1015 mag = transpose(mag);
1016 mag = mag2db(mag);
1017 ph = ph;
1018 ph = squeeze(ph(1,1,:));
1019 ph = transpose(ph);
1020
1021 Prefig(15);
1022 subplot(2,1,1);
1023 h11 = semilogx(wvec,mag,'r'); grid on;
        ylabel('Magnitude (dB)'); grid on;
1024 set(gca,'fontsize',15);set(gca,'
        GridLineStyle','-', 'linewidth',2);
1025 hold on;
1026 semilogx([10^2 10^5],[0 0],'k','linewidth
        ',2);
1027 hold on;
1028 title('Open Loop (Lo)');
1029 subplot(2,1,2);
1030 h12 = semilogx(wvec,ph,'r'); grid on;

```

```

        ylabel('Phase (deg)'); grid on;
1031 xlabel('Frequency (rad/s)');
1032 set(gca,'fontsize',15);set(gca,'
        GridLineStyle','-', 'linewidth',2);
1033 %h1 = findobj(gcf,'type','line');
1034 set(h11,'linewidth',3);
1035 set(h12,'linewidth',3);
1036 % Draw a vertical line from the -180deg
        line to the phase
1037 disp('Lo4');
1038 Mar = allmargin(Lo4)
1039 x = Mar.PMFrequency;
1040 PM = Mar.PhaseMargin;
1041 semilogx([x x],[-180 -180+PM], 'k', '
        linewidth',2);
1042
1043 semilogx([1e2 1e5],[180 180], 'k', '
        linewidth',2);
1044 hold on;
1045 semilogx([1e2 1e5],[-180 -180], 'k', '
        linewidth',2);
1046 hold on;
1047 semilogx([1e2 1e5],[-540 -540], 'k', '
        linewidth',2);
1048 Prefig(15);

```

1049

```
1050 legend([h3 h11], 'Lo1', 'Lo4');

1 % m File: Imp_Plot_PAPER.m
2
3 clear all;
4 close all;
5 clc;
6 %% Load the Data;
7
8 load('Data.mat');
9
10 %% Minium of the Plots
11
12 disp('Design 1');
13 Min_kd_Tu1 = kd(find(PeakTu1 == min(
    PeakTu1)))
14
15 %% Plot
16
17 Tu_Ratio_PM3_1 = abs(1./[2* sin(deg2rad(
    Lu1_PM3(:)/2))]);
18
19 figure;
20 plot(kd, mag2db(Tu_Ratio_PM3_1), 'b', '
    LineWidth', 2);
```

```

21 hold on;
22 plot(kd,PeakTu1,'b—','LineWidth',2);
23 hold on;
24 plot(kd,PeakSu1,'b:','LineWidth',2);
25 hold on;
26 grid on;
27 plot(kd,6*ones(size(PeakSu1)),'k','
        LineWidth',2);
28 hold on;
29 semilogx([Min_kd_Tu1 Min_kd_Tu1],[0 min(
        PeakTu1)],'k','linewidth',2);
30 [hL,hObj] = legend('PMc3 Ratio','|T-c|-\
        infty}','|S-c|-\infty');
31 h_axes = findobj(gcf,'type','axes');
32 set(h_axes,'LineWidth',2,'FontSize',18);
33 h_line = findobj(gcf,'type','line');
34 set(h_line,'LineWidth',2);
35 h_axes = findobj(gcf,'type','axes');
36 set(h_axes,'LineWidth',2,'FontSize',20);
37 % set(hL,'FontSize',20);
38 hTL=findobj(hObj,'type','line'); % get
        the lines , not text
39 set(hTL,'LineWidth',2) % set linewidth
40 hTL=findobj(hObj,'type','Text'); % get
        the text (legend)

```

```

41 set(hTL, 'FontSize',20) % set fontsize
42 xlabel('k_i', 'FontSize',20);
43 ylabel(' |1/[2 sin(PM_{c-3}/2)]| and |S_c|
         -{\infty}, |T_c|_{-\infty} ');
44 title(' |1/[2 sin(PM_{c-3}/2)]| and |S_c|_{-
         {\infty}}, |T_c|_{-\infty} versus k_i ');

```