Design and Modeling of a High Current Switching Regulator

by

Danielle Coffing

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1997

Author .....
Department of Electrical Engineering and Computer Science
May 8, 1997

Certified by...
Martin F. Schlecht
Professor of Electrical Engineering
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Design and Modeling of a High Current Switching Regulator

by

Danielle Coffing

Submitted to the Department of Electrical Engineering and Computer Science
on May 8, 1997, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

A design method for switching regulators using Matlab and Simulink has been developed. The Matlab script presented calculates the necessary bandwidth of the loop given the power distribution network characteristics and the maximum acceptable output voltage variation. The compensation network is then calculated for given characteristics of the output filter components. The script also analyzes the stability over the range of possible load currents. The transfer functions describing the system are loaded into a regulator model in Simulink so that transient simulations can be performed. This design method has been correlated with Spice and four breadboards. The design cycle time can be decreased by using Matlab to gain intuition for how parameter and component variation affect the stability and transient performance of the regulator. This method presented is then used to determine the necessary characteristics of the error amplifier and comparators used in the MC33470 design. Finally, the OTA and comparator designs are presented.

Thesis Supervisor: Martin F. Schlecht
Title: Professor of Electrical Engineering

Acknowledgments

I gratefully thank the following people who have given me their support on this thesis:

Doug Domke, operations manager, for the opportunity to do this project.

Bob Vyne, design manager; and Al Zahedi, product manager; for the thesis description, administrative and technical support and oversight, and overall guidance and direction.

The Intel program managers; Bob Dotson, project leader; Jeff Morud, applications; and Todd Manes, production and test; for providing detailed oversight and help on specific design and operational requirements.

Also, never-ending technical support from Tom Somerville, Tom Petty, Troy Stockstad, and Richard Griffith in design; Monty Palmer and Theron Barnes in layout; and Jaime Vargas in test.

Finally, Professor Schlecht, for his thoughtful explanations and valuable advice.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Microprocessors are moving towards lower operating voltages and higher current requirements [11, 34]. The lower operating voltages are necessary to limit the power dissipation caused by both the increase in clock speeds to greater than 200 MHz and the increasing number of transistors in each successive generation IC [13]. At the same time, the tolerance level for supply voltage variation is decreasing. This has created a market for high performance power supplies using voltage regulator ICs that can provide exceptional load regulation and stability while minimizing cost. Because each new generation of microprocessor requires a lower operating voltage, voltage regulators are being designed with an onboard digital-to-analog converter (DAC) to increase flexibility and enhance long term usability [13, 15]. The regulator IC also contains a pulse width modulator (PWM), logic, and an error amplifier, but the power FETs, compensation network, and output inductor and capacitor are all external. Because the external components are chosen by the user, a list of possible component values should be provided as part of the total regulator design solution. These values will give the user guidelines as to what choices would provide the best transient response and stability to meet the performance requirements of the particular application while staying within a certain price range. Simulation problems can occur in both creating these guidelines and designing the basic system regulator.

This thesis will focus on two aspects of the regulator design problem. First, a design methodology will be developed using Matlab that avoids many of the typical

simulation problems. Secondly, the design parameters extracted using this methodology will be used to design the error amplifier and comparators needed for the regulator. This thesis work will be conducted at the Amplifier and Power group of Motorola's Logic and Analog Technologies Group in Tempe, Arizona. A standard switching regulator targeting the microprocessor market will be designed. This buck regulator will use a 5 V input voltage to provide an output voltage between 1.8 and 3.5 V, programmed by an integrated 5-bit DAC in 50 mV increments below 2.1 V and 100 mV increments above 2.1 V. The regulator will maintain the output voltage to within $\pm$ 5% of its nominal value during a transition between no load and maximum load of 14 A. This $\pm$5% window includes variations due to both the voltage reference and the load transient effects. The regulator IC has to be able to slew at 30 A/μs at the output pins which drive the external power devices.

Due to the large current demands, the regulator must be very efficient to reduce power dissipation. The regulator must have an efficiency rating of at least 80% at maximum rated load current and a minimum of 40% at low load conditions. To protect itself and the microprocessor it powers, the regulator must provide several protective functions. First, it needs to power down if the output voltage goes more than 15% higher than desired. Secondly, if the load current increases beyond a specified level, the regulator needs to provide a constant current that will not damage the IC. The goal of this design is to provide all of the above functionality and versatility and yet cost less than the currently favored discrete linear regulator.

1.1 Background

As microprocessor technologies and architectures become more advanced, the amount of supply voltage variation that they can tolerate becomes more restrictive. This results in a more complex voltage regulator design, including, but not limited to, the use of nested control loops, more complex error amplifiers, and extra logic in an attempt to improve transient performance. The additional features required can make simulating the voltage regulator system much more difficult. At the same time,

increasing complexity makes top level system simulations more important since there can now be more opportunities for errors to occur.

Simplification of the top level schematic can help Spice[1] simulation convergence problems as well as dramatically improve run time performance. These simplifications include using macromodels to represent some subcircuits. For example, a comparator can be modeled as a differential input high gain block with the output limited between ground and the power supply.

For this thesis work, two simulation packages, PSpice and MCSpice, were available. MCSpice is Motorola's internal version of Spice. Neither package alone provides a complete design solution. Top level behavioral modeling and system simulation is adequately supported in PSpice, but is not fully implemented in the most recent version of MCSpice. For transistor level simulation, MCSpice uses an advanced, proprietary SSIM MOSFET model. However, PSpice still relies on an antiquated Level 3 MOS model. This model drastically overestimates both output impedance and moderate inversion transconductance. A more efficient and reliable design tool is necessary to increase system level understanding and to address transistor level design issues.

Along with the simulation problems mentioned above, the designer also must address the problem of deciding which control mode to use in the regulator design. Currently, three control modes are commonly used in the industry for switching regulators. However, these modes are not compatible with each other, utilizing different board layouts, compensation techniques and external components. As each successive microprocessor generation requires more stringent power supply controls, the use of voltage regulator modules will become prevalent and one control method will more than likely become the industry standard. Therefore, the competition needs to be surveyed so that the right control method can be chosen which will lead to the highest probability of project success.

[1]Spice stands for Simulation Program with Integrated Circuit Emphasis

1.2 Design Strategy

This project has three main goals. First, the switching regulator IC must be less expensive than other regulator ICs currently available. This dictates how large the die area can be and what sort of package can be used. The limit on die size advocates the use of the simplest possible solution to reduce the area needed. However, the regulator also needs to be well designed so that it can provide reasonable performance with a variety of external components. This will allow the customer to choose the most cost effective solution for their application. Secondly, the regulator must be completely compatible with similar products already commercially available so that each regulator will have comparable performance for the same external components. Finally, the regulator design must be completed quickly so that it can reach the market in time to be designed into microprocessor power supplies.

The first objective is to understand the system level issues and use this understanding to develop a software algorithm to assist in the regulator design. This algorithm should be generic enough to allow it to be used with future regulator design programs. The first part of the software to assist with the switching regulator design is a Matlab program, or "script". This script will allow the system level design of a switching regulator to be interactive. It should also provide almost instantaneous feedback on system performance when any key parameter, either external or internal to the system, is modified. For example, the script should be able to calculate the necessary compensation values to meet the loop bandwidth specification and desired system phase margin. Also, it should be able to model the effect of the error amplifier characteristics on the loop performance and find the total cost of the external components used. The second part of the software used in the design process will be Simulink. Simulink allows transient responses to be simulated using Matlab transfer functions.

The Matlab algorithm developed will provide information on the effects of the subsystem parameters on the regulator performance. For example, changing the open loop gain of the error amplifier will affect the phase margin and transient response of

the regulator and the designer needs to know if this change will be significant. Once these effects are understood, the error amplifier and comparator specifications can be determined. The second objective of this thesis is to determine these specifications and use them to design the comparators and error amplifier. Because the project goals include designing a regulator controller IC that is completely compatible with competitors' parts and completing the design as quickly as possible, developing a Matlab design method is necessary.

1.3 Thesis Organization

This thesis is organized as follows. Chapter 2 will present an overview of linear and switching regulators and the three most commonly used control methods for switching regulators. Chapter 3 will describe Matlab and the advantages of using it to understand a regulator system in the frequency domain. A simple frequency domain model of the regulator will also be derived in this chapter. Chapter 4 will develop a more detailed regulator model that accounts for parasitic component values and second order effects. A Matlab script will be developed that allows the regulator design process to be interactive and gives the designer an intuitive feel for how parameter variations affect the stability of the regulator.

In Chapter 5, the Matlab design methodology will be verified using PSpice and three breadboards built with similar regulator ICs. Chapter 6 will use the Matlab script to understand the design issues involved in Motorola's high current switching regulator, the MC33470, and extract subsystem parameters. Chapter 7 will describe the error amplifier design process using the parameters provided by the Matlab script. Chapter 8 will describe the requirements of the maximum and minimum comparators and their design process. Finally, the conclusion will be presented in Chapter 9.

Chapter 2

Voltage Regulators

Voltage regulators can be used to provide microprocessors with a well-controlled power supply at the desired operating voltage from a pre-existing supply. Each successive microprocessor generation requires a lower operating voltage and higher supply current. The trend in power supply voltage is shown in Figure 2.1 [18]. The drop in the supply voltage has been driven by two requirements. First, each new microprocessor generation contains an ever-increasing number of transistors. The Pentium® Pro[1], for example, has 5.5 million transistors. To allow such a large number of transistors to be integrated onto a single integrated circuit, minimum device spacing and size has decreased. Therefore, lower supply voltages are needed to keep the electric field from exceeding the dielectric breakdown of these high integration technologies. Second, decreasing the supply voltage decreases the power dissipation in the IC for a given drain current and clock frequency. The dynamic power dissipation is given as

$$P_{dyn} = \frac{1}{2} V_{dd}^2 C_l f_{clk} \tag{2.1}$$

As clock frequencies continue to increase, higher supply currents are required to account for an increase in parasitic, oxide and junction capacitances associated with these very high density integrated circuits. Therefore, even though the supply voltage has decreased over time, the power dissipation has increased dramatically due to

[1]Pentium is a registered trademark of Intel Corp.

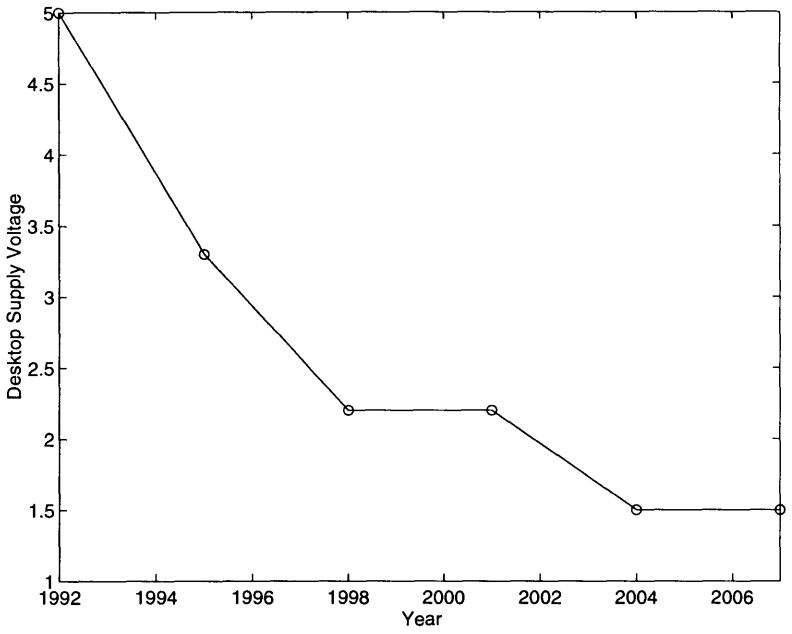


Figure 2-1: Microprocessor supply voltage trend.

the increasing clock speeds, supply current, and number of transistors. The power dissipation trend is shown in Figure 2.2 [18].

Because microprocessors have moved to lower operating voltages and higher supply currents, the demands placed on regulator efficiency and regulation tolerance have become more restrictive. In the past, regulator efficiency was not as critical because the total power dissipation was low enough that it did not cause excessive heating in the power supply unit. However, regulator efficiency can no longer be ignored as modern microprocessors approach power dissipation levels in excess of 25 W. Also, as microprocessor supply voltages decrease, the relative tolerance of the regulated voltage does not. For example, a 3.3 V, $\pm 5\%$ supply would require the regulator to maintain the nominal voltage within a 330 mV tolerance window. However, for a supply voltage of 1.8 V, this $\pm 5\%$ tolerance requirement has reduced the same window to just 180 mV [9]. Therefore, as the supply voltage decreases, the voltage regulator needs to be able to control the output voltage more accurately to ensure that the microprocessor performs properly and provides long term reliability.

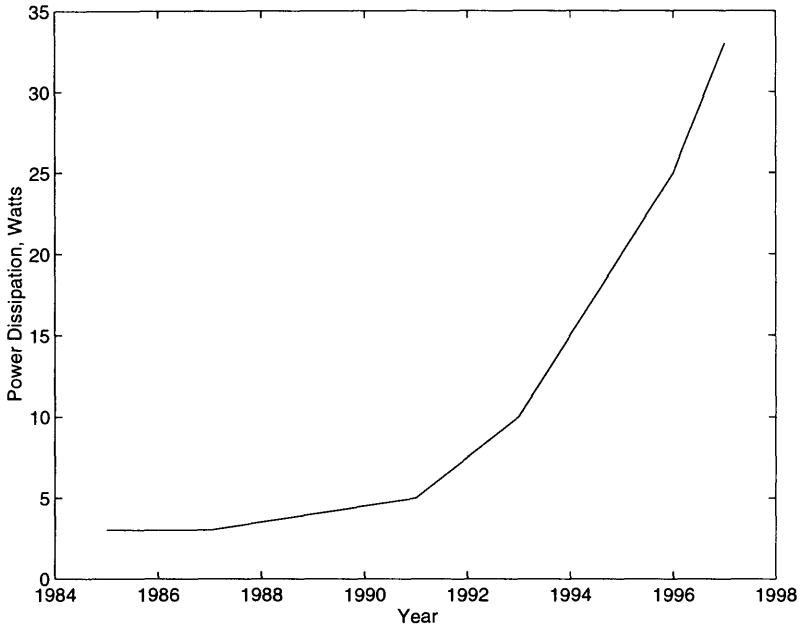


Figure 2-2: Microprocessor power dissipation trend.

2.1 Switching versus Linear Regulators

In the past, if power dissipation was secondary to cost and complexity, microprocessor power dissipation levels were low enough to allow discrete linear regulators to be used to convert from the 5 volt supply to the microprocessor supply voltage [26].

A simplified block diagram of a linear regulator is shown in Figure 2.3 [1]. The regulator uses an error amplifier to compare the output voltage to a reference voltage. The amplifier then generates a control signal proportional to the error between the desired and actual output voltages. This linear control signal drives a pass transistor at some point between saturation and cutoff so that the correct amount of supply current is supplied to the load to keep the output voltage at the correct value. An input filter can be used to prevent rapid changes in load current from affecting the 5 volt supply. The output filter is used to reduce the ripple in the output voltage due to large load current changes.

The principle advantage of using a linear regulator is that it can have fewer components than a switching regulator and can therefore be less expensive. However, for applications in which efficiency is a critical design goal, a linear regulator may not be

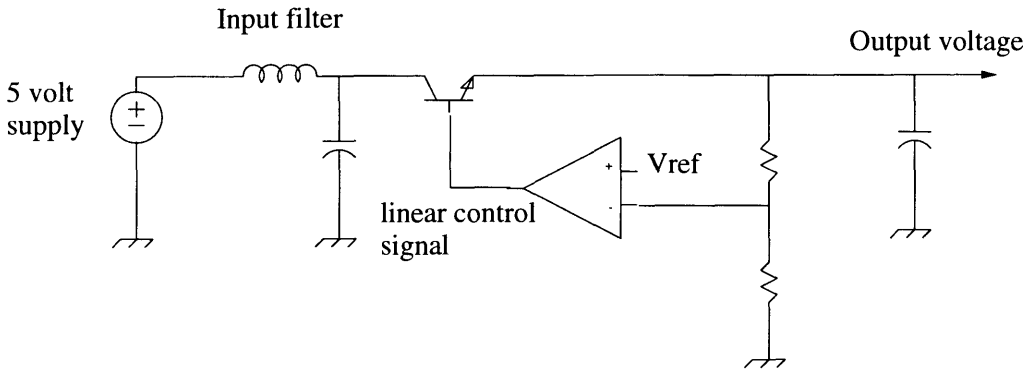


Figure 2-3: Block diagram of a linear regulator.

appropriate. The linear control signal generated by the error amplifier tends to keep the pass transistor biased such that a voltage drop always exist between its collector and emitter (or drain and source). This differential voltage can be as high as several volts. The power dissipation in the pass transistor, which represents as much as 95% of all the efficiency loss in a linear regulator, is given as the differential voltage times the load current. As microprocessor operating voltages decrease and supply current requirements increase, this power dissipation increases.

Switching regulators, in general, can achieve a much higher efficiency rating than a linear regulator. Historically, this has come at the expense of additional components and increased circuit complexity. However, modern, low cost process technologies have reduced both the cost of the regulator IC and the associated peripheral components, making it a very attractive alternative for power supply designers. A simplified block diagram of a switching regulator is shown in Figure 2.4. The output voltage is compared to a reference voltage and a digital signal is produced by a comparator to indicate whether the output voltage is above or below the reference voltage. If the output voltage is too low, the comparator enables the oscillator signal to drive the pass transistor. The oscillator signal is a square wave that cycles between ground and the input supply voltage. This ensures that the transistor is always either in saturation or cutoff. In saturation, the voltage from collector to emitter is very low, and therefore the power dissipation is low. In cutoff, the current through the transistor

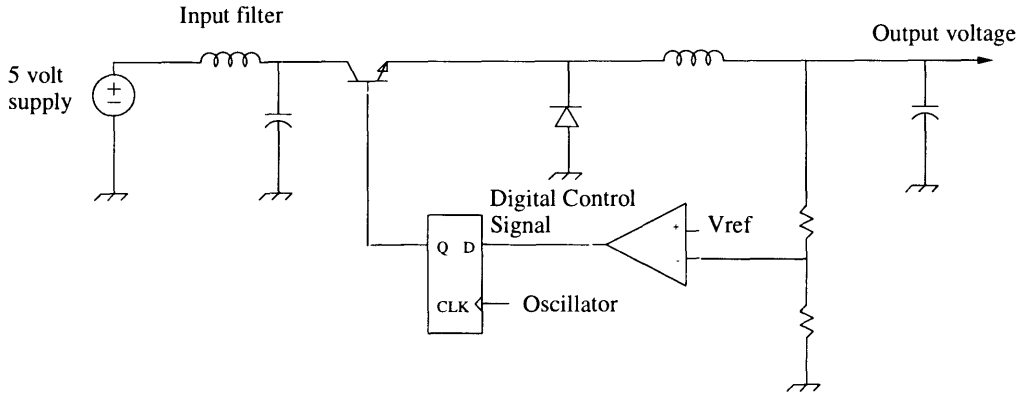


Figure 2-4: Switching regulator block diagram.

is very low, and again the power dissipation in the transistor is low. If the output voltage is too high, the comparator disables the oscillator signal and the transistor is kept in the cutoff region. Because no current is being supplied to the load, the load current being removed from the output filter capacitor will reduce the output voltage. Once it falls below the reference voltage, the comparator will again enable the oscillator drive to the pass transistor.

Three common topologies for switching regulators are buck, boost, and flyback [8]. Each of these topologies is capable of operating in both discontinuous and continuous modes. The discontinuous mode refers to the situation in which the current though the inductor goes discontinuous during part of the clock cycle. This changes the location of the poles and zeros in the output filter and therefore changes the system stability. In the continuous mode, the current in the inductor is never zero. Whether the regulator operates in discontinuous mode or continuous mode is determined by the external components, specifically whether a FET or diode is used between the coil and ground. Since the external components, in general, are chosen by the customer, an understanding of these two situations is required.

Vendor	Part Number	Control Method
Unitrode	UC3570	Voltage
Unitrode	UC3886	Average Current
Linfinity	LMX1600A	Proprietary (VRM Only)
MicroLinear	ML4900	Voltage
Raytheon	RC5042	Voltage/Current
Cherry	CS5150	Proprietary (Dual Feedback)
Maxim	MAX797	Peak Current
Elantec	EL7560/61	Peak Current
Linear Tech	LTC1553	Voltage

Table 2.1: Summary of switching regulator control method by company

2.2 Switching Regulator Control Methods

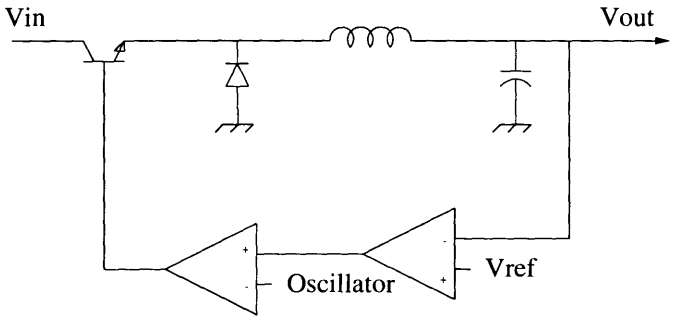
Several different switching regulator control methods exist that can be utilized to provide the level of performance needed. However, due to differences in compensation techniques, packaging and external components, the control methods are not compatible. Therefore, to be competitive in the microprocessor power supply market, proper selection of the control method most widely used by other vendors and preferred by customers becomes critical. The information available on high current switching regulator ICs from throughout the industry was examined and three regulator control methods were found to dominate. These methods are voltage mode control, average current mode control, and peak current mode control. Information on eight switching regulator control ICs designed for high end microprocessors, including the Intel Pentium® Pro, was collected and a summary of part numbers and the control mode used is provided in Table 2.1 [4, 6, 10, 16, 17, 19, 20].

This benchmark survey helps to illustrate the diversity of switching regulator controller architectures currently available. Without a clear industry standard, each control mode must be examined in detail. The next sections outline the benefits, and disadvantages, of each control mode in an effort to determine the most appropriate choice for this design program.

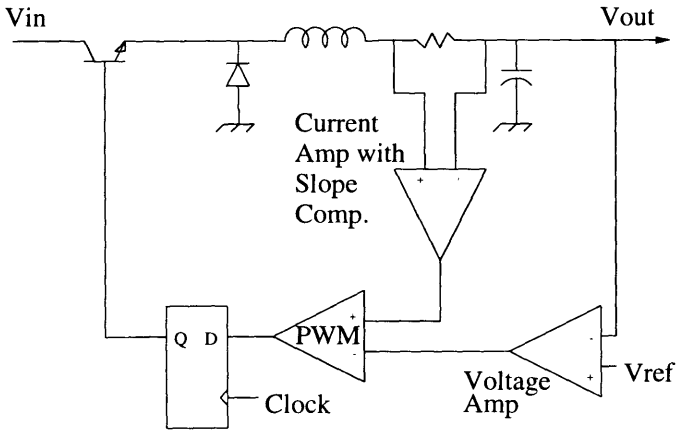
2.2.1 Voltage Mode Control

Voltage mode control was the first method developed for use in switching regulators [7]. A simplified schematic of a voltage mode controlled regulator is shown in Figure 2.5a [18]. The duty cycle of the control waveform is generated by comparing the error signal to a sawtooth waveform using the pulse width modulator (PWM) comparator. The error amplifier signal is determined solely by comparing the output voltage to the reference voltage. Because only one control signal is used to determine the duty cycle of the waveform applied to the power FET devices, voltage mode control is the simplest control method. This simplicity means that few external components are required, and the internal IC design consumes less die area, allowing this control method to be, in general, the least expensive solution. In addition, the stability analysis and simulation modeling can be performed more easily than for control methods using multiple control loops.

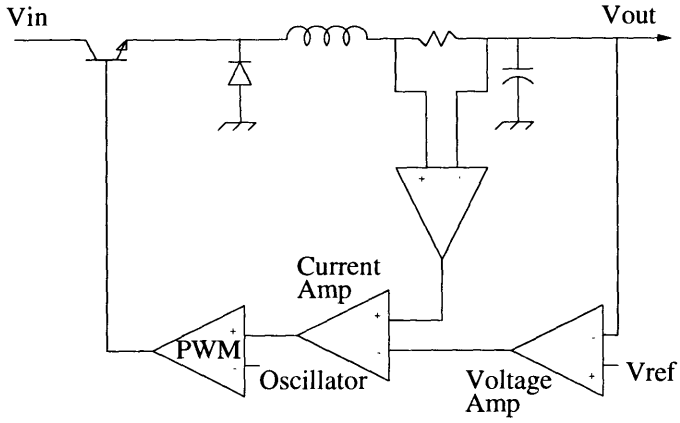
However, with this simplicity comes some disadvantages [18]. For voltage mode control, the two poles due to the output filter need to be compensated directly. This is accomplished by adding positive phase, or a "phase boost" at the double pole location. This compensation technique causes the loop bandwidth of the regulator to be limited if an acceptable phase margin is to be maintained. Likewise, the phase margin of the system can be very sensitive to variations in the components used with this compensation method. Any variation of the filter capacitor ESR, R_{DSon} of the power FETs, ESR of the inductor, or compensation component values can change the system phase margin by several degrees, resulting in a change in the transient performance of the regulator. Also, instability can be caused by adding an input filter if the impedance of the filter is chosen incorrectly.



a. Voltage Mode



b. Peak Current Mode



c. Average Current Mode

Figure 2-5: a) Voltage mode control. b) Peak current mode control. c) Average current mode control.

Unlike current mode control techniques, the current limiting functionality is not inherent in this control method and must be added separately. Finally, the gain due to the PWM is a function of the input voltage and oscillator peak to peak voltage. If the input voltage changes, the resulting loop bandwidth and gain will change, again effecting the transient performance of the regulator [17].

2.2.2 Peak Current Mode Control

A block diagram of peak current mode control is shown in Figure 2.5b [18]. This control method generates a control signal by comparing the inductor current to the output voltage. The slope of the inductor current is given by Eq. 2.2.

$$\Delta I = \frac{v_{in} - v_{out}}{L} \tag{2.2}$$

This inductor current has a sawtooth shape similar to that generated by the oscillator used in voltage mode control. However, unlike the constant slope of the oscillator waveform, the slope of the current waveform varies as a function of input and output voltages. This results in a constant loop gain and bandwidth which is independent of the magnitude of the input voltage. Also, because both the current in the inductor and the output voltage are being sensed, rather than just the output voltage, the compensation needed to account for the output filter frequency response is simplified. This results in an additional feedback loop requirement, but serves to minimize the effect of the pole from the output filter inductor [17]. Under normal regions of operation only one pole affects the stability of the system and therefore the compensation is greatly simplified, allowing a higher bandwidth to be achieved.

Peak current mode control has an additional advantage. The current limiting function is automatically provided by the additional current feedback loop. Peak current mode control also has some disadvantages. First, the current loop has low gain, which is further reduced as the current through the output filter inductor becomes discontinuous. Also, this control mode is more die area intensive due to the need for a current amplifier and related circuitry. In addition, the stability analysis

becomes more complicated than voltage mode control because two loops need to be considered, rather than just one. Finally, the system becomes unstable at duty cycles greater than 50% unless slope compensation techniques are employed [3, 7, 14, 24, 31]. This instability is well documented in the literature and can be avoided, but the necessary compensation increases circuit complexity and cost.

2.2.3 Average Current Mode Control

An average current mode control loop is shown in Figure 2.5c [18]. This method is similar to the peak current control method, except that another amplifier has been added. This high gain amplifier, referred to as the current amplifier, generates an output signal based on the current through the inductor and the error signal from the error amplifier. This has the effect of comparing the desired output current to the actual output current which allows it to have the fastest response to changes in load current of the three control methods presented here. This control method is more complicated than the peak current control method due to the extra amplifier needed, but it has the advantage of not needing slope compensation for situations in which the duty cycle exceeds 50%. As with peak current control, average current control has been well documented [6, 7, 18, 24] and design strategies have been developed to maximize its performance. This control method was a proprietary technique developed by Unitrode and remains underutilized in the industry.

After consultation with potential customers and experts in the field, the decision was made to design this high current switching regulator using voltage mode control. This type of control would provide relative simplicity, market compatibility and a robust level of performance, all at minimal cost. This thesis will concentrate solely on the design and modeling of this type of buck regulator with the understanding that the principles and ideas presented here can be used with other architectures and control methods.

Chapter 3

Regulator Top-Level Modeling

Although voltage mode has been chosen as the control method of choice due in part to its relative simplicity, the regulator top-level modeling and simulation can still present problems using traditional approaches. The ability to perform top-level modeling is important for several reasons. First, large high integration systems, such as switching regulator, are generally designed using a top down approach with a number of designers each responsible for their own subsystem, or functional block. A top level simulation provides each designer not only with subsystem performance parameters which are not immediately obvious from the system specification, but also with information regarding interactions between each subsystem. For example, in a switching regulator, variations in the magnitude of the output filter capacitor can have a rather profound effect on the soft start circuit biasing value. Secondly, a top level simulation allows a designer to gain an understanding of how a variation in an external component can effect overall system performance. For example, variations in the equivalent series resistance (ESR) of the output filter capacitor can dramatically effect the phase margin of the system. Finally, these simulations can be used to quickly confirm top level system functionality.

3.1 Spice as a Control Systems Tool

IC designs are often done on a transistor level with Spice. However, the transient response of a large system such as a regulator can take many hours to simulate. Also, the combination of a mixed-mode circuit, and one or more feedback loops can cause the simulation to either not find a bias point or worse, find a bias point which is inaccurate. To further complicate matters, bias and supply currents in a high current switching regulator can differ by as much as six orders of magnitude. This can cause a myriad of problems in circuit simulators that are unaccustomed to simulating such a wide range of simulataneous operating conditions. For example, the load current may be as high as 10 or 12 amps, yet biasing circuits for the error amplifiers, comparators and other analog circuits are typically in the range of 10 to 20 microamps.

Efforts by the designer to assist or correct Spice simulation problems through adjustment of tolerance values can often lead to inconclusive results. The designer is left with no indication from the simulator if the algorithm in the simulation engine failed due to a mathematical difficulty, or whether the system itself is unstable. Also, Spice is not generally regarded as a controls system design tool. However, systems on a chip are increasingly oriented toward integration of complex control functions where knowledge of system stability and stability margin are critical to the success of the project.

One alternative approach using Spice is to create a simplified model minus the nonlinear components such as the logic circuitry, power FETs, comparators and pulse width modulator. With this model, state space averaging could be utilized to find the open loop frequency response of the system. However, even if these simulations converge and are valid, an excessive amount of time is required to complete the analysis due to the size of the system, making it a challenge for the designer to gain intuition regarding the system performance. Therefore, it becomes difficult for a recommendation to be made to the customer about the external components necessary to get the best regulator performance for the lowest price. In an effort to decrease simulation time, macromodeling or behavioral modeling can be used instead of transistor

level simulation. In an effort to solve a majority of the problems presented here, an approach has been developed for this project using a commercially available software tool specifically engineered for control systems analysis.

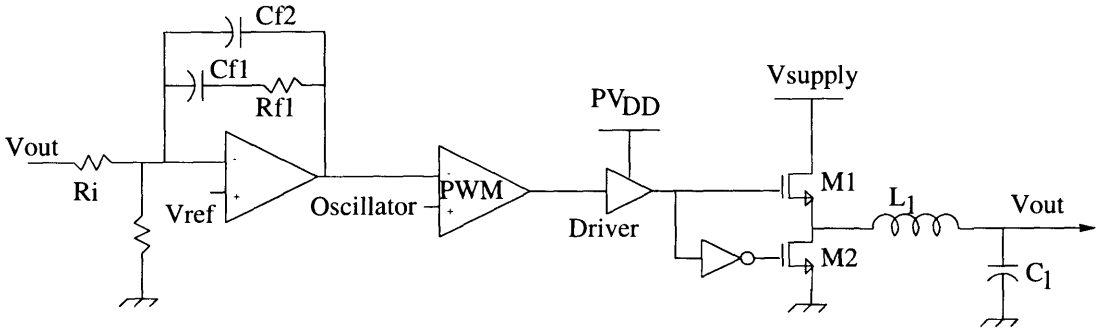
3.2 Using Matlab and Simulink as Control Systems Tools

Matlab is a mathematical software package that is used for matrix manipulation and graphical representation of data. The Matlab control system toolbox and Simulink, a graphical user interface for manipulating system transfer functions and performing system transient simulations, are used along with the basic Matlab functions in this project.

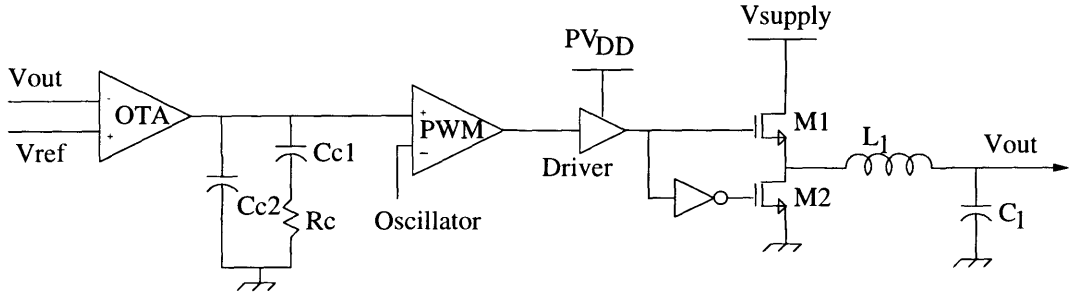
3.2.1 Basic Regulator Modeling

In designing a system such as a switching regulator, the most important advantage of using Matlab over transistor level modeling is found in the frequency response plots that can be generated. In order to find the frequency characteristics of a particular regulator, a model needs to be derived. This can be done by separating the regulator into smaller subsystems, or functional blocks, and finding the transfer function for each. Each block is then recombined into a top level schematic. Once the frequency characteristics are generated for each block in the switching regulator, these characteristics can then be used to determine the output filter components, the necessary compensation values, and the gain bandwidth product of the error amplifier to ensure stability and the desired phase margin is achieved. A basic switching regulator, using an operational amplifier as the error amplifier and employing voltage mode control, is shown in Figure 3.1a. Figure 3.1b shows the same regulator with an operational transconductance amplifier (OTA) used as the error amplifier.

To express the system in terms of a transfer function, the first step is to remove the nonlinear elements by using state space averaging [25, 27, 36]. For example,



a.



b.

Figure 3-1: a) Switching regulator using an operational amplifer as the error amplifier. b) Switching regulator using an operational transconductance amplifier as the error amplifier.

the power FET driver applies either a logic 1 (represented as PV_{DD} in Figure 3.1) or ground (logic level 0) to the gate of the top power FET (M1) which biases the device either completely off ($V_{gs} < V_{th}$) or into conduction ($V_{gs} > V_{th}$). Therefore, the node connecting the two power FETs (the source of M1 and the drain of M2) is either close to V_{supply} or to ground. The output filter, which includes the inductor L_1 and capacitor C_1, is an impedance divider. The input voltage to this impedance divider can be expressed as $(V_{supply})D$ where D is the duty cycle generated by the PWM. This input voltage is the average voltage applied to the output filter. As the duty cycle increases, M1 is on for a longer period of time and the input filter sees a higher average voltage. This analysis applies to the regulator with either an OTA or operational amplifier because the functionality of the output filter, PWM, power FETs, and FET driver remains the same. For now, it will be assumed that both L_1 and C_1 are ideal and that the power FETs have zero on-resistance. The transfer function between the FET driver and the output voltage is shown in the following equation.

$$H(s) = \frac{Z_c}{Z_l + Z_c} \tag{3.1}$$

Letting $Z_c = \frac{1}{Cs}$ and $Z_l = Ls$, the transfer function reduces to Eq. 3.2.

$$H(s) = \frac{1}{LCs^2 + 1} \tag{3.2}$$

In the following chapter, the model will be expanded to include effects of parasitic elements and non-ideal components. In Matlab, a transfer function can be described as a ratio of two vectors, one for the numerator and one for the denominator, with the coefficients of each vector arranged in descending order of the power of the complex variable s. For example, the previous transfer function could be written as `num=[1]` and `den=[LC 0 1]`. Therefore, this relationship allows the power FETs, FET driver, and PWM to be eliminated in order to express this system in terms of a single transfer function.

The output voltage of the error amplifier is proportional to the duty cycle of the regulator. However, this model has so far neglected the gain due to the PWM and

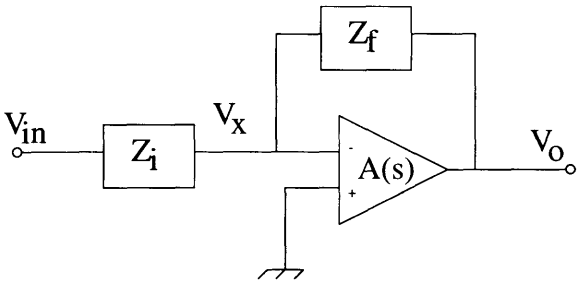


Figure 3-2: Operational amplifier shown with input and feedback impedance networks.

oscillator. The output of the error amplifier, regardless of which amplifier architecture is chosen, is compared to a sawtooth waveform to generate a digital signal whose pulse width determines the duty cycle. The smaller the range of the oscillator waveform, the less the output of the error amplifier is required to swing for a given input. This results in higher loop gain. The gain due to the PWM is therefore given as

$$A_{PWM} = \frac{V_{supply}}{\Delta V_{oscil}} \cdot e^{-j(2\pi f)T/2} \tag{3.3}$$

where the peak-to-peak value of the oscillator sawtooth voltage is given by V_{oscil} [18]. T is the period associated witht the switching frequency. The term $e^{-j(2\pi f)T/2}$ represents the phase delay due to the PWM. The transfer function derivations described thus far are valid for systems employing an operational amplifier or OTA (Figure 3.1a or 3.1b). However, frequency domain modeling of the error amplifier is strongly dependent on the type of amplifier architecture chosen.

The operational amplifier and its compensation can be drawn as shown in Figure 3.2. The reference voltage for the regulator is attached to the positive terminal of the operational amplifier, which is small signal ground. Nodal analysis provides us with the following relationships:

$$\frac{V_{in} - V_x}{Z_{in}} + \frac{V_o - V_x}{Zf} = 0 \tag{3.4}$$

$$V_o = AV_x \tag{3.5}$$

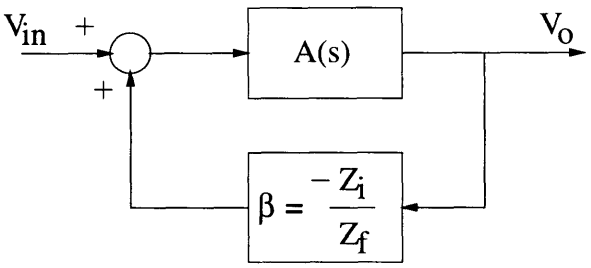


Figure 3-3: Block diagram of an operational amplifier and feedback factor β.

where A is a function of frequency. Solving for V_o/V_{in} gives

$$\frac{V_o}{V_{in}} = \frac{-A}{\frac{Z_{in}}{Z_f}(A-1)-1} \tag{3.6}$$

Assuming $A - 1 \simeq A$, Eq. 3.6 reduces to

$$\frac{V_o}{V_{in}} = \frac{A}{1+A\beta} \tag{3.7}$$

where β is $-Z_{in}/Z_f$. Therefore, the operational amplifier and compensation can be modeled as a block diagram as shown in Figure 3.3. Depending on the compensation used, Z_{in} may be a single resistor or some combination of resistors and capacitors. Likewise, Z_f may be a single capacitor or a combination of several passive components.

Once the transfer functions of Z_i, Z_f, and A have been found, Matlab can automatically find the complete numerical transfer function of Figure 3.3 with the command `feedback`, which calculates the closed loop transfer function given the feedback factor and the plant transfer function [29]. An example of this modeling process can be illustrated with the error amplifier configuration shown in Figure 3.1a. The input impedance R_i is assumed to be 1 kΩ, the feedback resistor R_f is 5 kΩ, and the feedback capacitors are $C_{f1} = 5600$ pF and $C_{f2} = 4700$ pF. Solving symbolically,

$$Z_i = R_i \tag{3.8}$$

and

$$Z_f = \frac{1}{C_{f2}s \parallel (R_f + \frac{1}{C_{f1}s})} \tag{3.9}$$

A simplified expression for Z_f can be given as:

$$Z_f = \frac{R_f C_{f1} s + 1}{s(R_f C_{f1} C_{f2} s + C_{f1} + C_{f2})} \tag{3.10}$$

When vectors representing transfer functions are defined, the convention is used that a `1` at the end of the variable name indicates the numerator and a `2` represents the denominator variable. For example, `h1` would be the numerator of H(s) and `h2` would be the denominator of H(s). An example of how Matlab is used to manipulate these transfer functions into the form shown in Figure 3.3 is provided below:

```
% feedback component values
>> rf=5000;
>> cf1=5600e-12;
>> cf2=4700e-12;
>> ri=1000;

% compute feedback network transfer function
>> zi1=[ri]
zi1 =
         1000
>> zi2=[1]
zi2 =
     1
>> zf1=[rf*cf1 1]
zf1 =
    0.0000    1.0000
>> zf2=[rf*cf2*cf1 cf1+cf2 0]
zf2 =
```

```
1.0e-07 *
 0.0000    0.1030         0
```

The feedback factor shown in Figure 3.3 is given as Z_i/Z_f. This can be expressed using Matlab by using the `series` command. This command takes two transfer functions and calculates the equivalent transfer function for their series combination. Therefore, Z_i/Z_f can be expressed as the series combination of Z_i and $1/Z_f$, or in Matlab, as `series(zi1, zi2, zf2, zf1)`. To account for the situation where the feedback factor contains a higher order numerator than denominator, which is not allowed in Matlab, a high frequency pole is added to the transfer function. From a practical standpoint, this high frequency pole can be ignored, but from a mathematical standpoint, it causes the order of the numerator to be the same as that of the denominator, satisfying the Matlab criteria. Adding this pole to the feedback factor is the same as adding a zero to Z_f. Instead of writing

$$Z_{f1} = R_f C_{f1} s + 1 \tag{3.11}$$

Z_{f1} can be written as

$$Z_{f1} = (R_f C_{f1} s + 1)(\frac{1}{f_h} s + 1) \tag{3.12}$$

where f_h is the location of the high frequency zero in rads/sec. This technique is illustrated below:

```
% high frequency zero
>> fh=1e9;


% recompute numerator coefficients
>> zf1=[rf*cf1/fh rf*cf1+(1/fh) 1]
zf1 =
    0.0000    0.0000    1.0000


% compute new feedback factor
```

```
>> [comp1,comp2]=series(zi1,zi2,zf2,zf1)
comp1 =
   1.0e-04 *
    0.0000    0.1030         0
comp2 =
    0.0000    0.0000    1.0000
```

To complete the error amplifier model shown in Figure 3.3, the open loop transfer function of the operational amplifier needs to be defined. If the operational amplifier is assumed to have a very high low frequency gain and a single dominant pole, the gain as a function of frequency can be described as

$$A(s) = \frac{A}{\frac{1}{f_p}s + 1} \tag{3.13}$$

where f_p is the dominant pole location and A is the low frequency gain. For illustration, if we let $A = 100,000$ and $f_p = 20$ rad/sec, the open loop characteristic of the operational amplifier can be modeled in Matlab as shown below:

```
% low frequency gain
>> a=1e5;

% dominant pole location (in rad/sec)
>> fp=20;

% A(s)
>> a1=[a]
a1 =
      100000
>> a2=[1/fp 1]
a2 =
    0.0500    1.0000
```

Now the model for the operational amplifier and its associated compensation network shown in Figure 3.3 is complete. The transfer function for the entire system can now be found using the `feedback` command. The first two variables passed to the feedback command are the numerator and denominator coefficients of the plant. The second set of variables is the numerator and denominator coefficients of the feedback factor. The fifth variable indicates whether positive or negative feedback should be used.

```
% compute transfer function of operational amplifier
% with compensation
>> [op1,op2]=feedback(a1,a2,comp1,comp2,-1)
op1 =
   1.0e+05 *
         0    0.0000    0.0000    1.0000
op2 =
    0.0000    0.0000    1.0800    1.0000

% plot transfer function
>> bode(op1,op2)
```

The `bode` command plots the frequency characteristics of the operational amplifier and compensation with the result shown in Figure 3.4. The compensation values were chosen for illustration purposes only. In this example, it can be seen that only 20° of positive phase was added. If this was insufficient for a particular design, a simple iteration of the subsystem component and parameter values could be performed to optimize the amount of phase addition and its location. The high frequency zero added to Z_f can also be seen in Figure 3.4. If this zero occurred at a low enough frequency that it impacted the accuracy of the model, its effect can easily be minimized by redefining the variable f_h to a larger value and having Matlab recalculate the transfer function coefficients.

The transfer function of an operational transconductance amplifier (OTA) can be

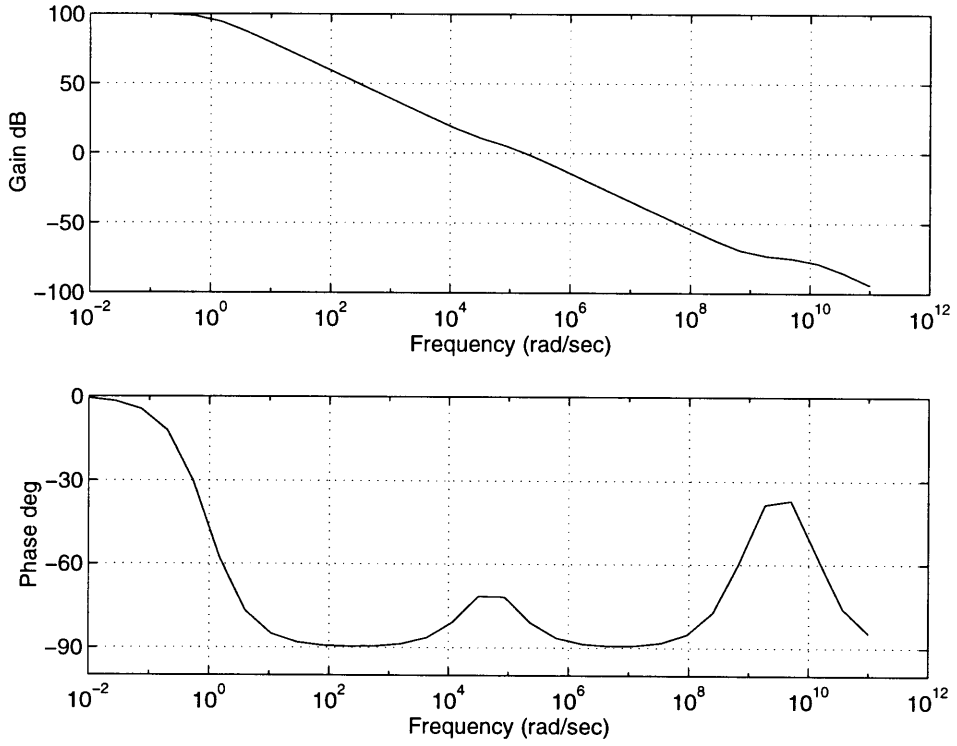


Figure 3-4: Frequency characteristics of an operational amplifier and compensation.

derived in a manner similar to that just described for an operational amplifier. The principle difference between the two derivations stems from the differences in compensation methods used. The compensation for the operational amplifier results in a closed feedback loop around the amplifier and contained within the outer regulator loop. The OTA compensation, however, is referenced to ground. Therefore, the only feedback path around the OTA is the regulator loop itself.

An OTA provides an output current proportional to the input differential voltage. This relationship is given by

$$I_{out} = g_m \Delta v_{in} \tag{3.14}$$

where the transconductance of the amplifier, g_m, is a function of frequency. The output current is converted to a voltage that is applied to one input of the PWM comparator. The conversion is performed by the compensation components and the output impedance of the OTA. Because the compensation network contains capacitive elements between the output of the OTA and ground, the output impedance of the

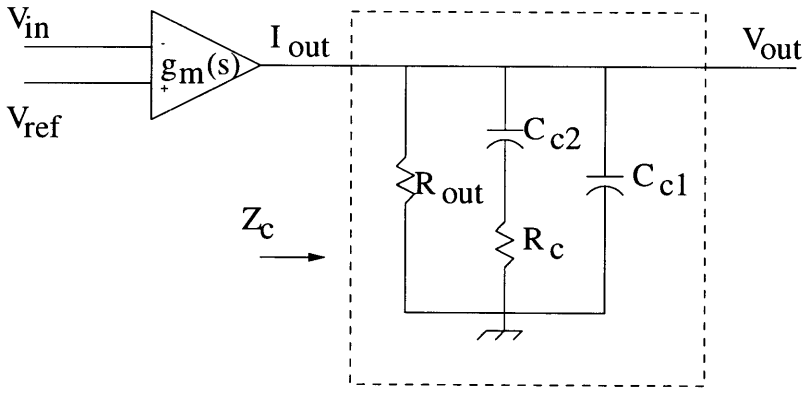


Figure 3-5: Block diagram of an error amplifier and compensation that includes the effect of the output impedance of the OTA.

OTA determines the DC gain. This gain is given by

$$A_v = g_m R_{out} \tag{3.15}$$

where R_{out} is the output impedance of the OTA. The output voltage of the OTA and compensation network, modeled as shown in Figure 3.5, is given by

$$V_{out} = I_{out} Z_c \tag{3.16}$$

where Z_c is the impedance of the compensation network in parallel with the output impedance of the OTA. Substituting Eq. 3.14 into Eq. 3.16, the transfer function of the OTA and compensation network is found to be

$$\frac{V_{out}}{V_{in}} = g_m Z_c \tag{3.17}$$

An example of modeling the OTA with Matlab can be illustrated using the compensation network shown in Figure 3.1b. The compensation values are assumed to be as follows: $C_{c1} = 100$ pF, $C_{c2} = 10$ nF, and $R_c = 20$ kΩ. Also, the output impedance of the OTA is assumed to be 1 MΩ with a transconductance of 1 $m\Omega^{-1}$ at low frequencies. The OTA is assumed to have a single dominant pole at a frequency of

$w_p = 10^6$ rad/sec. Therefore, the transconductance of the OTA can be modeled as

$$G_m = \frac{g_m}{\frac{1}{w_p}s + 1} \tag{3.18}$$

The compensation network and output impedance can be modeled by

$$Z_c = R_{out} \parallel (R_c + \frac{1}{C_{c2}s}) \parallel \frac{1}{C_{c1}s} \tag{3.19}$$

Manipulation and simplification of Eq. 3.19 provides the following relationship for the current to voltage conversion network:

$$Z_c = \frac{sC_{c2}R_cR_{out} + R_{out}}{s^2C_{c1}C_{c2}R_{out}R_c + s(R_{out}C_{c2} + R_{out}C_{c1} + C_{c2}R_c) + 1} \tag{3.20}$$

Matlab can be used to find the numerical representations of these transfer functions as shown.

```
% compensation values
>> cc1=100e-12;
>> cc2=10e-9;
>> rc=20e3;

% output impedance of OTA
>> rout=1e6;

% transconductance
>> gm=1e-3;

% dominant pole location
>> wp=1e6;

% compute gm(s)
>> gm1=[gm]
```

```
gm1 =
   1.0000e-03
>> gm2=[1/wp 1]
gm2 =
    0.0000    1.0000

% current to voltage conversion network
>> zc1=[cc2*rc*rout rout]
zc1 =
         200     1000000
>> zc2=[cc1*cc2*rout*rc rout*cc2+rout*cc1+cc2*rc 1]
zc2 =
    0.0000    0.0103    1.0000

% compute transfer function of OTA w/compensation
>> [ota1,ota2]=series(gm1,gm2,zc1,zc2)
ota1 =
   1.0e+03 *
         0         0    0.0002    1.0000
ota2 =
    0.0000    0.0000    0.0103    1.0000

% plot transfer function
>> bode(ota1,ota2)
```

An example of the open loop transfer function of the OTA with compensation network is shown in Figure 3.6.

After each section of the regulator has been modeled in the frequency domain, the subsystems can be combined as shown in Figure 3.7 to find the complete open loop transfer function of the system. Figure 3.7a shows the model of a regulator using an operational amplifier as the error amplifier while Figure 3.7b shows an OTA used as

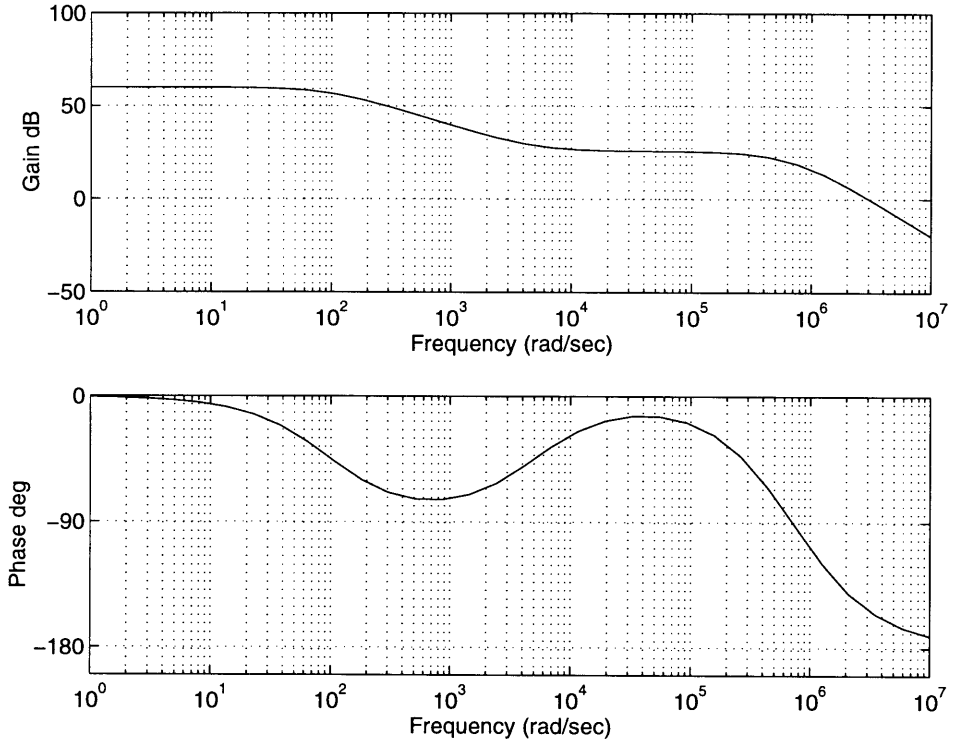
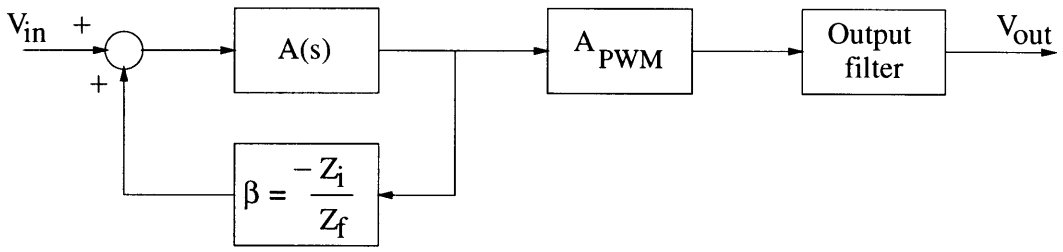


Figure 3-6: Open loop transfer function of an OTA and compensation.

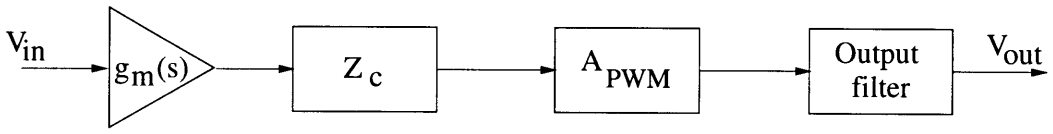
the error amplifier. After the transfer functions for each block have been determined, the complete transfer function can be found using the Matlab `series` command. Assuming an effective PWM comparator gain of 2, an output filter capacitor of 1000 μF, an output filter inductor of 2 μH, and that the OTA described above is used, the complete system function can be found as shown below.

```
% output filter
>> l=2e-06;
>> c=1000e-6;

% compute filter transfer function
>> f1=[1]
f1 =
     1
>> f2=[l*c 0 1]
```



a.



b.

Figure 3-7: Block diagram used in Matlab of the regulator using an a) operational amplifier and b) an OTA.

```
f2 =

    0.0000         0    1.0000

>> bode(f1,f2)


% pwm gain

>> pwm=2;

>> f1=f1*pwm

f1 =

     2


% compute complete system transfer function

>> [r1,r2]=series(ota1,ota2,f1,f2)

r1 =

   1.0e+03 *

         0         0         0         0    0.0004    2.0000
```

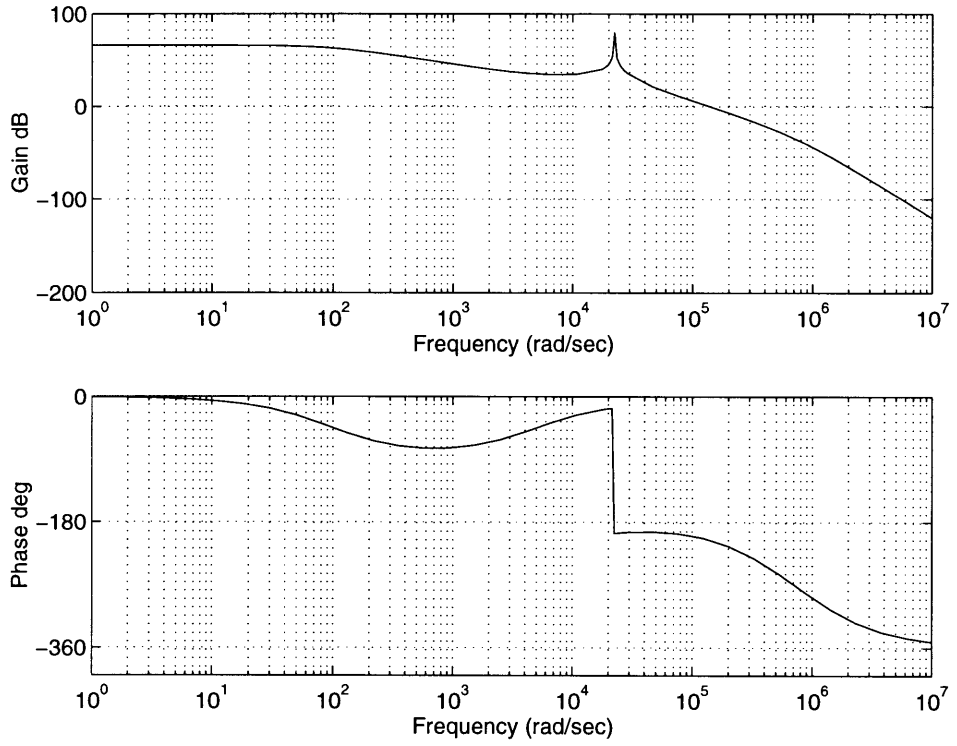


Figure 3-8: Complete system transfer function for the example in Section 3.2.1.

```
r2 =

    0.0000    0.0000    0.0000    0.0000    0.0103    1.0000

% plot system transfer function
>> bode(r1,r2)
```

The complete system transfer function for this example is shown in Figure 3.8. At the frequency at which the magnitude crosses 0 dB, or a gain of 1, the phase is already more negative than $-180°$ which indicates the system is unstable when in a unity gain feedback configuration. At this point, the designer would be faced with another iteration to determine which parameter, or parameters, caused the system to be unstable. This is neither efficient nor elegant. The design methodology presented thus far needs to be improved to allow greater flexibility in the iterative process.

3.2.2 Matlab Scripts

We now have a basic tool for determining the effects of changing key design parameters on the stability of the regulator system. However, redefining these parameters to iterate towards an optimum design solution is both cumbersome and vulnerable to errors. In addition, the assumption that each designer be familiar with Matlab does not exploit the full advantage of the tool. Both of these issues can be avoided by developing a Matlab program, or "script" to assist in the regulator design process. The script will implement an algorithm that allows the system level design to be interactive, requiring only that the designer enter key design parameters when prompted. No detailed understanding of the inner workings of the script, or knowledge of the Matlab tool is required. The algorithm should be fairly generic to allow its use with future switching regulator design programs. The script will also provide almost instantaneous feedback on system performance when any key design variable, whether it be internal to the IC or associated with an external component, is modified. Also, Matlab can be used to automatically calculate compensation values to ensure system stability. A simple example of a Matlab script is shown below.

```
%  scriptexample.m
%  This script plots the transfer function of an LC filter.
l=input([blanks(5) 'Enter inductor value (in uH): -->  ']);
l=l*1e-6;
c=input([blanks(5) 'Enter capacitor value (in uF): -->  ']);
c=c*1e-6;
f1=[1]
f2=[l*c 0 1]
bode(f1,f2)
title('LC filter')
```

This script prompts the user for capacitor and inductor values and then calculates and plots the transfer function of an LC filter. Matlab also supports control flow statements, such as for and while loops, as well as if statements, allowing the iterative

design process to be automatic and self contained. As the model for the switching regulator becomes more involved, including the addition of passive device parasitics and other non-idealities, this design methodology will become more of a necessity than a convenience.

3.3 Expanding the Regulator Model in Matlab

The regulator model developed thus far is very simplistic, accounting only for first order effects of the output filter, error amplifier and compensation, and PWM gain. To create a more viable representation of the regulator, several important design variables need to be incorporated into the regulator model, including output filter and board parasitics, power FET transfer characteristics, error amplifier limitations and effects due to input filter components.

Possibly the most important effect on system stability is that contributed by the parasitics of the output filter components. The two primary modeling concerns are the equivalent series resistance (ESR) and equivalent series inductance (ESL) of the output filter capacitor. These non-idealities are especially important for two reasons. First, since the output filter can see load current changes of more than 10 amps in a very short period of time, the magnitude of ESR and ESL can prove to be the limiting factor in how well the regulator IC can maintain transient load regulation [12]. Secondly, the ESR and ESL each contribute a zero to the output filter transfer function which can dramatically change the phase margin of the system. The output filter inductor also has an ESR which can effect the system phase margin and needs to be modeled.

The interconnect between the voltage regulator output filter and the microprocessor itself contains parasitic resistance and inductance that needs to be included in the system model. In addition, the microprocessor socket contains a number of bypass capacitors to ensure that high frequency, high load current changes do not result in a loss of load regulation. These components are part of the power distribution network and are included in the higher order model.

The simple model represents the power FETs as ideal switches with zero on-resistance. The magnitude of the channel impedance, or R_{DSon}, of the actual power FET is on the same order as the ESR of the output filter capacitor and therefore must be included. Also, the error amplifier, whether it be an operational amplifier of OTA, was assumed to have a single pole response. However, higher frequency, non-dominant poles do exist and may have an effect on system stability. In addition, if an OTA is used, the model should account for variations in output impedance with frequency.

Finally, some applications require that the input supply voltage be isolated from the regulator system using an LC filter. The filter components and associated parasitics need to be included in the complete model due to the effect it may have on the system transfer function.

Developing an accurate and concise model for a switching regulator power supply is a laborious process. However, the benefits of such a model, from rapid determination of system phase margin to accurate prediction of load regulation capability, is invaluable to the success of any regulator design program.

As the model begins to evolve using state space averaging techniques and validated assumptions, its complexity warrants the use of the Matlab design methodology. Used in conjunction with Simulink and the available Spice simulation engines, complete system level understanding and circuit level performance requirements will avail itself to members of the design team. Chapter 4 begins a more detailed analysis of this design methodology and outlines more specifically the additional factors necessary to complete the higher order model.

Chapter 4

Developing a Regulator Design Methodology using a Matlab Script

Matlab has been shown to be a very useful tool that can be used to quickly determine the frequency response and stability information of a regulator system for various combinations of design parameters. Also, the ability to write a script in Matlab allows the interface to be more intuitive, simple to use, and reduces errors that could occur if large numbers of commands had to be entered by hand. A script allows the system level design of a switching regulator to be interactive. It can also provide almost instantaneous feedback on system performance when any key parameter, either external or internal to the system, is modified. To use Matlab most effectively, a more detailed model of the regulator needs to be derived. This model can be used to develop a methodology to reduce regulator design time and increase the likelihood of initial project success.

4.1 Power Distribution Network

A modern high performance microprocessor, like the Intel Pentium® Pro, can have a change in load current from a minimum of 0 amps to a maximum of 14 amps at a rate

of up to 1000 amps/μs. These load current transients can occur at a frequency from between 100 Hz to 100 kHz [15]. Because of these large transient load current changes and switching speeds, the interconnect between the voltage regulator module and the microprocessor becomes a critical design parameter that can't be ignored. This interconnect system, comprised of parasitic components and bypass capacitors, is referred to as the power distribution network [18]. The parasitic inductance contained within this network limits the ability of the regulator to provide the required load current slew rate at the microprocessor input pins. Bypass capacitors are therefore added to the network as physically close to the microprocessor as possible to store charge and allow the regulator slew rate requirement to be greatly relaxed. Because of the particularly demanding load current transient requirements, the power distribution network design becomes critical to ensure the transient specifications can be met.

The interactive Matlab script will begin by asking the user to specify the components and parasitics in the power distribution bus between the regulator and the microprocessor. A plot of parasitic output impedance versus frequency will then be generated. The user will be prompted for the maximum regulator output impedance to allow the output to slew quickly enough to meet the regulator specifications, and two impedance lines will be drawn. The frequency at which they intersect is the necessary bandwidth of the error amplifier and compensation. Before the script is developed, however, an accurate model of the power distribution network must be created.

4.1.1 Modeling the Power Distribution Network in Matlab

To understand the effects of the power distribution network on the transient response of the regulator, the first step is to develop a model of the parasitic inductance and resistance values as well as the bypass capacitors. A simplified model of the regulator, power distribution network, and microprocessor is shown in Figure 4.1 [18]. C_b represents the bypass capacitors placed at the microprocessor pins. R_{cb} is the equivalent ESR of these capacitors, and L_{cb} is their equivalent ESL. The Pentium® Pro, for example, can contain as many as 30 or 40 1 μF bypass capacitors arranged in

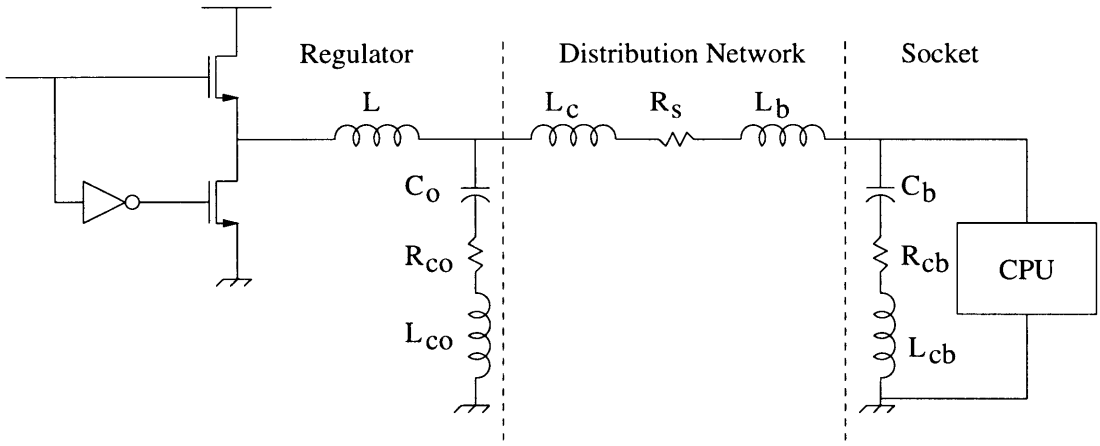


Figure 4-1: Power distribution network model.

parallel. Likewise, C_o represents the equivalent capacitance of the regulator output filter. R_{co} is the equivalent ESR and L_{co} is the equivalent ESL of the filter capacitor. L_c models the connector pin inductance of all the pins in parallel. L_b is the PC board power trace inductance between the regulator module and the microprocessor. Finally, R_s is the resistance between the regulator module and microprocessor. If a second type of output filter capacitor is used in parallel with the first, represented by C_o, R_{co}, and L_{co}, the model can be modified by adding another RLC combination in parallel with the original to accurate represent all of the output filter capacitors.

The output impedance of the regulator module, as seen by the microprocessor, determines how much variation will occur in the supply voltage for a given load current change. If the impedance looking into the output of the regulator and power distribution network is below a certain threshold for all frequencies, the regulated voltage will vary less than the maximum acceptable value when a current transient occurs. This impedance can be plotted as a function of frequency by first solving for the impedance symbolically, then defining component values in Matlab and using the `bode` command.

Referring to Figure 4.1, the output impedance can be found by manipulating Eq. 4.1.

$$Z_{out} = \frac{1}{\frac{1}{C_b s} + L_{cb}s + R_{cb}} \parallel (\frac{1}{C_o s} + L_{co}s + R_{co} + R_s + L_c s + L_b s) \qquad (4.1)$$

The impedance as a function of frequency is given by:

$$Z_{out}(s) = \frac{As^4 + Bs^3 + Cs^2 + Ds + 1}{Es^3 + Fs^2 + Gs} \tag{4.2}$$

where

$$\begin{aligned}
A &= C_o C_b L_{cb}(L_c + L_b + L_{co}) \\
B &= (L_c + L_b + L_{co})R_{cb}C_oC_b + L_{cb}C_oC_b(R_{co} + R_s) \\
C &= L_{cb}C_b + C_o(L_c + L_b + L_{co}) + R_{cb}C_o(R_{co} + R_s)C_b \\
D &= (R_{co} + R_s)C_o + R_{cb}C_b \\
E &= C_bC_o(L_c + L_b + L_{cb} + L_{co}) \\
F &= C_bC_o(R_{co} + R_{cb} + R_s) \\
G &= C_o + C_b
\end{aligned}$$

In the vector form used in Matlab, this is written as:

```
num=[co*cb*lcb*(lc+lb+lco) (lc+lb+lco)*rcb*co*cb+lcb*co*cb*(rco+rs)
lcb*cb+co*(lc+lb+lco)+rcb*co*(rco+rs)*cb (rco+rs)*co+rcb*cb 1];
den=[cb*co*(lc+lb+lcb+lco) cb*co*(rco+rcb+rs) (co+cb) 0];
```

The output filter capacitor value, C_o; effective ESL, L_{co}; and effective ESR, R_{co}; is usually the result of having several capacitors arranged in parallel. If the capacitors are equal, their impedance is given by the following equation

$$Z_{eq} = (\frac{1}{C_1 s} + L_1 s + R_1) \parallel (\frac{1}{C_2 s} + L_2 s + R_2) \tag{4.3}$$

where $R_1 = R_2$, $C_1 = C_2$, and $L_1 = L_2$. Therefore, Eq. 4.3 simplifies to:

$$Z_{eq} = \frac{(\frac{1}{C_1 s} + L_1 s + R_1)^2}{2(\frac{1}{C_1 s} + L_1 s + R_1)} \tag{4.4}$$

and finally to:

$$Z_{eq} = \frac{1}{2C_1 s} + \frac{L_1 s}{2} + \frac{R_1}{2} \tag{4.5}$$

This means that if n identical capacitors are placed in parallel, the total capacitance is given as

$$C_o = c \cdot n \tag{4.6}$$

where c is the capacitance associated with each. Also, the effective ESR is

$$R_{so} = \frac{ESR}{n} \tag{4.7}$$

where ESR is the equivalent series resistance for each capacitor. Likewise,

$$L_{co} = \frac{ESL}{n} \tag{4.8}$$

where ESL is the equivalent series inductance of each capacitor.

4.1.2 Effect of Power Distribution Network on System Performance

The previous section completed the derivation of the output impedance model over frequency. This model can be used to determine if the regulator is capable of meeting the transient specifications. The output voltage is only allowed to vary a certain number of millivolts for a current change of several amps. This is equivalent to specifying a maximum allowable output impedance [18]. The current can not change instanteously, implying that the required output impedance curve has a zero at a frequency given by

$$f_z = \frac{0.35}{t_r} \tag{4.9}$$

where t_r is the load current risetime. Combining these two curves gives

$$Z_{req} = Z_{max} \frac{1}{\frac{1}{f_z 2\pi} s + 1} \tag{4.10}$$

The power distribution network is capable of handling the load transient when the actual output impedance is less than the required impedance, Z_{req}. Therefore, the

frequency at which these two curves intersect is the minimum loop bandwidth needed to meet the transient voltage specification. This analysis is very useful because it allows the designer to understand the tradeoff between the size of the filter capacitor and the required loop bandwidth. As the filter capacitor is increased, the loop bandwidth specification is relaxed. An example of this type of analysis is illustrated in Figure 4.2.

4.1.3 Development of Matlab Script to Determine Loop Bandwidth Requirement

Now that the importance of the output impedance has been established in determining the necessary regulator loop bandwidth, the process can be automated with a Matlab script. First, the variables in the power distribution network transfer function are defined.

```
lco=0.34e-9;
cb=40e-6;
rcb=3e-3;
lcb=12e-12;
rs=1.29e-3;
lc=.26e-9;
lb=.5e-9;
```

Next, the maximum impedance in the mid-frequency region to meet the output current slew rate requirement without exceeding the voltage transient specification is defined by the user.

```
imp=input([blanks(5) 'Enter maximum output impedance in
mid-frequency region (ohms): -->  ']);
```

Because the current risetime is not infinite, the maximum impedance curve has a zero whose location is a function of the rise time as discussed in the previous section. This zero location is calculated after prompting the user for the load current rise time.

```
tr=input([blanks(5) 'Enter the load current rise time (s).
(Pentium Pro is 10-15 ns): -->  ']);
fc=0.35/tr;
```

The total filter capacitance and effective ESR are found using the number of capacitors, and the ESR and capacitance associated with each.

```
capnum=5;
capvalue=330e-6;
capesr=0.04;
co=capvalue*capnum;
rco=capesr/capnum;
```

The expression for the output impedance of the power distribution network as a function of frequency was calculated in Section 4.1.1 and shown in Eq. 4.2. The output impedance is expressed below in Matlab format as Z_{out}(s) = num(s)/den(s).

```
num=[co*cb*lcb*(lc+lb+lco) (lc+lb+lco)*rcb*co*cb+lcb*co*cb*(rco+rs)
lcb*cb+co*(lc+lb+lco)+rcb*co*(rco+rs)*cb (rco+rs)*co+rcb*cb 1];
den=[cb*co*(lc+lb+lcb+lco) cb*co*(rco+rcb+rs) (co+cb) 0];
```

To plot the output impedance versus frequency, the `bode` command is used. To satisfy the Matlab requirement that the order of the numerator be less than or equal to the order of the denominator, an extra pole is added to increase the order of the denominator. This pole is represented by the variable `temp` and is added at a frequency of 10^{12} rad/sec in this example, so that it does not affect the accuracy of the impedance curve in the region of interest. After the pole has been added, the output impedance is given by

$$Z_{out}(s) = \frac{num(s)}{den2(s)} \tag{4.11}$$

where `den2` is found in Matlab as shown below.

```
temp=[1e-12 1];
den2=conv(temp,den);
```

The magnitude and phase of $Z_{out}(s)$ as a function of frequency is obtained using the `bode` command. The variables `mag`, `phase`, and `w` are each stored in a column vector. The frequency variable `w` is converted from rad/sec to Hz and stored in the variable `f`. The phase vector generated is not used. The mag vector represents the output impedance over the frequencies contained in `f`.

```
[mag,phase,w]=bode(num,den2);
f=w/2/pi;
```

The output impendance as a function of frequency has now been determined with the approximation that an additional high frequency pole was added to satisfy the Matlab stability criteria. The data found is therefore valid from DC until a decade below the frequency of this pole. The required output impedance is found from Eq. 4.10 in Section 4.1.2. and can be expressed as

$$Z_{req}(s) = \frac{imp}{\frac{1}{f_c}s + 1} \tag{4.12}$$

As before, a high frequency pole is added to $Z_{req}(s)$ so that the order of the denominator is the same as the order of the numerator. The command `freqresp` is a low-level command used in the definition of the `bode` command in Matlab. It generates the magnitude of the output impedance as a function of the frequency vectors provided. The vector `mag1` contains the maximum output impedance of the regulator to meet the voltage transient specification.

```
[mag1]=freqresp([1/fc*imp imp],[1e-12 1],f);
```

All of the data needed to determine the required loop bandwidth is contained in the three vectors `mag`, `mag1`, and `f`. A plot of the required output impedance, contained in the vector mag, is constructed on a log log scale. The axis is rescaled to remove information distorted by the addition of the high frequency pole f_h. This is done by saving the current axis information in the vector `v`, and then rescaling such that the magnitude scale remains the same and the frequency axis is limited to 10^{10} rad/sec.

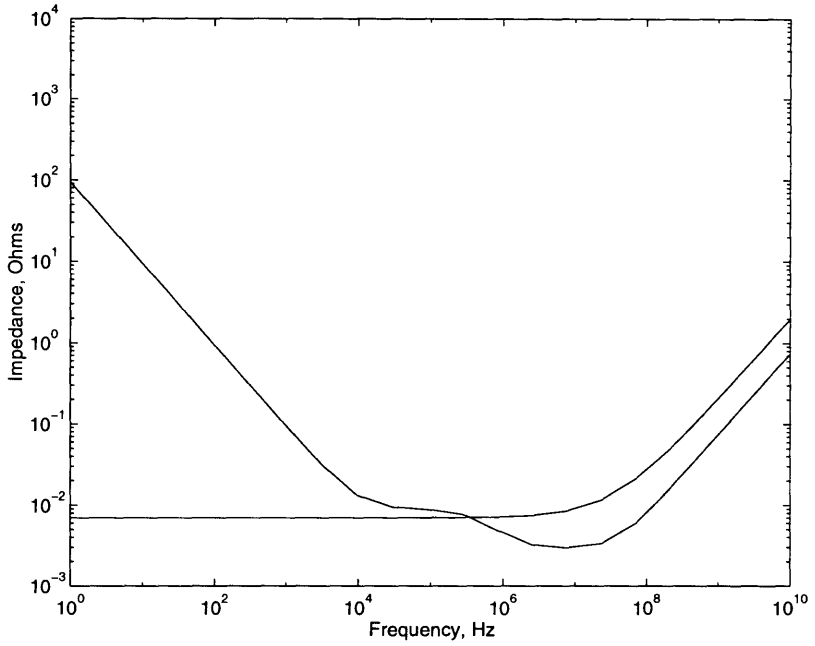


Figure 4-2: Ouput impedance versus frequency plots generated by Matlab.

```
figure
loglog(f,mag)
v=axis;
axis([1 1e10 v(3) v(4) ])
```

The maximum allowable output impedance is plotted on the same axis, in red. Then the axes are labeled.

```
hold on
loglog(f,mag1,'r')
xlabel('Frequency, Hz')
ylabel('Impedance, Ohms')
hold off
```

Referring to Figure 4.2, the user is provided with a visual indication of where the two impedance lines intersect. Matlab determines the location of this point by comparing the two vectors, `mag` and `mag1`, starting with the highest frequency and iterating the frequency vector `f` towards zero. Each time the maximum acceptable impedance is higher than the output impedance, the variable `loopbw` is overwritten

with the new value of frequency. Therefore, the required regulator loop bandwidth is given as the frequency at which the maximum acceptable impedance is lower than the output impedance. If the intersection of the two impedance curves occur more than once, the loop bandwidth is given as the highest frequency at which they intersect.

```
loop=length(f);
while mag(loop) < mag1(loop), loopbw=f(loop);, loop=loop-1; end
```

If the output filter component values provided by the user, in combination with the power distribution network values, result in a required loop bandwidth of greater than 10^{10} rad/sec, the variable `loopbw` is set to zero to indicate that the entered values are unreasonable.

```
if mag(length(f)) > mag1(length(f)),
     loopbw=0;
end
```

The script allows the user to then modify the component values and observe immediately the effect this will have on the system bandwidth requirements.

4.2 Output Filter and Power Devices

A simple transfer function for the regulator output filter was derived in Section 3.2.1. and was given as: $H(s) = 1/(LCs^2 + 1)$ (Eq. 3.2) This transfer characteristic was determined to be inadequate since it neglects the parasitics associated with the filter capacitor and inductor and ignores the effect due to the finite channel impedance of the power FETs. A more detailed model needs to be derived that accounts for the ESR and ESL of the capacitor, the ESR of the inductor, and the R_{DSon} of the power FETs. since these parameters have a significant effect on the regulator phase margin. Also, since this switching regulator needs to maintain the required supply voltage to the microprocessor from a no load (Stop Clock) condition to a maximum load demand in excess of 10 amps, understanding the transient performance dependence on these parameters becomes critical.

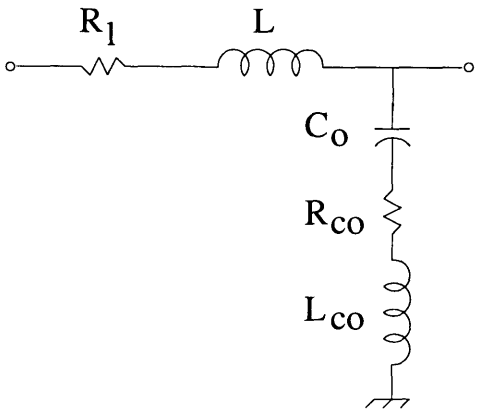


Figure 4-3: Model of the output filter used to find its transfer function.

4.2.1 Modeling Output Filter and Power Devices

Figure 4.3 illustrates a more complete model of the output filter, showing the ESR (R_{co}) and ESL (L_{co}) of the capacitor and ESR (R_l) of the inductor. Using a simple voltage divider relationship, the new transfer function is given as:

$$H(s) = \frac{(L_{co}C_o)s^2 + R_{co}C_o s + 1}{(L + L_{co})C_o s^2 + C_o(R_{co} + R_l)s + 1} \tag{4.13}$$

To show the impact of including the filter parasitics, a comparison of the transfer characteristics of the two models is illustrated in Figure 4.4 for the following component values.

L=2.2 μH

C_o=1650 μF

L_{co}=1 nH

R_{co}=10 mΩ

R_l=10 mΩ

The zeros of Eq. 4.14 are found using the following relationship.

$$\omega_z = \frac{-R_{co}C_o \pm \sqrt{(R_{co}C_o)^2 - 4L_{co}C_o}}{2L_{co}C_o} \tag{4.14}$$

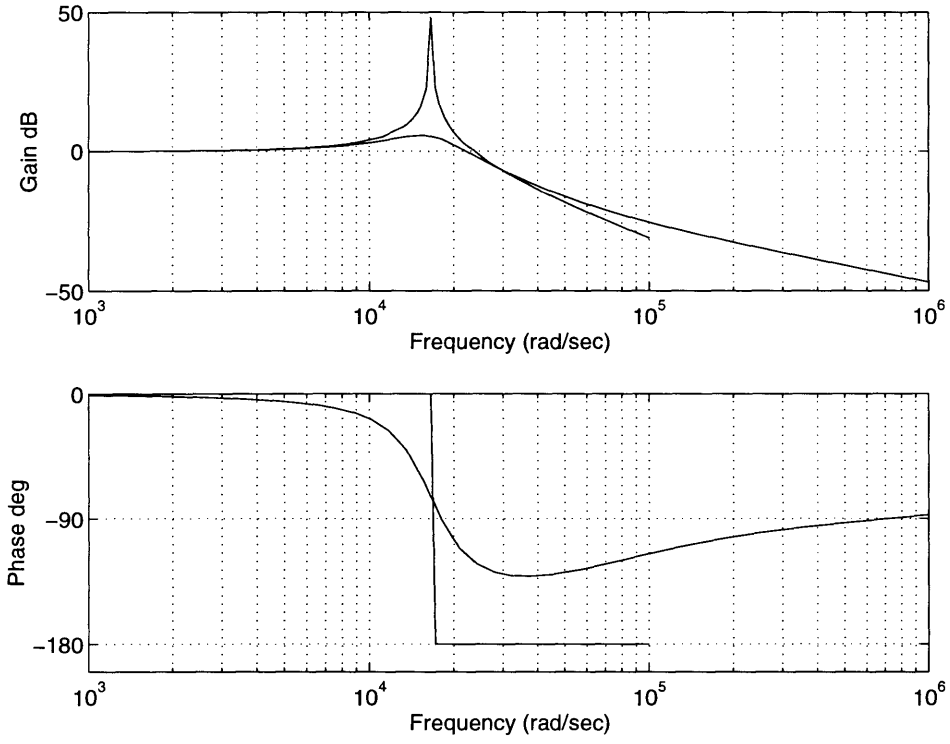


Figure 4-4: Comparasion between output filter models with and without parasitics included.

In general, $4L_{co}C_o$ can be neglected since it is much smaller than $(R_{co}C_o)^2$. Therefore the first zero occurs at approximately

$$\omega_{z1} = \frac{R_{co}}{L_{co}} \tag{4.15}$$

This is usually at a frequency high enough that it can be neglected. If L_{co} is small, the lower frequency zero is located at approximately

$$\omega_{z2} = \frac{1}{R_{co}C_o} \tag{4.16}$$

The poles of Eq. 4.13 are given by:

$$\omega_p = \frac{(-(R_{co} + R_l)C_o \pm \sqrt{(R_{co} + R_l)^2C_o^2 - 4LC_o)}}{2LC_o} \tag{4.17}$$

where we have assumed that $L + L_{co}$ is approximately equal to L. The real part of the two poles is located at

$$\mathcal{R}(\omega_p) = \frac{R_{co} + R_l}{2L} \tag{4.18}$$

If $(R_{co} + R_l)^2C_o^2 < 4LC_o$, the poles will have an imaginary component. As the quantity $R_{co} + R_l$ decreases, the magnitude of the imaginary component increases resulting in the phase contribution from each pole occuring at close to the same frequency. Referring again to Figure 4.4, we can see that for the simple LC filter model, with R_{co} and R_l both assumed to be zero, the 90° phase contribution from each pole occurred at the same frequency. By including both R_{co} and R_l (in this example, $R_{co} + R_l = 20$ mΩ) in the higher order model, the phase contribution from each pole is separated and the effective phase lag is more easily controlled. This is an important design parameter, indicating that a reduction in the ESR of the capacitor (R_{co}) and ESR of the inductor (R_l) can make compensation of the regulator far more difficult.

The location of the lower frequency zero, the imaginary component of the two poles, and the location of the poles on the real axis all require an accurate modeling

of the ESR of the inductor and capacitor. If the ESR of the inductor were to be neglected, the poles of the output filter would appear at a higher frequency and the filter would exhibit a lower Q than expected. Along with the previously mentioned problems, if the ESR of the capacitor is underestimated, the low frequency zero caused by this parasitic resistance would occur at a lower frequency than expected. Typical values of ESR and ESL values will be on the order of 10 mΩ. Varying their magnitudes by even a few milliohms can, in many cases, change the expected phase margin of the system by several degrees. Therefore, it is extremely important to not only include these values in the output filter model, but also to determine what percent variation they can have and still be able to guarantee system stability.

The R_{DSon} of the power FETs is another important parameter that needs to be modeled. When the top power FET is on (M1 in Figure 3.1), its channel impedance, or R_{DSon}, is in series with the filter inductor. Likewise, when the bottom power FET is on (M2 in Figure 3.1), its R_{DSon} is also in series with the inductor. This allows the R_{DSon} of the power FETs to be modeled as a series resistance. In the model developed above, the R_{DSon} can easily be included by redefining R_l as the quantity $(R_l + R_{DSon})$. In some regulator applications, two FETs are used in parallel for M1 and one is used for M2. In this situation, the effective value of the R_{DSon} is determined by the duty cycle. This can be accounted for in the model by using both the R_{DSon} of a single power FET and that of a parallel combination represented as $R_{DSon}/2$. This will allow the AC performance characteristics of the system transfer function to be verified for a power FET channel impedance in the range of $R_{DSon}/2$ to R_{DSon}.

4.2.2 Effect of Load Current Variation on Output Filter Model

From customer information, it was determined that the output load on the regulator could be modeled as a purely resistive element [15]. The MC33470 voltage regulator is required to provide from 0 to 14 amps with a regulated output range from 1.8 to 3.5 volts. This corresponds to an equivalent resistive load value of 0.13 Ω or greater.

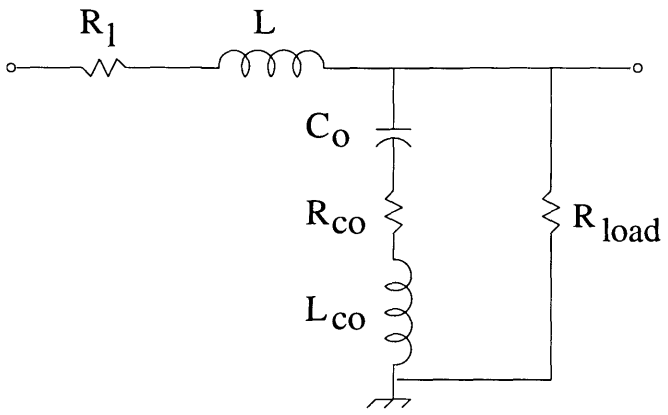


Figure 4-5: Model of the output filter including the effective load resistance.

The load can be modeled as shown in Figure 4.5 where a resistor is placed in parallel with the output filter capacitor. As the load current changes, the effective value of the resistive element changes causing a change in the location of the poles and zeros in the output filter transfer function.

The exact transfer function of the output filter including the load current can be derived as follows:

$$\frac{V_o}{V_i} = \frac{R_{load} \parallel (R_{co} + L_{co}s + \frac{1}{C_o s})}{(R_{load} \parallel (R_{co} + L_{co}s + \frac{1}{C_o s}) + R_l + Ls} \tag{4.19}$$

Therefore:

$$\frac{V_o}{V_i} = \frac{As^2 + Bs + C}{Ds^3 + Es^2 + Fs + G} \tag{4.20}$$

where

$$\begin{aligned} A &= R_{load}L_{co}C_o \\ B &= R_{load}R_{co}C_o \\ C &= R_{load} \\ D &= L_{co}LC_o \\ E &= R_{co}C_oL + R_{load}C_oL + R_lL_{co}C_o + R_{load}L_{co}C_o) \end{aligned}$$

$$F = (R_{load}R_{co}C_o + R_lR_{co}C_o + R_lR_{load}C_o + L)$$
$$G = R_{load} + R_l$$

This transfer function contains two zeros and three poles. In most cases, we can ignore the ESL of the capacitor which creates a high frequency pole/zero pair. The output filter transfer function will then reduce to:

$$\frac{V_o}{V_i} = \frac{R_{load}R_{co}C_os + R_{load}}{LC_o(R_{co} + R_{load})s^2 + (L + R_{co}R_{load}C_o + R_l(R_{co} + R_{load})C_o)s + R_{load} + R_l} \tag{4.21}$$

An example of the change in the output filter characteristic as a function of load current is shown in Figure 4.6 for the following component values.

C_o=1650 μF

L=2.2 μF

R_{co}=0.01 Ω

L_{co}=1 nH

R_l=0.01 Ω (which represents the R_{DSon} of the power FET and ESR of the inductor)

R_{load} varies from 0.13 Ω to infinity to represent a load transient from 0 to 14 amps. If the unity gain crossover frequency of the regulator loop is set at 3×10^4 rad/sec, a transition from maximum to minimum load current will change the phase margin of the system by almost 10°.

4.2.3 Development of Matlab Script to Determine Effect of Power Distribution Network and Load Current Variation

So far the output filter model has neglected the effects of the power distribution network, including both board parasitics as well as the microprocessor bypass capacitors with their associated parasitic elements. The output filter and power distribution network is shown in Figure 4.7. Deriving the transfer function to relate V_o to V_i by hand is tedious, and can be simplified using Matlab. An expression for V_o/V_i can be

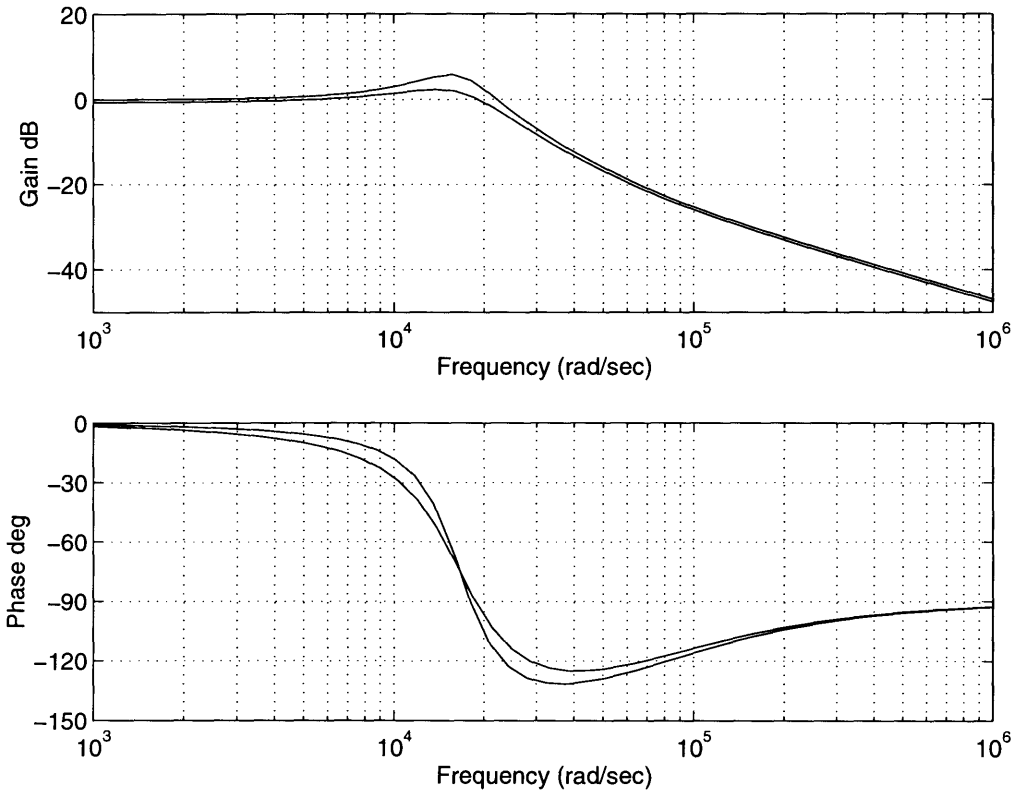


Figure 4-6: Example of output filter characteristic change due to changing load current.

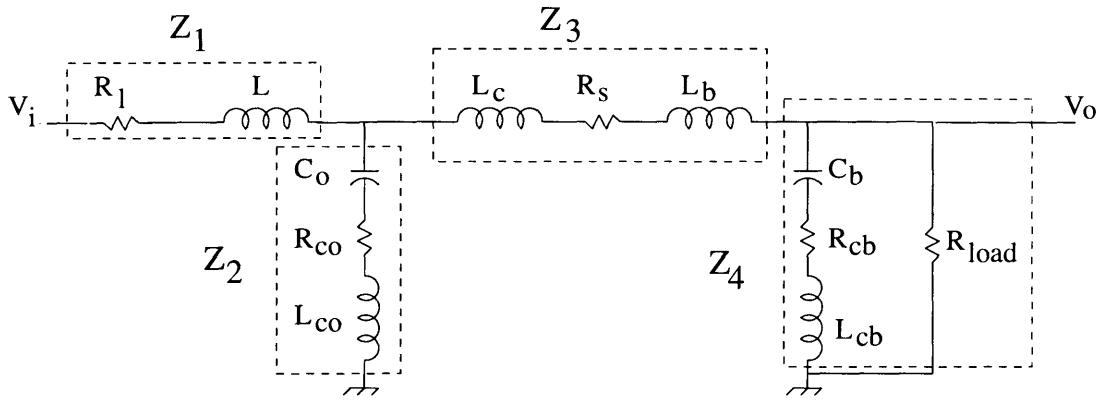


Figure 4-7: Model of the power distribution network used to find its transfer function.

found using a combination of basic voltage divider relationships. These relationships can be defined as follows:

$$S(s) = \frac{V_o}{V_f} = \frac{Z_4}{Z_3 + Z_4} \tag{4.22}$$

$$T(s) = \frac{V_f}{V_i} = \frac{Z_2 \parallel (Z_3 + Z_4)}{Z_1 + Z_2 \parallel (Z_3 + Z_4)} \tag{4.23}$$

therefore,

$$\frac{V_o}{V_i} = \frac{Z_2 \parallel (Z_3 + Z_4)}{(Z_1 + Z_2 \parallel (Z_3 + Z_4)} \cdot \frac{Z_4}{Z_3 + Z_4} \tag{4.24}$$

where

$$Z_1 = R_l + Ls \tag{4.25}$$

$$Z_2 = \frac{1}{C_o s} + L_{co}s + R_{co}s \tag{4.26}$$

$$Z_3 = R_s + (L_c + L_b)s \tag{4.27}$$

and

$$Z_4 = \frac{R_{load}L_{cb}C_b s^2 + R_{cb}C_b R_{load}s + R_{load}}{L_{cb}C_b s^2 + (C_b R_{load} + R_{cb}C_b)s + 1} \tag{4.28}$$

First, the impedances Z_1 through Z_4 are defined in Matlab. The numerator and denominator of Z_2 and Z_4 are defined separately to aid in the simplification of Eq. 4.24.

```
Z1=[l rdson];
```

```
Z2n=[lco*co rco*co 1];
Z2d=[co 0];
Z3=[(lb+lc) rs];
Z4n=[lcb*cb*rload rcb*cb*rload rload];
Z4d=[cb*lcb (rcb*cb+rload*cb) 1];
```

Eq. 4.22 above is defined in Matlab as S(s) where s_1 is the numerator and s_2 is the denominator.

```
s1=[rload*lcb*cb rload*rcb*cb rload];
s2=[(lb+lc)*lcb*cb (rs*lcb*cb+rload*lcb*cb+
(lb+lc)*rcb*cb+(lc+lb)*rload*cb)
(rs*rcb*cb+rload*rcb*cb+rs*rload*cb+(lc+lb)) rload+rs];
```

Using the chain rule relationship, $V_o/V_i = (V_f/V_i)(V_o/V_f) = S(s)T(s)$, once T(s) is found, a complete relationship for V_o/V_i can be determined numerically with the Matlab series command. Using the definitions for Z_1 through Z_4 above, the equation for T(s) can be expressed as follows:

$$T(s) = \frac{(Z_3 + Z_4)Z_2}{(Z_3 + Z_4)Z_2 + Z_1(Z_2 + Z_3 + Z_4)} \tag{4.29}$$

Expressing Z_2 and Z_4 in terms of their numerators and denominators:

$$T(s) = \frac{(Z_3 Z_{4d} + Z_{4n})Z_{2n}}{(Z_3 Z_{4d} + Z_{4n})Z_{2n} + Z_1(Z_3 Z_{4d} Z_{2d} + Z_{4n} Z_{2d} + Z_{4n} Z_{4d})} \tag{4.30}$$

S(s) and T(s) have been expressed in this form so that it is easier to find a numerical expression in Matlab. Problems can occur using the series and parallel commands to find each part of V_o/V_i individually because the order of the numerator can be higher than the order of the denominator at intermediate stages in the calculation. A high frequency pole could be used as before to avoid these problems. However, the power distribution network already adds several high frequency poles and zeros and any added to ease calculations will degrade the accuracy of the model.

The parallel combination of two transfer functions of the form $G(s) = g_1/g_2$ and $H(s) = h_1/h_2$ is given by

$$G(s) + H(s) = \frac{h_1 g_2 + h_2 g_1}{g_2 h_2} \tag{4.31}$$

This technique is utilized to develop an expression for the transfer function of the output filter in parallel with the power distribution network. The steps necessary to complete the analysis are outlined below.

```
% determine numerator coefficients of T(s)
% num1 represents the quantity z3z4d+z4n
[num1,den1]=parallel(Z3,Z4n,[1],Z4d);
out1=conv(Z2n,num1);
% determine denominator of T(s)
% num2 represents the quantity z2d(z3z4d+z4n)+z2nz4d
temp1=out1;
[num2,den2]=parallel(Z2d,Z2n,Z4d,num1);
temp2=conv(num2,Z1);
```

The denominator of T(s) is actually given as the sum of the vectors `temp1` and `temp2`. However, they cannot be summed directly because the two vectors are of different lengths. To correct this problem, the length of the vector `temp1` is increased, and the two vectors are summed to form the denominator of T(s). The complete expression for V_o/V_i is found by taking the series combination of T(s) and S(s) as shown below.

```
temp3=[0 temp1];
out2=temp2+temp3;
[t1,t2]=series(out1,out2,s1,s2);
```

To verify the validity of the Matlab model derived above, the output filter and power distribution network was constructed and simulated in PSpice. A comparison of the frequency characteristics of the two models is shown in Figure 4.8. The component values are the same as used in Section 4.1.3, except $R_{co} = 2.6$ mΩ.

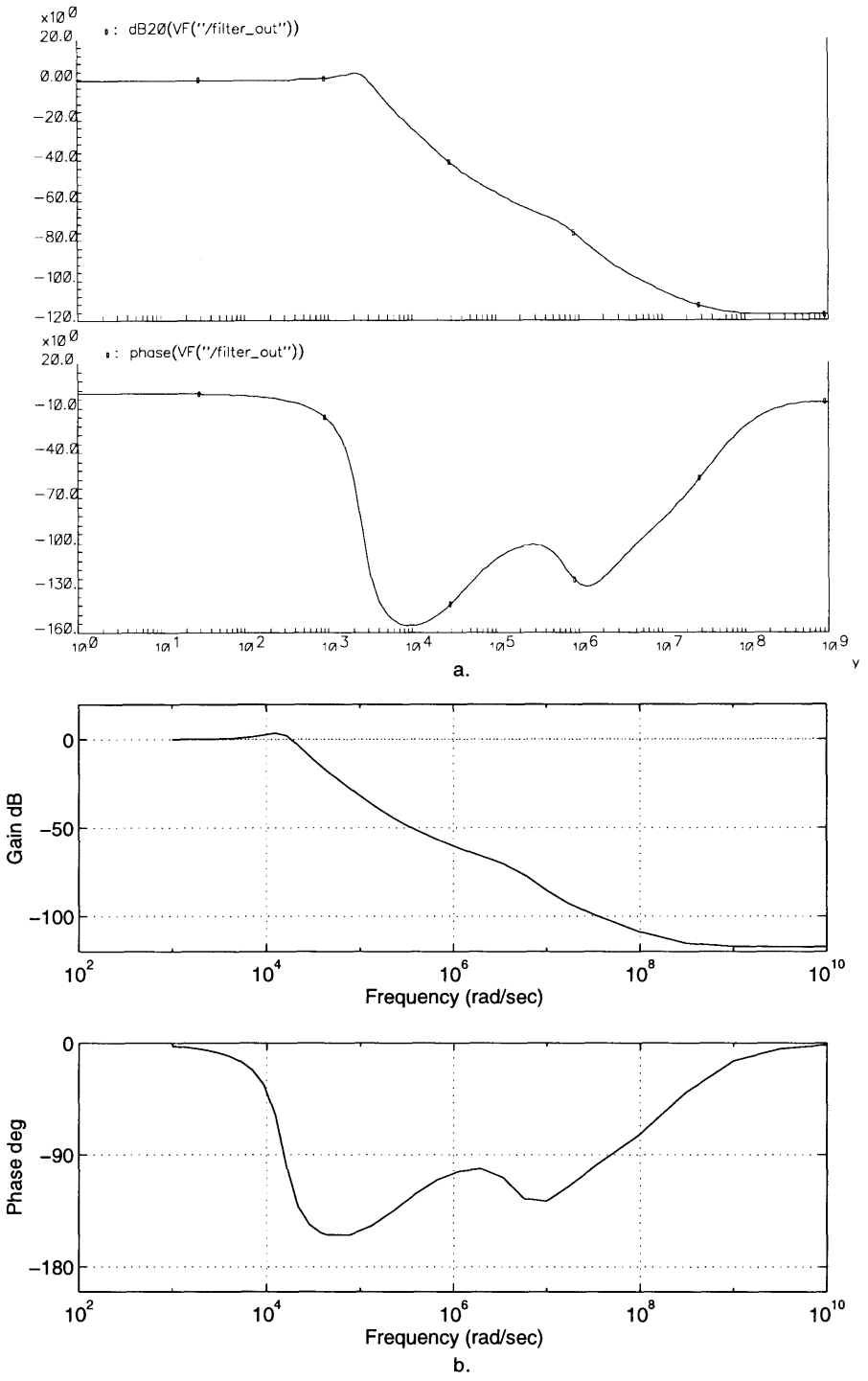


Figure 4-8: Using a) PSpice to verify the b) Matlab model of the power distribution network.

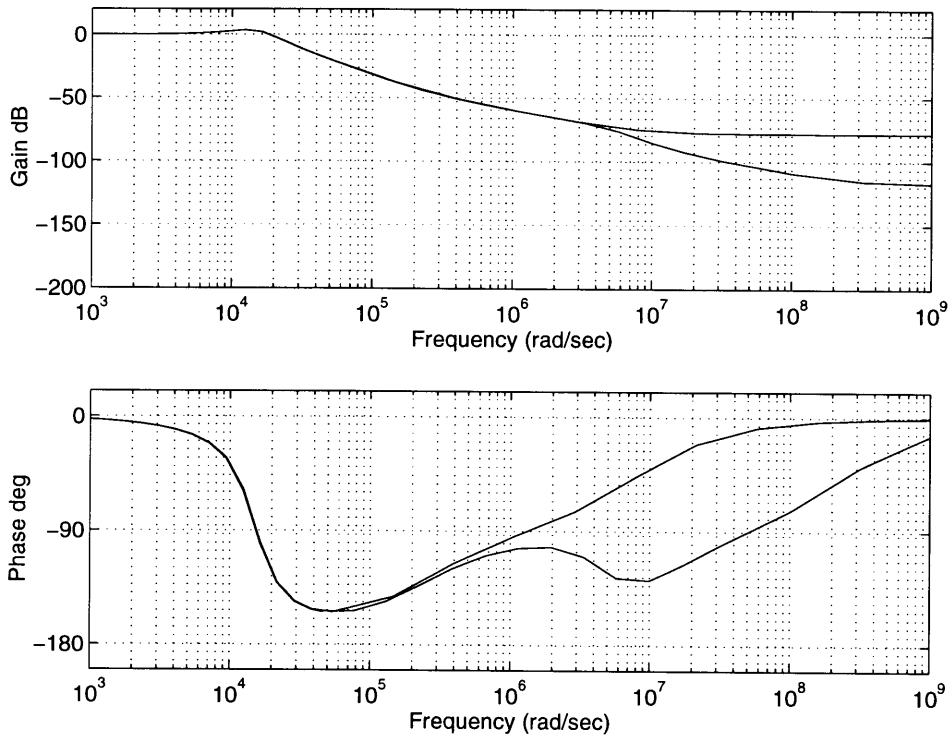


Figure 4-9: Comparison of output filter model with and without the power distribution network model.

The frequency response of the output filter, with and without the inclusion of the power distribution network, is illustrated in Figure 4.9. At low enough frequencies, the characteristics are reasonably well matched. However, as the loop bandwidth approaches 10^5 rad/sec, the effect due to the high frequency poles and zeros of the power distribution network become increasingly obvious. Therefore, the power distribution network can only be neglected if the loop bandwidth is maintained well below the frequency at which the contribution from these high frequency poles and zeros can be felt.

The following section of script prompts the user to enter the load transient requirements for their particular application.

```
min=input([blanks(5) 'Enter minimun load (in Amps): -->  ']);
if min==0,
min=1e-6;
end
```

```
max=input([blanks(5) 'Enter maximum load (in Amps): -->  ']);
minv=input([blanks(5) 'Enter minimun output voltage: -->  ']);
maxv=input([blanks(5) 'Enter maximum output voltage: -->  ']);
rload1=maxv/min;
rload2=minv/max;
```

The user can now verify that the system stability criteria is met for minimum and maximum load conditions.

4.3 Error Amplifier and Compensation

The basic model for the error amplifier with compensation network was developed in Section 3.2.1 for both an OTA and operational amplifier. From the values provided by the user for both the error amplifier and compensation network, the Matlab model and associated script would generate the corresponding frequency characteristic. The error amplifier compensation has a large impact on the phase margin of the regulator. If the output filter components are modified to alter the output voltage transient response or decrease the system cost, the compensation components will also need to change to preserve the desired phase margin. This can be a very time consuming process for the designer to manually recalculate the compensation values each time the output filter values change. Therefore, the Matlab algorithm needs to incorporate a routine to automatically calculate the necessary compensation values given the error amplifier characteristics and output filter components. This algorithm should offer the user the option of several types on compensation schemes, including single pole, single zero or double pole, double zero methods. Also, it should allow the user to manually enter compensation component values so that the effect of changing other system parameters can be observed.

4.3.1 OTA and Operational Amplifier and Compensation Modeling

In Section 3.2.1, the operational amplifier model was found to be governed by the transfer function in Eq. 3.13 which is repeated below.

$$A(s) = \frac{A}{(\frac{1}{f_p}s + 1)}$$

A is the low frequency open loop gain and f_p is the low frequency pole location. The model can be made more complete by accounting for the high frequency, non-dominant pole so that the new operational amplifier transfer function becomes:

$$A(s) = \frac{A}{(\frac{1}{f_{p1}}s + 1)(\frac{1}{f_{p2}}s + 1)} \tag{4.32}$$

The amplifier transfer characteristic with a type II compensation scheme [35] was derived in Section 3.2.1. However, other compensation techniques are available, the simplest type of which is the low frequency single pole approach. This method is referred to as type I [35] and can be implemented as shown in Figure 4.10a. For this method,

$$Z_f = \frac{1}{Cs} \tag{4.33}$$

and

$$Z_i = R \tag{4.34}$$

If we assume an ideal amplifier frequency response, the transfer function for the error amplifier with compensation reduces to Z_f/Z_i or

$$H_1(s) = \frac{1}{RCs} \tag{4.35}$$

The Bode diagram of this transfer function is shown in Figure 4.10b. This method requires the fewest components and is therefore very cost effective provided it is useable for a particular regulator application. However, it has the disadvantage of limiting

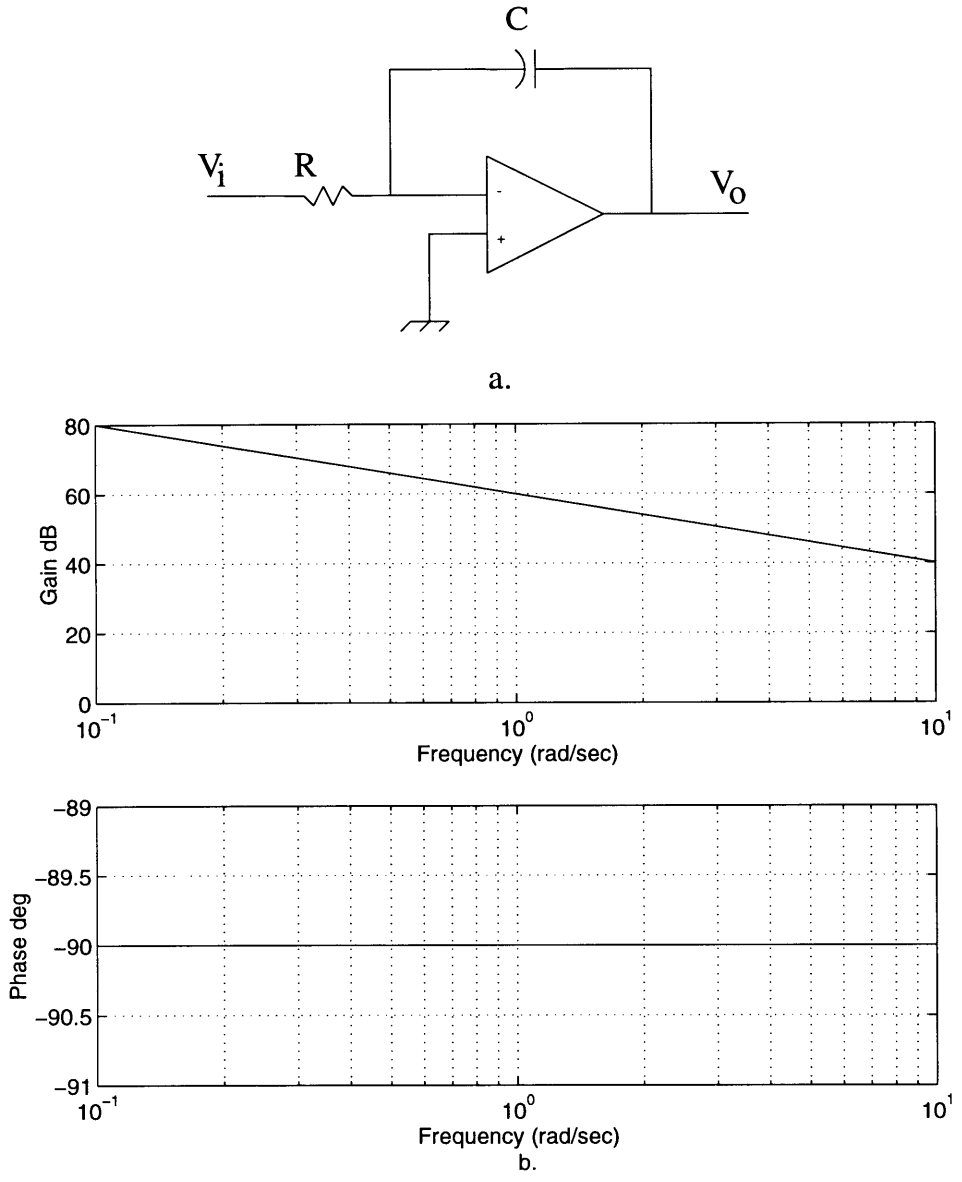


Figure 4-10: a) Block diagram and b) typical Bode plot of a type I compensation scheme.

the regulator loop bandwidth to a very low value. The single pole approach is used to reduce the loop gain to less than one at a frequency below the two poles contributed by the output filter. For regulators used in microprocessor applications, this double pole combination usually occurs around 10 kHz. Using a type I compensation scheme would require that the loop bandwidth be less than 10 kHz, which is usually too low when the switching frequency of the regulator is approximately 200 to 300 kHz. Therefore, type I will not be considered further in this thesis because it has little application to this type of regulator program.

The architecture of a type II [35] compensation method is shown in Figure 4.11a. In Section 3.2.1, the feedback network relationship was found in Eq. 3.10 and is repeated below.

$$Z_f = \frac{(R_f C_{f1} s + 1)}{s(R_f C_{f1} C_{f2} s + C_{f1} + C_{f2})}$$

with $Z_i = R_i$. If we assume an ideal amplifier characteristic, the transfer function of the amplifier and compensation is given as:

$$H_2(s) = \frac{Z_f}{Z_i} = \frac{(R_f C_f 1 s + 1)}{s(R_f C_{f1} C_{f2} s + C_{f1} + C_{f2}) R_i} \tag{4.36}$$

This transfer characteristic, illustrated in Figure 4.11b, contains a single pole at DC, a second high frequency pole, and a zero. The placement of this zero is set relative to the second pole to achieve a phase lead characteristic dependent on the spacing of the doublet. Up to 90° of phase addition is possible if the zero and high frequency pole are widely separated. The maximum phase lead is positioned such that the negative phase contributed by the double pole from the output filter is partially cancelled by the compensation network. This allows the cross over frequency of the regulator to be somewhat higher than the double pole frequency, unlike with type I compensation.

As discussed in Section 4.2.1, if the ESR of the output filter capacitor is approaches zero ohms, the negative phase contributed from the filter may approach −180°. This is difficult to compensate with a type II compensation technique and still preserve an acceptable phase margin at the desired loop bandwidth. In this situation, a type III

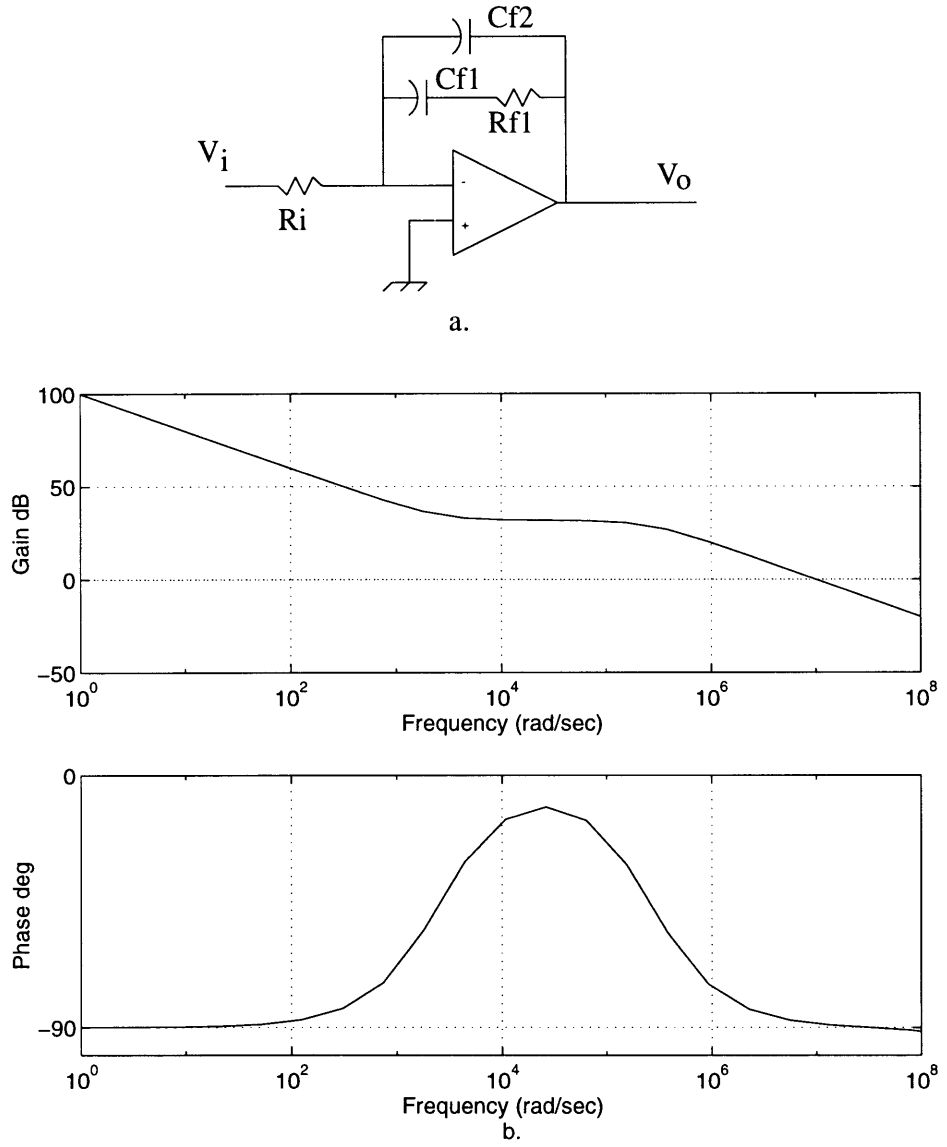


Figure 4-11: a) Block diagram and b) typical Bode plot of a type II compensation scheme.

[35] method can be used. The type III architecture is shown in Figure 4.12a with Z_f and Z_i as follows:

$$Z_f = \frac{R_f C_{f1} s + 1}{s(R_f C_{f1} C_{f2} s + C_{f1} + C_{f2})} \tag{4.37}$$

$$Z_i = \frac{R_{i2} R_{i1} C_1 s + R_{i1}}{(R_{i1} + R_{i2}) C_i s + 1} \tag{4.38}$$

With an ideal amplifier, the transfer function of the operational amplifier and compensation network, illustrated in Figure 4.12b, is given as shown in Eq. 4.39.

$$H_3(s) = \frac{Z_f}{Z_i} = (R_f C_{f1} s + 1) \frac{((R_{i1} + R_{i2}) C_i s + 1)}{s(R_f C_{f1} C_{f2} s + C_{f1} + C_{f2})(R_{i2} R_{i1} C_1 s + R_{i1})} \tag{4.39}$$

This characteristic has a single pole at DC and, depending on the selection of the component values, can contain two high frequency pole - zero pairs. This type of compensation method can therefore contribute up to 180° of positive phase, allowing it to compensate a much larger phase lag from the output filter components. However, it has the disadvantage of requiring many more external compensation components than either a type I or type II method and therefore would only be used when absolutely necessary or cost and simplicity were not important issues.

The final type of compensation that will be considered is a series RC network, as illustrated in Figure 4.13a with

$$Z_i = R_i \tag{4.40}$$

and

$$Z_f = \frac{R_f C_f s + 1}{C_f s} \tag{4.41}$$

The transfer function of the amplifier and compensation network, assuming ideal amplifier characteristics, is given as shown in Eq. 4.42.

$$H_4(s) = \frac{Z_f}{Z_i} = \frac{R_f C_f s + 1}{C_f R_i s} \tag{4.42}$$

The Bode diagram of this transfer function is shown in Figure 4.13b. It has a single, low frequency pole and a high frequency zero. This results in a 90° phase

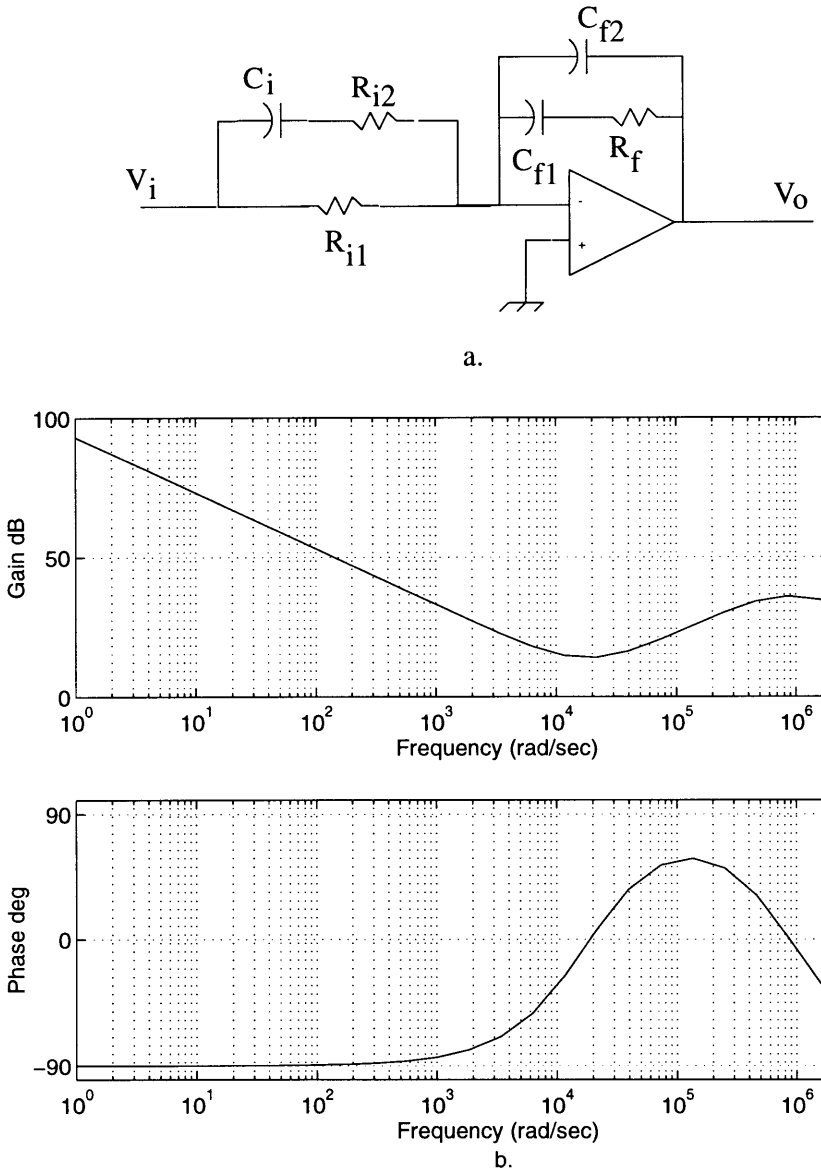


Figure 4-12: a) Block diagram and b) typical Bode plot of a type III compensation scheme.

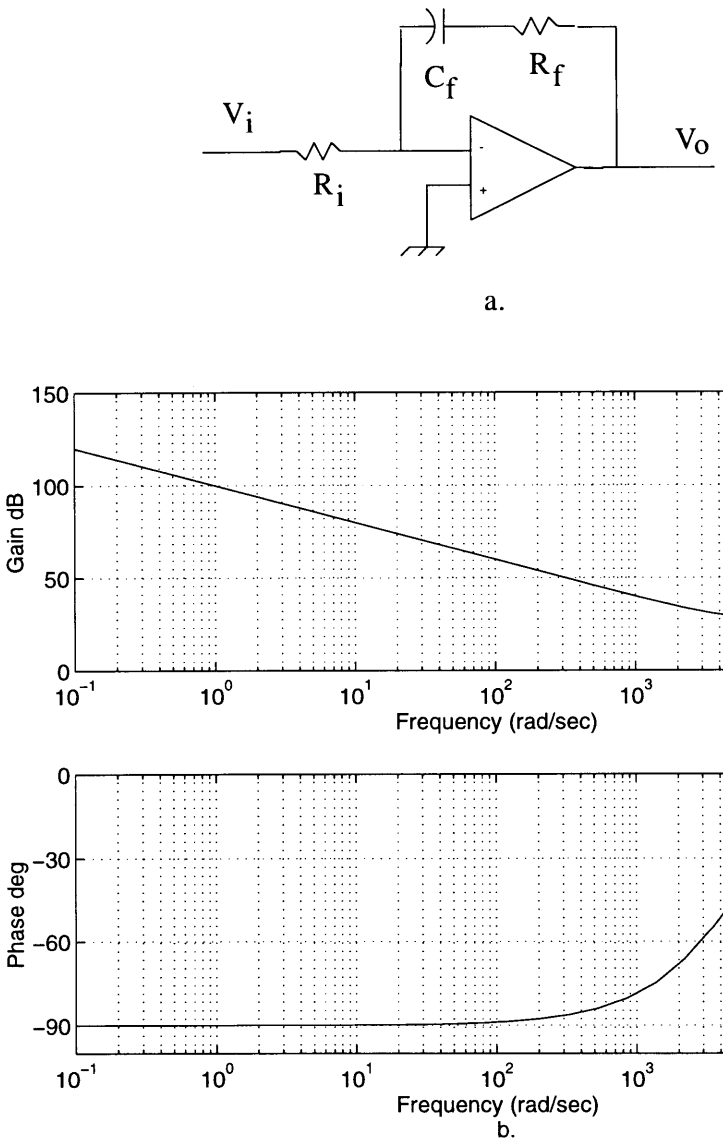


Figure 4-13: a) Block diagram and b) typical Bode plot of a series RC compensation scheme.

lead characteristic above the pole frequency. This method is similar to the type II technique, with the exception that the series RC method does not have the second high frequency pole. The advantage of this approach is that is has less components, and is therefore less costly to the customer, than a type II, while still achieving basically the same functionality. The disadvantage is that without the second high frequency pole, the loop gain at high frequencies is not reduced fast enough, leaving the system more suseptible to noise.

A simplified model was derived for an OTA in Section 3.2.1 in Eq. 3.18 and is summarized below:

$$G_m(s) = \frac{I_{out}}{\Delta v_{in}} = \frac{g_m}{\frac{1}{\omega_p}s + 1}$$

where g_m is the low frequency transconductance of the amplifier and wp is the location of the dominant pole. The non-dominant poles of an OTA are generally at a much higher frequency than those associated with an operational amplifier. These high frequency poles will therefore be neglected to preserve the simplicity of the OTA model.

Compensation methods for the OTA resemble those used with an operational amplifier. The first method that will be considered is the single pole, or type I architecture as shown in Figure 4.14. The transfer function of the OTA and compensation is given by

$$H(s) = G_m(s)Z_c(s) \tag{4.43}$$

where $Z_c(s)$ is the output impedance of the OTA in parallel with the compensation network. For a type I compensation method,

$$Z_c(s) = \frac{R_{out}}{R_{out}C_{c1}s + 1} \tag{4.44}$$

This compensation method has the same limitations as the type I described for an operational amplifier. To achieve the desired phase margin of the system, the pole due to the compensation needs to be added at a frequency lower than that of the

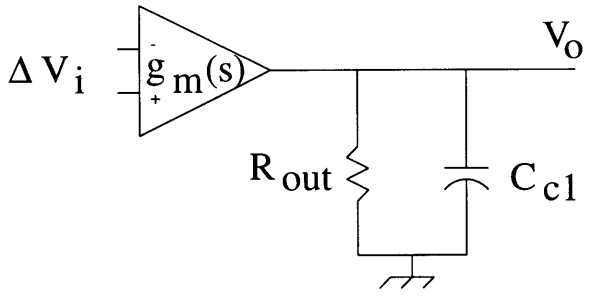


Figure 4-14: Block diagram of a type I compensation scheme using an OTA.

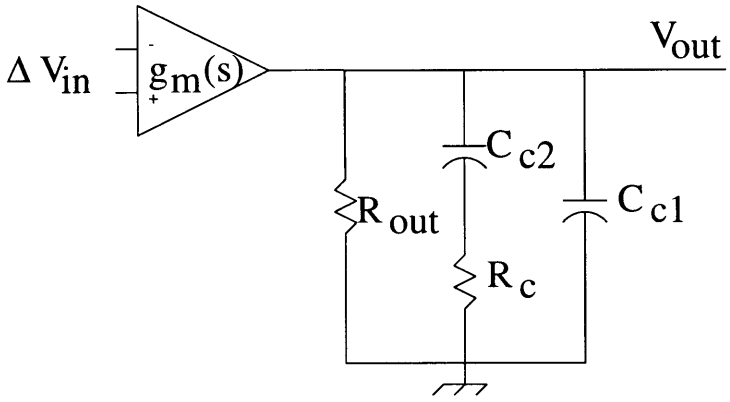


Figure 4-15: Block diagram of a type II compensation scheme using an OTA.

output filter double pole. In many situations, this restricts the loop bandwidth to a value too low to handle a modern microprocessor's rapid load current transients.

A type II compensation scheme with an OTA is shown in Figure 4.15. This method was used as an example in Section 3.2.1 and Eq. 3.20, and the results are summarized below:

$$Z_c = \frac{sC_{c2}R_cR_{out} + Rout}{s^2C_{c1}C_{c2}R_{out}R_c + s(R_{out}C_{c2} + R_{out}C_{c1} + C_{c2}R_c) + 1} \tag{4.45}$$

This compensation has a zero and 2 poles arranged such that a phase lead of up to 90° is created. However, the range of frequency over which the phase lead occurs

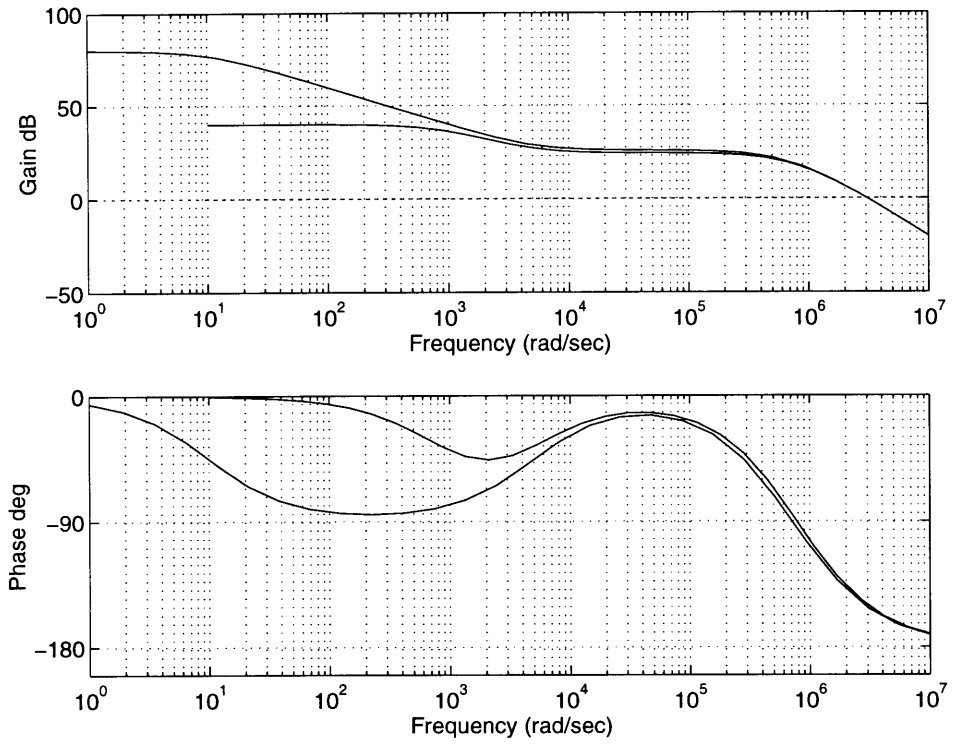


Figure 4-16: Bode plot for type II compensation with $R_{out} = 100$ kΩ and 10 MΩ.

is dependent of the output impedance of the OTA. For example, using the values from Section 3.2.1 and varying R_{out} between 100 kΩ and 10 MΩ, the two OTA and compensation transfer function extremes are shown in Figure 4.16. When an operational amplifier is used, the feedback path around the amplifier tends to minimize the effects of the frequency characteristics of the amplifier on the compensation. Because the compensation of an OTA does not create a feedback loop around the OTA, the transfer function of the OTA and compensation is more suspectible to variations in OTA characteristics. Therefore, if a type II compensation method is used, the possible variation in output impedance has to be considered to ensure the system stability is acceptable over the entire output impedance range.

Series RC compensation of an OTA is shown in Figure 4.17. Z_c is given as:

$$Z_c = \frac{R_c R_{out} C_{c1} s + R_{out}}{(R_{out} + R_c) C_{c1} s + 1} \tag{4.46}$$

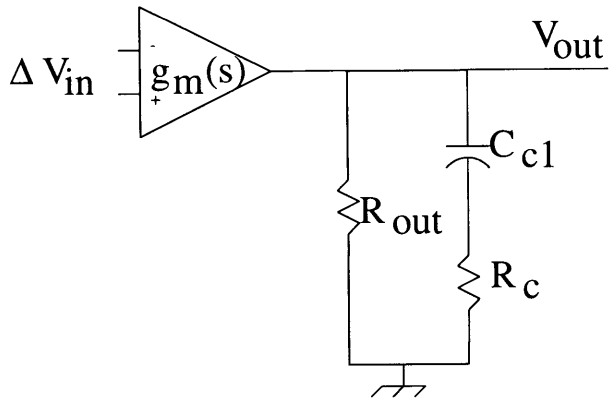


Figure 4-17: Series RC compensation using with an OTA.

This compensation technique creates a zero at $1/C_{c1}R_c$ and a pole at $1/(R_{out}+R_c)C_{c1}$. As with the type II compensation of an OTA, the series RC pole location is dependent on the output impedance of the OTA. This compensation method has the same disadvantages and advantages with an OTA as it does when used with an operational amplifier. While it achieves approximately the same results using fewer external components, it is much more susceptable to noise than the type II approach.

4.3.2 Development of Matlab Script to Find Compensation Network

During the regulator design process, the designer may need to change the error amplifier compensation technique and component values multiple times. Because of this, the design cycle time can be reduced by including an algorithm in the Matlab script to calculate compensation values automatically for each type of possible compensation architectures. The script should allow the user to manually enter the desired compensation values rather than calculating them automatically so that the effects of changing error amplifier parameters on the system stability for a given set of compensation values can be observed. The Matlab script has been developed with these considerations in mind.

In reference [35], a method for calculating compensation components based on

the K factor is presented. In this method, K is a measure of the distance between the poles and zeros in the compensation technique and therefore it determines how much phase boost is provided. This method has been incorporated into Matlab for use with an operational amplifier and type II and III compensation techniques. The implementation of this methodology is outlined below.

First, the desired loop bandwidth is defined as ω_c. This value could be either user defined or found from the power distribution network calculations. The phase of the output filter is then found at this frequency and is defined as the parameter p.

```
p=angle(out1(1)*(wc*i)+out1(2))-angle(out2(1)*(wc*i)^2
+out2(2)*(i*wc)+out2(3));
```

The phase is converted from radians to degrees. The desired phase margin is defined as m. These two values will then allow the required phase boost contribution from the compensation to be calculated.

```
p=p*180/pi;
m=pm;
boost=m-p-90;
```

If the amount of phase addition is greater than 90°, K is set to its maximum value of 10 since type II compensation can not provide a phase boost of greater than 90°.

```
if boost>90,
     k=10;
end
```

K is then calculated using the relationship defined in reference [35]. Again, K is limited to a value of 10 to restrict the spacing of the pole-zero pair.

```
k=tan((boost/2+45)*pi/180);
if k>10,
     k=10;
end
```

The variable g is defined to be the inverse of the loop gain at ω_c, the loop crossover frequency. By requiring the transfer function of the error amplifier and compensation to be g at ω_c, the loop gain is set to be 1 at ω_c.

```
g=1/abs((out1(1)*(wc*i)+out1(2))/(out2(1)*(wc*i)^2
+out2(2)*(i*wc)+out2(3)));
```

R_1 is assumed to be 1 kΩ, and the other compensation values are found using the equations from the reference. If a different value of R_1 is desired, all other values can be scaled appropriately to maintain the location of the poles and zeros.

```
r1=1e3;
c1=1/(wc*g*k*r1);
c2=c1*(k\^2-1);
r2=k/(wc*c2);
```

Type III compensation values are calculated in a manner similar to that for a type II approach with the primary difference being that the maximum phase addition is limited to 180°. This also results in a larger allowable value for the K factor. The Matlab script is shown below.

```
p=angle(out1(1)*(wc*i)+out1(2))-angle(out2(1)*(wc*i)^2
+out2(2)*(i*wc)+out2(3));
p=p*180/pi;
m=pm;
boost=m-p-90;
if boost>180,
     k=100;
end
k=(tan((boost/4+45)*pi/180))^2;
if k>100,
     k=100;
end
```

```
g=1/abs((out1(1)*(wc*i)+out1(2))/(out2(1)*(wc*i)^2
+out2(2)*(i*wc)+out2(3)));
r1=1e3;
c1=1/(wc*g*r1);
c2=c1*(k-1);
r2=k^0.5/(wc*c2);
r3=r1/(k-1);
c3=1/(wc*k^0.5*r3);
```

Although the series RC compensation method is not defined in reference [35], the K value technique can be used to find compensation components by adapting the method shown for the type II approach. This is illustrated below.

```
p=angle(out1(1)*(wc*i)+out1(2))-angle(out2(1)*(wc*i)^2
+out2(2)*(i*wc)+out2(3));
p=p*180/pi;
m=pm;
boost=m-p-90;
if boost>90,
     boost=90;
end
g=1/abs((out1(1)*(wc*i)+out1(2))/(out2(1)*(wc*i)^2
+out2(2)*(i*wc)+out2(3)));
r1=1e3;
x=-tan((-boost)*pi/180)/wc;
r2=g*r1;
c1=x/r2;
```

When an OTA is used in place of an operational amplifier, the compensation components are modeled in parallel with the output impedance of the amplifier. This causes the transfer functions to be more complicated than those used in the K factor method. Because of this, the compensation values are found and then adjusted

iteratively until the required phase margin and loop bandwidth are achieved. A variable `gnuf` is initalized to zero to indicate that the compensation values will not meet the required performance requirements. A variable `tries` is also defined to limit the number of iterations that will be performed. The following script was developed for type II compensation using an OTA.

```
gnuf=0;
tries=0;
```

An initial attempt at determining appropriate compensatin values is performed by setting $R_1 = 20$ kΩ in Figure 4.15 and placing the zero to coincide with the lowest frequency pole in the output filter. C_2 is set to $C_1/100$ so that the phase boost is approximately 80°. Increasing the value of C_2 results in less positive phase addition. The phase margin and crossover frequency of the loop is determined using these values, and the variable `tries` is then incremented. If the phase margin is not adequate, the location of the zero due to the compensation is moved to a slightly higher frequency. The steps are repeated until either an acceptable phase margin is achieved or the variable `tries` reaches its maximum value.

```
while gnuf==0,
     r1=20e3;
     c1=1/r1/(abs(min(real(roots(out2))))+tries*100);
     c2=c1/100;
     tries=tries+1;
     comp1=[c1*r1*rl rl];
     comp2=[c1*c2*r1*rl (rl*c1+rl*c2+c1*r1) 1];
     [eamp1,eamp2]=series(ota1,ota2,comp1,comp2);
     [tot1,tot2]=series(eamp1,eamp2,out1,out2);
     cresults(loopc,1)=r1;
     cresults(loopc,3)=c1;
     cresults(loopc,4)=c2;
```

```
[temp1,cresults(loopc,5),temp2,cresults(loopc,6)]=margin(tot1,tot2);
cresults(loopc,6)=cresults(loopc,6)/2/pi;

     if (cresults(loopc,5)>30 | tries>100),
          gnuf=1;
      end
end                                % end of while loop
```

The compensation method for a series RC configuration is similar to that used for the OTA with type II compensation and is included below.

```
gnuf=0;
tries=0;
while gnuf==0,
     r1=20e3;
     c1=1/r1/(abs(min(real(roots(out2))))+tries*100);
     tries=tries+1;
     comp1=[c1*r1*rl rl];
     comp2=[c1*(r1+rl)  1];
     [eamp1,eamp2]=series(ota1,ota2,comp1,comp2);
     [tot1,tot2]=series(eamp1,eamp2,out1,out2);
     cresults(loopc,1)=r1;
     cresults(loopc,3)=c1;
[temp1,cresults(loopc,5),temp2,cresults(loopc,6)]=margin(tot1,tot2);
cresults(loopc,6)=cresults(loopc,6)/2/pi;

     if (cresults(loopc,5)>minpm | tries>100),
          gnuf=1;
     end
end                                % end of while loop
```

This section outlines the use of a Matlab script to implement a method for automatically determining compensation values for a switching regulator. This methodology can also be used to assist in the design of the error amplifier by providing specifications and limits for key performance parameters.

4.3.3 Determination of Amplifier Characteristics

The characteristics of the error amplifier are important in determining the stability performance of a switching regulator. For example, variations in the open loop gain of an OTA can change the loop crossover frequency. Also, variations in the output impedance of the OTA can change the compensation frequency characteristics, which can change the phase margin of the system. Continuing with the methodology described thus far, an algorithm is developed to vary the amplifier characteristics to determine how susceptible the system is to parameter variation. Once acceptable boundaries have been determined for the amplifier charateristics, the amplifier can be designed to the required specifications. This becomes especially critical when designing a switching regulator IC where the performance requirements can often be limited to system specifications only, with no information regarding subsystem performance is available to the designer.

Once the script has suggested an output filter capacitor combination and the user has made the final decision on which capacitors to use, Matlab uses the regulator data to iteratively determine how much the error amplifier parameters can vary before the system performance is degraded to an unacceptable level. First, the phase margin and loop bandwidth achieved with the nominal system parameters is saved to the variables `typpm` and `typbw`, respectively. The variable `ctc` represents the row number of the capacitor selected by the user.

```
typpm=cresults(ctc,5);
typbw=cresults(ctc,6);
```

Next, a while loop is used to create a multiplication factor, `n1`, that is incremented

by one each time the loop is repeated. The loop is interrupted when the loop factor reaches 11 or the amplifier characteristics fail to meet certain specifications. The latter condition results in the variable `brokeyet` being set to 1.

```
brokeyet=0;
n1=1;
while (brokeyet==0 & n1<11),
n1=n1+1;
```

The multiplication factor `n1` is used to vary the DC gain and the location of the two poles of the operational amplifier. The gain and pole locations are alternately multiplied and divided by `n1` to take into account each possible combination of operational amplifier characteristic variation. The first case is shown below.

```
g1=[av*n1];
g2=[1/p1*n1 1];
h1=[1];
h2=[1/p2*n1 1];
```

The new system phase margin and bandwidth are then determined.

```
[vampchar1,vampchar2]=series(g1,g2,h1,h2);
[eamp1,eamp2]=feedback(vampchar1,vampchar2,comp1,comp2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);
[j1,varypm,j2,varywc]=margin(tot1,tot2);
varywc=varywc/2/pi;
```

The variable `totvary` is incremented during each iteration of loop if the following requirements are satisfied. First, the regulator bandwidth must be at least half of its nominal value. Secondly, the system phase margin must be greater than the minimum acceptable value. The user is also allowed to select a phase margin and capacitor type regardless of those selected by the script. In that case, the default values in the program are not used.

```
totvary=0;
if (varypm>minpm | varypm>typpm & varywc>typbw*0.5),
     totvary=totvary+1;
end
```

The susceptability of the system to parameter variations is monitored by observing how close the phase gets to $-180°$ before the loop crossover frequency is reached. A decrease in the ESR of the capacitor causes negative phase to be added to the system. The system could become unstable if a small phase margin had already existed. The most negative phase prior to the crossover frequency is saved to the variable `rbstv`. The variable `minpml` is defined to be either $10°$ or the minimum phase margin, whichever is smaller. If `rbstv+180` is greater than `minpml`, the regulator performance is deemed to be acceptable with the given variation in error amplifier characteristics and the variable `totvary` is incremented.

```
[j1,j2,j3,wca]=margin(tot1,tot2);
[mag2,phase2,w2]=bode(tot1,tot2);
w3=find(w2<wca);
phase2=phase2(1:length(w3));
rbstv=min(phase2);
if (rbstv+180)>minpml,
     totvary=totvary+1;
end
```

The two tests outlined above are performed for each of the four variations in error amplifier parameters. If system performance requirements for each iteration through the loop are met, the variable `totvary` will have a value of eight. Therefore, if `totvary` is less than eight, the system failed at least one test and the while loop is terminated.

```
if totvary<8,
     brokeyet=1;
end
```

```
end                    %end of while loop
```

The user is informed of the maximum factor by which the operational amplifier parameters can vary without degrading the regulator's performance below a minimum acceptable level.

```
disp(' ')
disp([blanks(5) ' The open loop gain and pole location of the
operational amplifier can vary by the'])
disp([blanks(5) ' following factor without becoming unstable.'])
opampvar=n1-1
```

A similar approach was developed for an OTA. However, because an OTA is used in an open loop configuration, the typical amount of parameter variation is much less than that allowed with an operational amplifier. Therefore, rather than the variable `n1` increasing by 1 for each iteration of the while loop, it is increased by 0.1 up to a limit of 2.1.

4.4 Input Filter Modeling

In most microprocessor applications, the input supply voltage to the regulator can also be used to power other subsystems. The operation of the switching regulator produces a large ripple component in the input current which can create a noisy supply voltage, adversely affecting the performance of other subsystems being powered from this common supply. This problem can be avoided by using an input filter to smooth the large current transients of the input power supply as shown in Figure 4.18.

4.4.1 Instability Problems Associated with the Input Filter

Improper selection of the input filter component values can cause either the regulator to become unstable or alter and degrade its performance. These problems occur when the input impedance of the regulator, Z_i in Figure 4.19, becomes less than the output

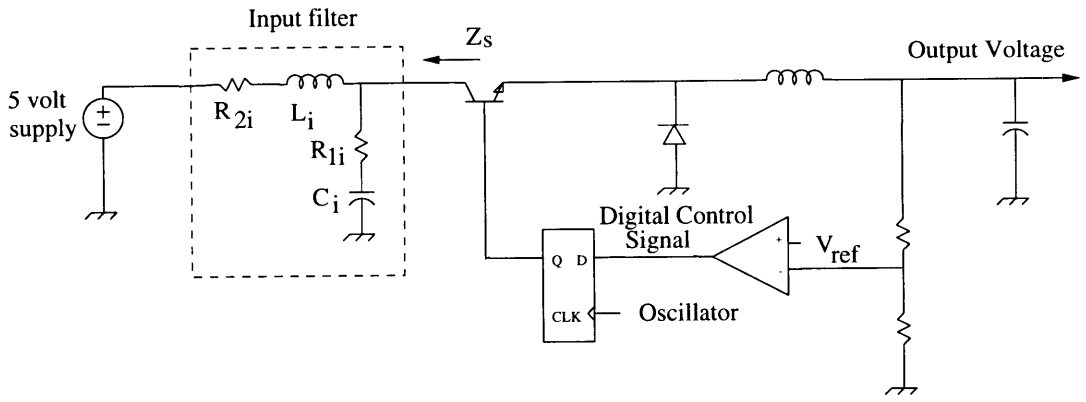


Figure 4-18: Example of an input filter implementation.

impedance of the input filter, Z_s [22, 23]. Stability can be assured by choosing the output impedance of the input filter to be lower than the input impedance of the regulator over all frequencies. Also, the input filter cutoff frequency should be lower than the output filter cutoff frequency as illustrated in the following relationships:

$$\omega_s < \omega_o \tag{4.47}$$

$$\frac{1}{\sqrt{L_i C_i}} < \frac{1}{\sqrt{LC}} \tag{4.48}$$

The input impedance of the regulator is given as:

$$\frac{1}{Z_i} = \frac{-T}{1+T}\left(\frac{D^2}{R_{load}}\right) + \frac{1}{1+T}\left(\frac{D^2}{Z_{ei}}\right) \tag{4.49}$$

[22] where T is the loop gain given by

$$T = A_{PWM} F(s) G(s) \tag{4.50}$$

$G(s)$ is the transfer function of the error amplifier and compensation, $F(s)$ is the output filter transfer function, D is the duty cycle, and A_{PWM} is the gain due to the pulse width modulator. Z_{ei} is the impedance looking into the output filter with the load current modeled as a resistor. Through proper design of the input filter,

instability problems can be avoided.

4.4.2 Matlab Modeling to Predict Instability

Matlab can be used to verify that the input filter chosen will not degrade the performance of the switching regulator. The transfer function T represented by Eq. 4.47, is equivalent to the vectors `tot1` and `tot2` defined in Section 4.3.2. The input impedance of the output filter is found to be

$$Z_{ei} = (LC_o(R_{co} + R_{load})s^2 + (L + C_o(R_{DSon}(R_{co} + R_{load}) \\ +R_{co}R_{load}))s + R_{DSon} + R_{load})((R_{co} + R_{load})C_o s + 1)^{-1} \tag{4.51}$$

The duty cycle is given by

$$D = \frac{V_{out}}{V_{in}} \tag{4.52}$$

in a steady state condition. The Matlab script developed to graphically compare the output impedance of the input filter and the input impedance of the regulator is shown below. The script uses the following variable definitions for this example:

```
a=5/1.5;          %pwm gain
c=1.65e-3;        %output filter capacitor
l=2.7e-6;         %output filter inductor
esr=0.007;        %esr of the output filter capacitor
rdson=0.02;       %rdson of the fets + esr of the inductor
c1=10e-9;         %ota compensation capacitor
c2=100e-12;       %ota compensation capacitor
r1=20e3;          %ota compensation resistor
rl=3e6;           %output impedance of the ota
rload=0.6;        %load resistor on the output filter
d2=1/3.2;         %duty cycle squared
li=1e-6;          %input filter inductor
ci=.66e-3;        %input filter capacitor
```

```
r1i=0.01;        %esr of the input filter capacitor
r2i=0.1;         %esr of the input filter inductor & source resistance
ota1=[0 1e-3];
ota2=[1e-8 1];
out1=[a*c*esr a];
out2=[l*c c*(esr+rdson) 1];
comp1=[0 c1*r1*rl rl];
comp2=[c1*c2*r1*rl (rl*c1+rl*c2+c1*r1) 1];
[eamp1,eamp2]=series(ota1,ota2,comp1,comp2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);
zei1=[l*c*(esr+rload) l+rdson*c*(esr+rload)+esr*rload*c rdson+rload];
zei2=[0 (esr+rload)*c 1];
[t1,t2]=series(out1,out2,eamp1,eamp2);
```

The input impedance of the regulator is then calculated and defined as $Z_i = Z_{i1}/Z_{i2}$. The vector Z_i is plotted as a function of the frequency vector `f`.

```
[z2,z1]=series(t1,t2*rload,zei2,zei1);
zi1=conv((t1+t2),z1);
zi2=conv(t2*d2,z2);
f=logspace(-2,8);
[zi]=freqresp(zi1,zi2,f);
loglog(f,zi)
```

With the voltage source in Figure 4.18 set to zero, the output impedance of the input filter is given by:

$$Z_s = \frac{L_i C_i R_{1i} s^2 + (R_{1i} R_{2i} + L_i)s + R_{2i}}{L_i C_i s^2 + (R_{1i} + R_{2i})s + 1} \tag{4.53}$$

The magnitude of this impedance is plotted on the same axes as the input impedance of the regulator and is shown in Figure 4.19.

```
zs1=[li*ci*r1i (r1i*r2i+li) r2i];
zs2=[li*ci (r1i+r2i) 1];
```

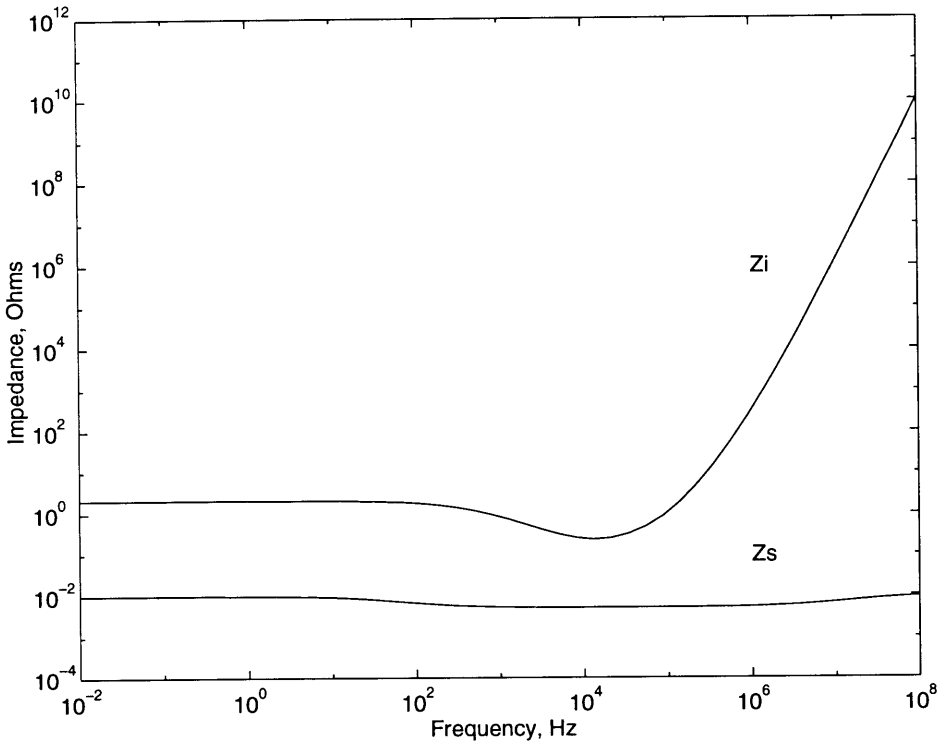


Figure 4-19: Input impedance of the regulator and output impedance of the input filter.

```
[zs]=freqresp(zs1,zs2,f);
hold on
loglog(f,zs,'r')
%red line should be lower than yellow line for
%stability.
axis([1e-2 1e8 0.1*min([min(zs) min(zi)]) 10*max([max(zs) max(zi)])])
```

The two curves are now compared across the frequency range of interest. If the input impedance of the regulator is greater than the output impedance of the input filter at every point, the variable `filtergood` is set to 1 to indicate that the filter characteristics will not cause instability problems.

```
filtergood=1;
n=1;
for n=1:1:length(f);
```

```
if zs(n)>zi(n),
     filtergood=filtergood-1;
end
n=n+1;
end
filtergood
if filtergood<1,
     filtergood=0;
end
```

The second requirement is that the cutoff frequency of the input filter needs to be lower than that of the output filter. If this requirement is met, the variable `filtergood2` is set to 1.

```
wi=1/sqrt(l*c);
ws=1/sqrt(li*ci);
filtergood2=1;
if ws>wi,
filtergood2=0;
end
filtergood2
hold off
```

The script presented here can now be used to design an input filter that does not adversely affect the performance of the switching regulator system.

4.5 Simulink

Simulink is a graphical extension of Matlab that allows transient simulations to be performed with the transfer functions generated by Matlab. Each transfer function is represented by its appropriate subsystem in a Simulink block diagram. These subsystem blocks are then arranged by the user or programmer to model the regulator

system. Along with transfer function blocks, Simulink also allows logic and non-linear functions to be implemented such as rate limiting and delay elements. The results of performing a transient simulation on a particular block diagram can be observed by viewing the waveform at any place in the system using the "oscilloscope" function block or by saving the data to a vector and plotting it in Matlab as a function of time.

4.5.1 Regulator Modeling in Simulink

A switching regulator can be modeled in Simulink with either an OTA or operational amplifier as the error amplifier. An example of a Simulink model using an OTA is shown in Figure 4.20. This model contains all of the sections of the regulator necessary to perform a transient simulation. The coarse comparators and PWM are both modeled using a functional block which accepts a differential input that is multiplied by a very high gain (100,000 is this particular example). The output is then limited to between 0 and 5 volts to represent the available supply voltages. The output of the comparators are also slew rate limited in Simulink which allows the user to model how quickly the comparators can change state. Since the comparators are also required to drive a certain amount of load capacitance associated with various logic gates, their outputs are slew rate limited to allow the user to study the effects of this non-ideal comparator characteristic.

The Simulink block diagram also contains logic which limits the power FET drive signal to a given minimum and maximum duty cycle. The FETs are modeled as ideal switches that pass the value of the either the variable V_{cc} or ground to a summation node prior to the output filter transfer function. This applies either V_{cc} or ground to the output filter at the duty cycle determined by the PWM and associated logic. The output voltage is regulated to the voltage programmed by the DAC, which is modeled as a constant at the input to the OTA. The transfer function of the OTA, compensation, power distribution network, and output filter are generated in Matlab and loaded automatically into Simulink. This is an important advantage to this type of model since these transfer functions can be quite complicated to derive and would

be difficult and time consuming to enter them manually.

Unlike the state space averaged model developed in Matlab, this model includes the coarse comparators, oscillator, PWM, logic, and power FETs. The effect on system stability of all of these components with the exception of the coarse were included in the Matlab frequency domain model. These comparators were not necessary to include in the frequency domain model because they have limited impact on system performance during normal steady state operating conditions. However, in a transient simulation, it is important to include these comparators to observe their effect on the output voltage during a large step change in load current.

4.6 Final script

The final Matlab script is shown in Appendix A. This version incorporates all of the parameter modeling described in the previous sections. The user is prompted for the relevant regulator system information such as PWM characteristics, output filter inductor value, power FET on-resistance, and error amplifier type. The script then determines compensation values, cost, number and type of output filter capacitors, and loop bandwidth.

4.6.1 Cost and Performance Optimization

The Matlab script begins the regulator design process by asking users if they wish to modify the output filter capacitor data available in the pre-existing data base. For each type of capacitor, the capacitor value, ESR, cost, and a factor indicating size and reliability is listed. The data for a particular capacitor is listed in a row of the Matlab matrix `ctype`. The capacitor information can be deleted by deleting a row or created by entering new information as prompted by the Matlab script. After each change is made, the matrix `ctype` is display so that the user can verify that the desired changes have been made. An example of this portion of the user interface is shown below.

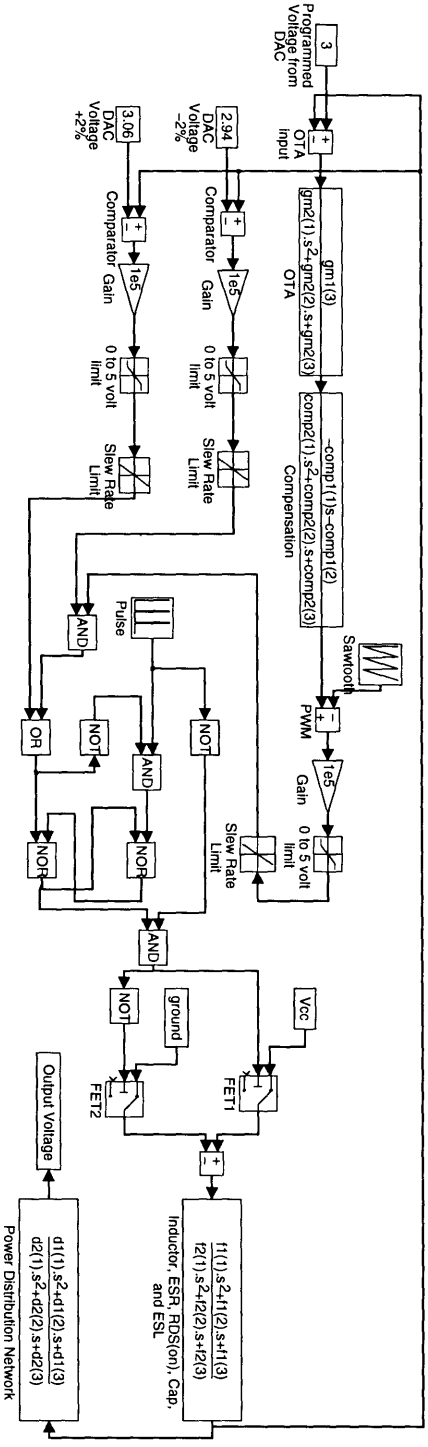


Figure 4-20: Example of a Simulink model used in transient analysis.

```
Capacitor data is currently as shown below.
     Value      ESR        Factor     Cost
ctype =
    0.0003    0.0300    1.0000    1.0000
    0.0150    0.0200    3.0000    0.1200
    0.0018    0.0390    2.0000    0.1200
    0.0003    0.1000    1.0000    0.5000
    0.0003    0.0180    1.0000    1.0000

       1:  Use this data.
       2:  Enter data for a new capacitor.
       3:  Delete data for a capacitor.

     Enter your choice: -->  3
     Enter row to delete: -->  5

    Value      ESR        Factor     Cost
ctype =
    0.0003    0.0300    1.0000    1.0000
    0.0150    0.0200    3.0000    0.1200
    0.0018    0.0390    2.0000    0.1200
    0.0003    0.1000    1.0000    0.5000

       0:  Continue editing capacitor data.
       1:  Quit editing data.

     Enter your choice: -->  0

       1:  Use this data.
       2:  Enter data for a new capacitor.
```

```
   3:  Delete data for a capacitor.

 Enter your choice: -->  2
 Enter new capacitor value (F): -->  200e-6
 Enter new capacitor ESR (ohms): -->  .03
 Enter new capacitor size and reliability factor: -->  2
 Enter new capacitor price ($): -->  .75

    Value      ESR       Factor    Cost
ctype =
    0.0003    0.0300    1.0000    1.0000
    0.0150    0.0200    3.0000    0.1200
    0.0018    0.0390    2.0000    0.1200
    0.0003    0.1000    1.0000    0.5000
    0.0002    0.0300    2.0000    0.7500

   0:  Continue editing capacitor data.
   1:  Quit editing data.

 Enter your choice: -->  1
```

After editing the capacitor data, the user is prompted for the maximum output filter capacitor ESR. This value is determined by the allowable output voltage variation divided by the maximum load current step change. Because the ESR and reliability requirements of the output filter capacitors can be quite restrictive, their cost can be a significant fraction of the total external component cost. Therefore, the user is prompted for a price limit for these capacitors. The script then finds the number of capacitors of each type necessary to meet the ESR requirement. If the total cost of the necessary number of capacitors is greater than the price limit set by the user, that capacitor type is marked as unacceptable. The necessary number of capacitors, total cost, and acceptablity are stored in the first three columns of the matrix `cdesign`,

respectively. The fourth column of cdesign indicates the total capacitance that results from using a sufficient number of capacitors to meet the ESR specification. An example is shown below.

```
Enter maximum ESR for acceptable transient response
(ohms): -->  .004
Enter maximum allowable cost of filter capacitors ($): -->  4

    Number     Cost       Acceptable
cdesign =
    8.0000    8.0000   -1.0000
    5.0000    0.6000    1.0000
   10.0000    1.2000    1.0000
   25.0000   12.5000   -1.0000
    8.0000    6.0000   -1.0000

2 type(s) of capacitors will meet both the ESR and price spec.
```

Once the capacitors have been investigated to see which types will meet the ESR and cost specifications, an output impedance versus frequency analysis, as described in Section 4.1, is performed for each type of capacitor that has been deemed acceptable. The script offers three possibilites for the power distribution network parameters. First, the values used in the paper "Fueling the Megaprocessors - Empowering Dynamic Energy Management" [18] can be chosen so that the functionality of the algorithm can be verified by matching the results with those from the paper. The second option is to enter the power distribition values manually. The final option is to use the power distribution values taken from the Intel Pentium® Pro specification available in reference [15].

The output impedance versus frequency analysis provides the minimum bandwidth the loop must have for the regulator to be able to meet the output voltage transient specifications. The user is then prompted for the regulator switching fre-

quency. If the necessary loop bandwidth is higher than the switching frequency, the capacitor type will not be suitable for the regulator application and is marked as unacceptable. If no capacitor types are acceptable after this analysis is completed, the program concludes and informs the user of possible solutions such as altering the power distribution network, increasing the total capacitance, or decreasing the ESR. The user is also prompted for the desired phase margin of the regulator. The minimum phase margin acceptable, `minpm` is then set to either 30°, or 5° less than the desired phase margin, whichever is a smaller value. The variable `minpml` is set to the smaller value of either minpm or 10° for use later in testing the regulator design for robustness.

The user is then asked to choose between an OTA or operational amplifier to be used as the error amplifier. After the amplifier type has been chosen, the user is asked to chose between four methods of compensation. The first method, type II [35], is a lead compensation capable of provide up to 90° of phase lead. The second method, type III [35], can provide up to 180° of phase lead as described in Section 4.3. This method has not been implemented for an OTA. If the user selects this type and an OTA as the error amplifer, type II is used instead. The third type is a series RC configuration that provides a single pole and zero. Finally, if the compensation values are already known, the fourth choice can be selected which allows component values for a type II configuration to be entered manually.

If an operational amplifier has been chosen, the user is prompted for its open loop gain and first two pole frequencies. From this information the open loop transfer function of the operational amplifier represented by A(s) in Figure 3.2 is found. If an OTA is chosen, the user is prompted for the transconductance, dominant pole frequency, and output impedance. These values are used to find g_m(s) in Figure 3.5. At this point, the script begins to loop through the `cdesign` matrix. For each capacitor configuration that has been found acceptable, the output filter transfer function is redefined using that capacitor value and ESR. The user is also asked for the Rdson of the FETs and the ESR of the inductor so that the output filter can be modeled properly. The gain due to the PWM and oscillator is determined by the

peak to peak voltage of the oscillator sawtooth waveform and input supply voltage and is included in the output filter model.

The compensation values for the regulator loop using an operational amplifier are found as discussed in Section 4.3.2. The desired crossover frequency inserted into the compensation routine is dependent on the regulator switching frequency and the required loop bandwidth found from the analysis of the power distribution network. If the required loop bandwidth is between 1/10 and 1/5 of the switching frequency, 1/5 of the switching frequency is used as the targeted crossover frequency. If the required loop bandwidth is less than 1/10 of the switching frequency, the targeted crossover frequency is set to 1/10 of the switching frequency. Finally, if the required loop bandwidth is greater than 1/5 of the switching frequency, the desired crossover frequency is set to 1.2 times the required loop bandwidth. This method is used because the compensation scheme for the operational amplifier may not exactly meet the targeted crossover frequency. If the target crossover frequency was the required loop bandwidth, no margin is left for calculation errors or component variations.

Each time compensation components are found for a particular type of capacitor, the compensation algorithm is performed twice. At the end of the first run, the minimum phase at frequencies below crossover is found. This is done for two reasons. First, Matlab determines the phase margin of a system finding the phase at the frequency at which the magnitude drops to one. However, if the phase is less than $-180°$ below crossover and then becomes much less negative at the crossover frequency, the system might be unstable while Matlab would predict a phase margin indicating a stable system. This situation is quite common in high performance regulator systems with a very low ESR for the output filter capacitor. Finding the minimum phase below crossover ensures that the phase margin predicted by Matlab is accurate.

Secondly, the minimum phase below crossover frequency can vary as the ESR of the capacitors and Rdson of the power FETs vary. The variable `minpml` is used to verify that the minimum phase is greater than a certain safety margin given by `-180+minpml`. After the first attempt at compensation, if the minimum phase is less than `-180+minpml`, the compensation is redone. During the second attempt, the

desired loop bandwidth is set to 1.2 times the required loop bandwidth. In many cases, this is a significant reduction in the desired bandwidth which makes achieving a reliable compensation easier. Also, the desired phase margin is set to 60° if it had been previously defined to a lower value. These two definitions are used in the second loop compensation, and the new minimum phase value is found.

The basic regulator compensation when an OTA is used has been described in Section 4.3.2. As with the operational amplifier compensation, the OTA compensation method involves two attempts. If at the end of the first attempt the minimum phase below the crossover frequency is greater than `-180+minpml`, the compensation is complete. If it is less than `-180+minpml`, the compensation algorithm is repeated with $R_1 = 100$ kΩ rather than 20 kΩ for both the series RC and type II compesation scheme. Also, for type II compensation, C_2 is set to $C_1/1000$ rather than $C_2 = C_1/100$. These two changes have the effect of moving the phase boost to a lower frequency and also spreading the phase boost to a wider range of frequencies. This aids in raising the lowest phase that occurs before the crossover frequency which creates a more robust compensation scheme.

These compensation algorithms are run for each type of capacitor that has been determined to meet the price and ESR specifications entered by the user. The results are stored into the matrix `cresults`. After the compensation is completed, the results are screened to find which types of capacitors allow the regulator to meet all of the requirements.

First, the loop bandwidth obtained for a particular capacitor combination is compared to the required loop bandwidth for that capacitor combination. If the loop bandwidth obtained is not high enough, the matrix `cresults` is altered to indicate that that capacitor combination is not acceptable. Next, the phase margin obtained is compared to the desired phase margin and the capacitor type is marked unacceptable if it does not meet the minimum acceptable value. Finally, the minimum phase below crossover, which has been stored in the `crobust` vector, is compared to the minimum acceptable value and the capacitor type is marked unacceptable if the minimum phase is too negative.

After all of these tests have been performed, the number of remaining capacitor combinations that meet all specifications is counted. If no types are acceptable at this point, the program is aborted and the user is advised of parameters that could be changed to allow more capacitor types to work.

At this point, compensation values have been found for all capacitors that met the specifications created by the power distribution network analysis, and the resulting system transfer functions have been checked against the frequency specifications. Two variables, `alpha1` and `alpha2`, are defined to indicate the relative importance of cost and size in the regulator design. The vector whichcap is created, and the value `alpha1*totalcost+alpha2*totalsize` is found for each acceptable capacitor type. The script then recommends the capacitor type that has the minimum value of `whichcap`. If the user decides to use a different capacitor type than the recommended one, all of the data in `ctype`, `cdesigns`, `crobust`, and `cresults` is printed to allow an informed choice to be made.

After the user chooses a type of capacitor, all further calculations are performed only for that type. To determine the effects of the load current on the frequency response of the regulator, the user is prompted for the maximum and minimum load current and maximum and minimum programmable output voltages. This generates a maximum and minimum value of the load resistor used to model the load current. If an operational amplifier is being used as the error amplifier, the compensation transfer function is redefined with the values that correspond to the capacitor type chosen. Next, the open loop gain and pole locations are varied as described in Section 4.3.3 to determine how much variation is allowed while maintaining the desired frequency performance. If an OTA is used as the error amplifier, the same analysis is performed to determine the allowable variation in the transconductance and output impedance. In this case, however, the compensation transfer function must be redefined with each variation in the output impedance because the output impedance is modeled as part of the compensation network.

Next, the user is asked if an input filter will be used with this regulator design. If an input filter is chosen, the user is then prompted for the component values and

parasitics of the input filter. In Section 4.4.2, the algorithm used to determine if a particular input filter will cause instability was presented. In this part of the script, the input impedance of the regulator is assumed to be as low as possible by setting the load resistance to its smallest value and the steady state duty cycle to its largest value. This creates the worst case scenario for instability due to the input filter. After the two curves are plotted as described in Section 4.4.2, the user is informed of whether the input filter is acceptable. If it is not, the user is allowed to enter different component values for the input filter.

The method described in Section 4.2.3 to determine the effect of the output filter on the system transfer function is used in the final script with both the minimum and maximum values of the load resistor. The `margin` command is used to store the phase margin and crossover frequency of both cases. Their difference is found and displayed.

Finally, after the regulator design is finished from a frequency and stability point of view, the transient response can be simulated using Simulink as discussed in Section 4.4. If the user chooses to perform a transient simulation with Simulink, it is automatically started for the correct error amplifier and compensation configuration with the Simulink transfer functions predefined. Also, the user can choose to run the simulation with or without the maximum and minimum comparators. All that is left for the user to do is to start Simulink for the desired simulation time and time steps.

4.6.2 Design Methodology Summary

Designing a switching regulator with Simulink and Matlab on a system level has several advantages. First, the Matlab script is able to calculate the necessary compensation values for a given phase margin and cross over frequency. Matlab also can determine what combination of output filter capaictors should be used to meet a certain cost limitation. These advantages eliminate most hand calculations and reduce the design cycle time. Because the regulator system transfer function can be generated for any combination of components, loop bandwidth, and error amplifier characteristics, it also gives the designer insight into the effect each component has

on the system performance and cost.

Simulink simulations are valuable because they can be used to quickly find the system transient response with the transfer functions generated in Matlab. Because Simulink uses transfer functions rather than a transistor level simulation, the requirements of regulator subcircuits can be found without doing unnecessary design work. For example, the impact of an error amplifier with a particular frequency response on the system transient performance can be found without actually designing the amplifier. This reduces the number of design iterations needed. Using Matlab and Simulink to design switching regulators reduces design cycle time and increases the designer's understanding of the system. Once the key system parameters have been designed and the system has the desired stability, it can be simulated on a transistor level with Spice to verify its performance.

Chapter 5

Script Verification

A design methodology, outlined in the preceding sections, has been developed using Matlab and Simulink. This method has been correlated with both breadboards built from competitors' parts and PSpice simulations using macromodeling techniques. This correlation ensures that all necessary second order effects of the regulator have been modeled correctly and that the design method is valid.

5.1 Verification Procedure

The flexibility of the Matlab Design Methodology allows model accuracy verification using any available regulator IC and external component combination. This gives the Matlab design procedure a significant advantage since it can be correlated with any competitor's part and then used to design the next generation of voltage regulators. The verification procedure is performed using several different breadboards with a wide range of expected phase margins as well as Spice simulations and Matlab and Simulink representations of these breadboards.

The first microsecond of a typical voltage transient is shown in Figure 5.1 [18]. The load current change, occurring at a rate of 30 A/μs at the microprocessor supply, is too rapid for the regulator to respond immediately, and therefore the parasitic values of the output filter capacitor play an important role during the initial period of the transient response. The change in output voltage during a current transient

is due to three effects. First, the current in the output filter inductor cannot change instantaneously, and therefore when the load current changes from a low to high value, the extra current required by the microprocessor is temporarily provided by the output capacitor. As a result, the capacitor voltage begins to drop at a rate given by

$$\frac{dv}{dt} = \frac{I}{C} \tag{5.1}$$

The load current will ramp up to a constant value causing the voltage on the capacitor to drop as a quadratic function of time. The capacitor voltage then decreases linearly as the current reaches its final value. The second transient effect, due to the ESR of the capacitor, creates a voltage drop across this resistance as the load current is removed from the capacitor. The third effect is due to the ESL of the capacitor. This causes the voltage transient due to the ESL to be a pulse during the load transient of a magnitude given by

$$V = L\frac{di}{dt} \tag{5.2}$$

Therefore, a faster load current change will result in a larger transient spike in the output voltage. These three transient effects are illustrated in Figure 5.1. The breadboards used in this verification process do not have the 1 μF ceramic capacitors in parallel with the output filter capacitor. In the actual microprocessor board layout, these capacitors would alter the output voltage variation equations given above.

The output voltage transient in response to a step change in load current also needs to be analyzed over a longer period of time. A typical voltage transient response measured for 300 μs is shown in Figure 5.2. The spike that appears in the first microsecond of the response is due to the effects discussed in the previous paragraph. The transient response shown in Figure 5.2 has several distinct characteristics that need to be measured in order to verify an accurate correlation with the developed model. These include the settling time (τ_s), the percent overshoot (P.O.) and, if an oscillation occurred, the damped natural frequency of that oscillation (ω_d). Also, the parasitic series resistance (R_s) of the connector between the microprocessor and the regulator can be found by noting the final value of the output voltage under a heavy

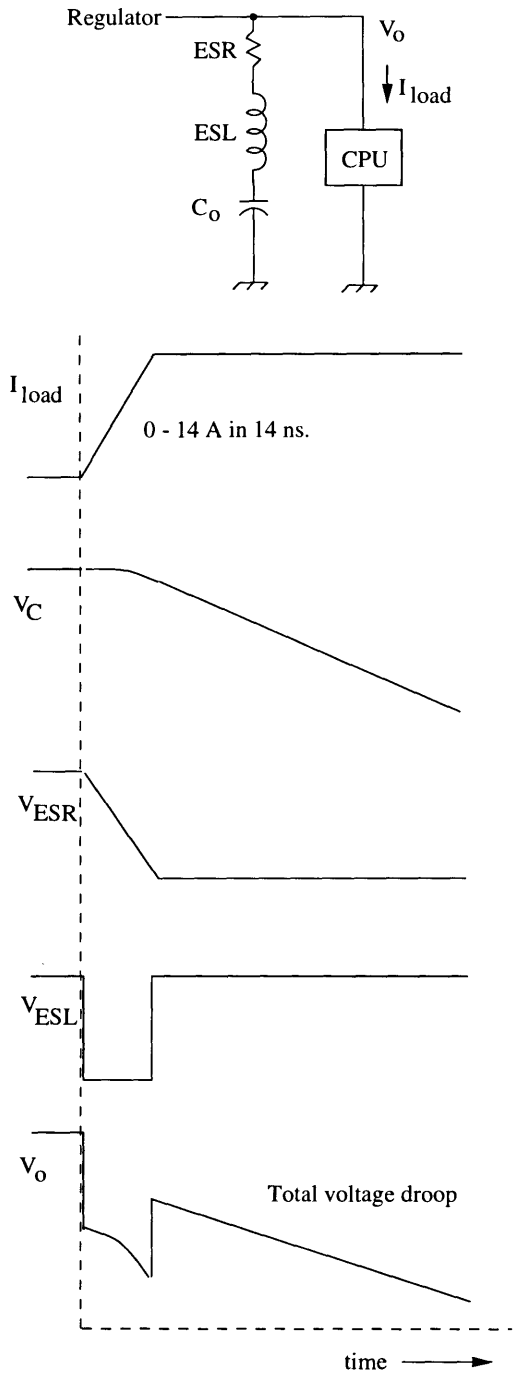


Figure 5-1: Voltage transients caused by a load current step over a 500 ns time period.

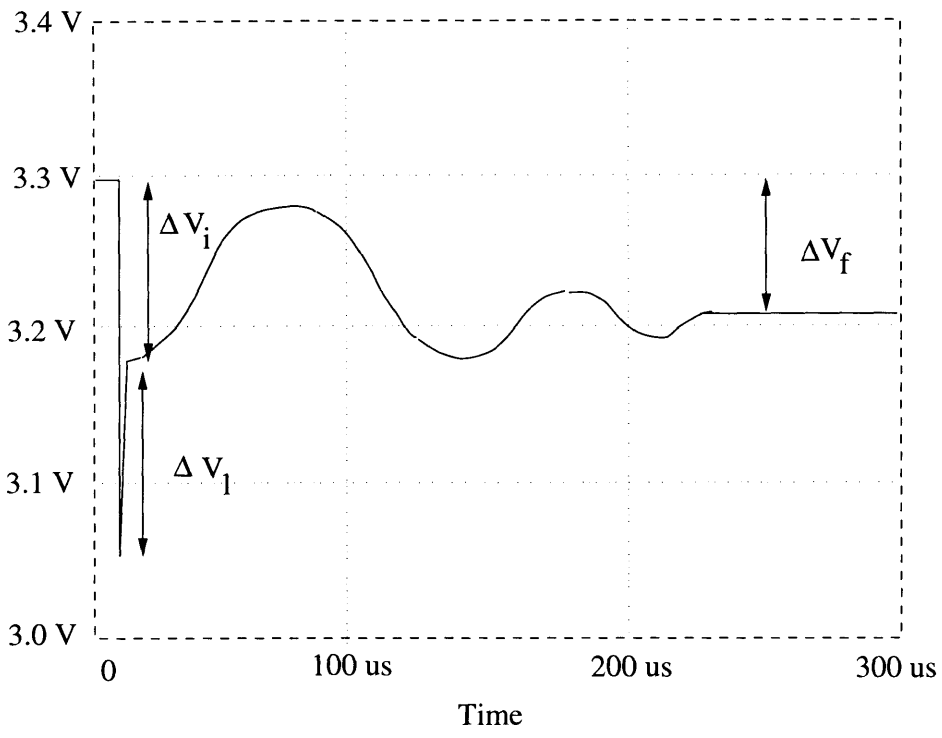


Figure 5-2: Example voltage transient caused by a load current step over a 300 μs time period.

load condition and dividing that voltage by the load current.

By examining the breadboard output waveforms shown in Figures 5.1 and 5.2, the ESR, ESL, R_s, phase margin, P.O., ω_d, and τ_s can be found. Next, the ESR, ESL, and R_s values can be used in PSpice simulations to verify that the P.O., ω_d, and τ_s parameters match the breadboard performance. Also, the regulator phase margin can be estimated from the P.O. and τ_s and correlated with the phase margin predicted by the Matlab regulator model.

5.2 Breadboards

Several regulator ICs, similar to the MC33470, were already available from competitors such as Maxim, Linear Technology, Micro Linear, and Elantec prior to the beginning of this design. Three breadboards were constructed with Linear Technol-

ogy's LTC1430 regulator IC by Mr. R. Dotson of Motorola's Amplifier and Power group. Each breadboard contained an output filter capacitor from three different manufacturers so that the script could be verified over a wide range of possible phase margins and component values. The LTC1430 is a switching regulator controller optimized for use in regulators providing 10 amps at 3.3 volts. The LTC1430 regulator IC was used because it was readily available and similar to the design requirements of the MC33470.

5.2.1 LTC1430 Breadboard Using AVX Capacitors

The first breadboard schematic using the LTC1430 regulator IC is shown in Figure 5.3. The output voltage is fed back to the negative input of the OTA through the SENSE+ pin. The OTA is compensated with a lead network at the COMP pin. The FREQSET pin, which is not connected in this configuration, causes the internal oscillator to run nominally at 200 kHz. An OTA is used as the error amplifier with a typical transconductance of 650 $\mu\Omega^{-1}$. The low frequency gain of the OTA was measured to be $A_v = 264$. Using the relationship

$$A_v = g_m R_{out} \tag{5.3}$$

R_{out} is found to be 406 kΩ. Also, the OTA can source and sink 100 μA. The LTC1430 has coarse comparators to drive the output of the PWM to its maximum or minimum value if the output voltage is outside the desired output voltage by more than $\pm$3%. Four 1 Ω resistors in parallel are connected in series with a switchable FET to simulate an 11 amp load transient on the output when the gate of the MOSFET switch is pulled to the 12 V supply. The 3.3 V regulated output voltage is then dropped across an effective 0.25 Ω load. This allows the 30 A/μs load transient to be applied. This breadboard uses 6 AVX tantalum low ESR capacitors, part number TPSE337M006R0100, connected in parallel. Each of these capacitors has a maximum ESR at 100 kHz of 100 mΩ. The breadboard uses Motorola's MMSF7N03HD power FETs, which have an R_{DSon} at room temperature of 28 mΩ specified at $V_{gs} = 5$ volts

with a drain current of between 3 and 15 amps. At 125°C, this R_{DSon} value can increase by as much as 50% and decreases slightly as V_{gs} increases. In this breadboard, two FETs are connected in parallel for M1 while only one is used for M2.

A 0 to 11 amp load current square wave was applied to the regulator output at 4 kHz. The rising and falling edges occurred at 30 A/μs. The output waveforms are shown in Figure 5.4. Each photograph is a different time scale representation of the same output voltage waveform. Figure 5.4a illustrates the shortest time scale waveform, from which the ESR, ESL and R_s values can be extracted. The voltage waveform shown in this figure does not have the same shape as the one shown in Figure 5.1 due to the 20 MHz bandwidth limiting of the oscilloscope. The $ESR + R_s$ value can be found from the photograph by the following relationship:

$$ESR + R_s = \frac{\Delta V_i}{I_{load}} \tag{5.4}$$

and therefore 150 mV/11 A = 13.6 mΩ. V_i is defined as shown in Figure 5.2. The maximum ESR of each capacitor is 100 mΩ. Therefore, with six capacitors connected in parallel, the maximum ESR expected is given as

$$\frac{ESR}{n} \tag{5.5}$$

or ESR = 100 mΩ/6 = 16.7 mΩ, where n is the number of output filter capacitors in parallel. The inductor value was measured to be 2.8 μH with an ESR of less than 1 mΩ. The ESL is given as

$$ESL = \frac{\Delta V_l}{\frac{dI}{dt}} \tag{5.6}$$

or ESL = 100 mV/(30 A/μs) = 3.3 nH. R_s can be found by observing the final value of the regulated output voltage and noting the offset. From Figure 5.4a, the R_s value is found to be

$$R_s = \frac{\Delta V_f}{I_{load}} \tag{5.7}$$

or R_s = 30 mV/11 A = 2.7 mΩ. Using Eqs. 5.3 and 5.6, the ESR = 13.6 mΩ - 2.7 mΩ = 10.9 mΩ which is 65% of the maximum ESR value taken from the data sheet.

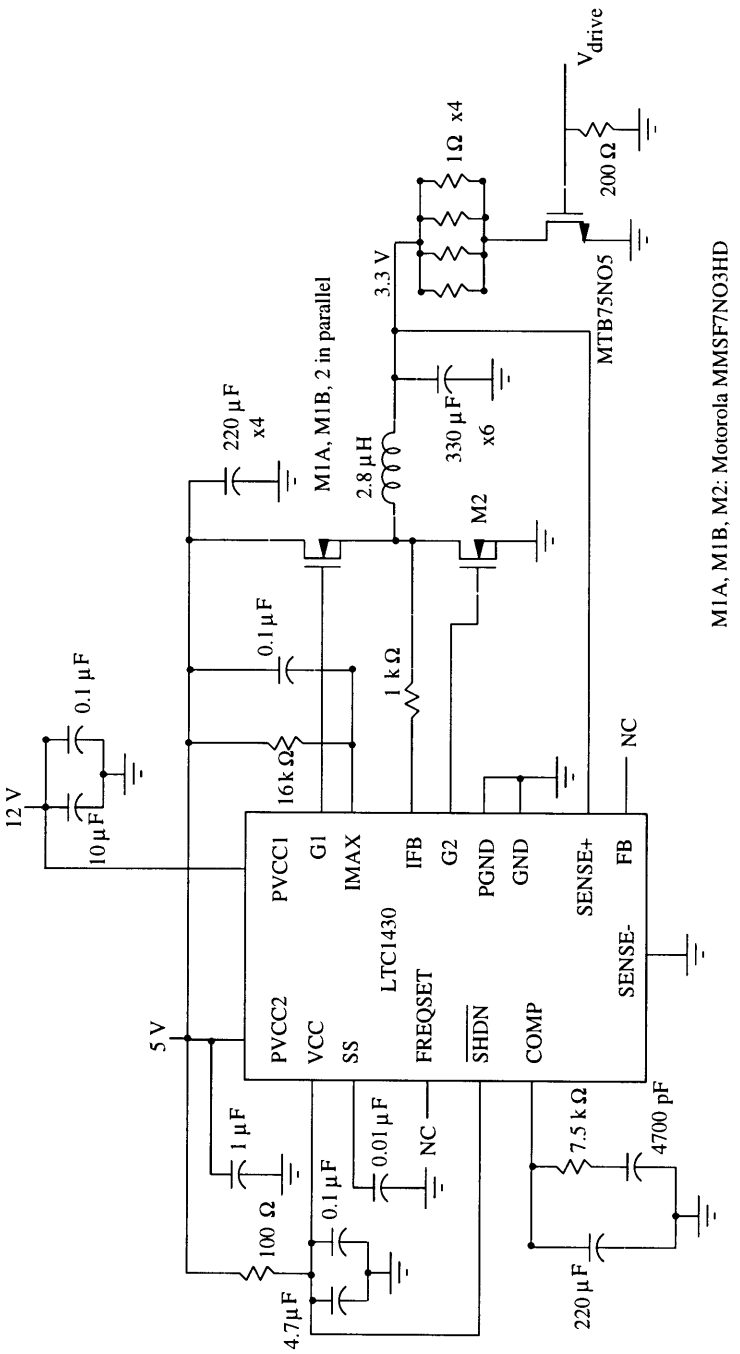
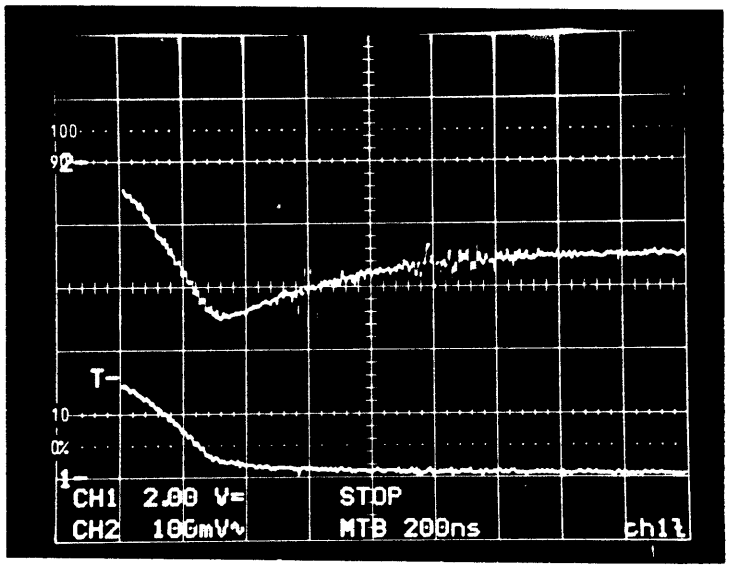


Figure 5-3: LTC1430 breadboard using AVX capacitors.



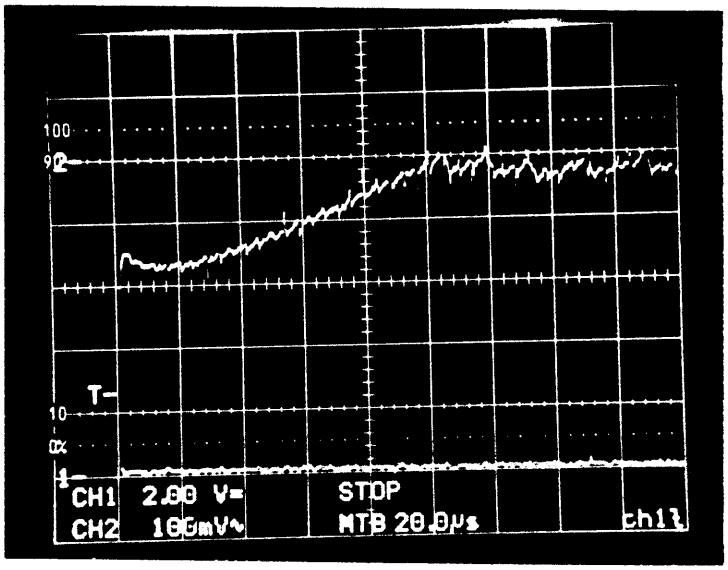


Figure 5-4: Transient response of LTC1430 breadboard with AVX capacitors: a) 0 - 2 μs. b) 0 - 200 μs.

In Figure 5.4b, it is observed that the output voltage has no overshoot in response to the load current step. This indicates that the phase margin is 60° or greater. Also, the regulator requires 100 μs, or 20 clock cycles, to restore the output voltage to the desired value.

The PSpice schematic used to simulate this breadboard is shown in Figure 5.5. The OTA is modeled as a voltage controlled current source with a transconductance of 650 $\mu\Omega^{-1}$ and an output current limit of $\pm$ 100 μA. The OTA subcircuit model is shown below.

```
*************OTA subcircuit model**************
.subckt ota 1 2 3 4
************************************************
*  vin+=1
*  vin-=2
*  vout+=3
*  vout-=4
************************************************
g1 3 4 TABLE {V(1,2)*650e-6} (-100e-6A,-100e-6A 100e-6A,100e-6A)
.ends
```

The oscillator is modeled by a piecewise linear voltage source and is described in PSpice as follows:

```
******Oscillator subcircuit model**************
.subckt oscil 1 2 3
************************************************
*  vout1=1
*  vout2=2
*  vgnd=3
************************************************
v1 1 3 DC 0 AC 0 PWL REPEAT FOR 1000 (0,3) (333n,1.5) (3.3u,3)
ENDREPEAT
```

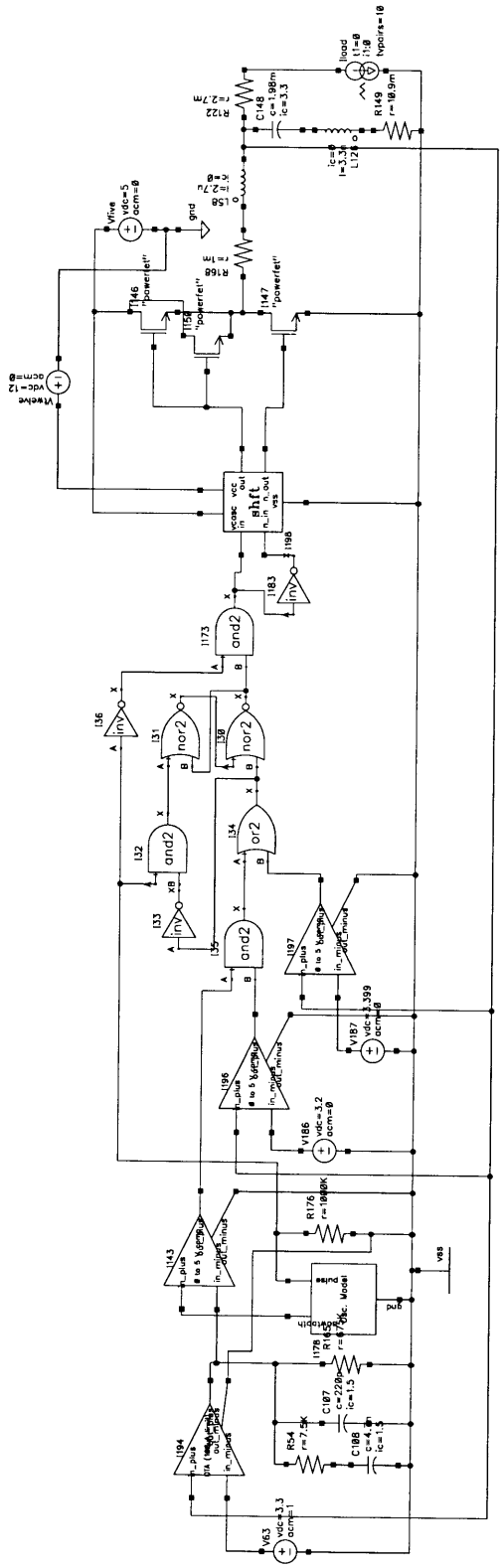


Figure 5-5: Pspice schematic of the LTC1430 breadboard using AVX capacitors.

```
v2 3 2 DC 0 AC 0 PULSE(0 -5 0 0 0 333n 3.3u)
.ends
```

The LTC1430 sawtooth waveform has a voltage range of 900 mV while this model uses a sawtooth waveform that has a 1.5 V signal range. This causes the low frequency gain factor of the breadboard to be 1.5/0.9 times greater than the low frequency gain factor used in the simulation. To account for this, the OTA output impedance is modeled as $R_{out} = 406\ \text{k}\Omega \cdot 1.5/0.9 = 675\ \text{k}\Omega$.

The maximum and minimum duty cycle comparators are modeled by a high gain voltage controlled voltage source with the output limited to ground and 5 volts. The model is shown below.

```
******Comparator subcircuit model**************
.subckt comparator 1 2 3 4
*********************************************
*  vin+=1
*  vin-=2
*  vout+=3
*  vout-=4
*********************************************
e1 3 4 TABLE {V(1,2)*1e5} (0v,0v 5v,5v)
.ends
```

The power FET model used by PSpice is that of the Motorola MMSF5NO3HD and is included in Appendix B [2, 5]. The model indicates that the nominal R_{DSon} of this power FET is 26.5 mΩ. A specific model for the power FETs used on the breadboard was unavailable. The R_{DSon} of the two power FETs connected in parallel on the breadboard was measured to be 18 mΩ, indicating that each FET has an R_{DSon} of approximately 36 mΩ.

The 11 A load transient that occurs at a rate of 30 A/μs is difficult to simulate even with the use of macromodeling techniques. Adjustment of tolerance values to assist PSpice convergence problems allowed the simulator to perform the first load

transient but failed to converge at the beginning of the second load transient. The output voltage for the first two microseconds of the transient is illustrated in Figure 5.6a. The complete simulation covering a two hundred microsecond transient took several hours to perform on a Sun Sparc station 10 and created a 31.5 Mbyte data file.

Figure 5.6b expands the time scale in Figure 5.6a. Figure 5.6a appears to have the same response as that represented in Figure 5.1. Neither figure was subject to the 20 MHz bandwidth limit imposed by the oscilloscope that alters the transient response shown in Figure 5.4b. A comparison of Figures 5.4b and 5.6b indicates that while the actual voltage regulator requires 100 μs to return the output voltage to the desired value, the PSpice data shows that the regulator requires 50 μs. Both figures show that the regulator has a phase margin of 60° or greater because there is no overshoot in response to the load transient. These curves shown in Figures 5.6 and 5.4 have the same defining characteristics as expected because the breadboard parasitic values were used in the PSpice simulation. This verifies that all important parasitic values have been included in the PSpice model presented in Figure 5.5.

The values taken from the breadboard were then entered into a modified Matlab script to find the expected phase margin of the regulator. The script used is shown below.

```
% Matlab Script to Analyze Phase Margin of LTC1430 with AVX Capacitors
% OTA Description including Compensation
gm=.65;          %OTA transconductance
gm=gm*1e-3;
p1=3e6;          %OTA dominant pole location
p1=p1*2*pi;
rf=7.5e3;        %compensation resistor
c1=4.7e-9;       %compensation capacitor
c2=220e-12;      %compensation capacitor
rl=406e3;        %output impedance of the OTA
% Output Filter Component Values and Parasitics
```

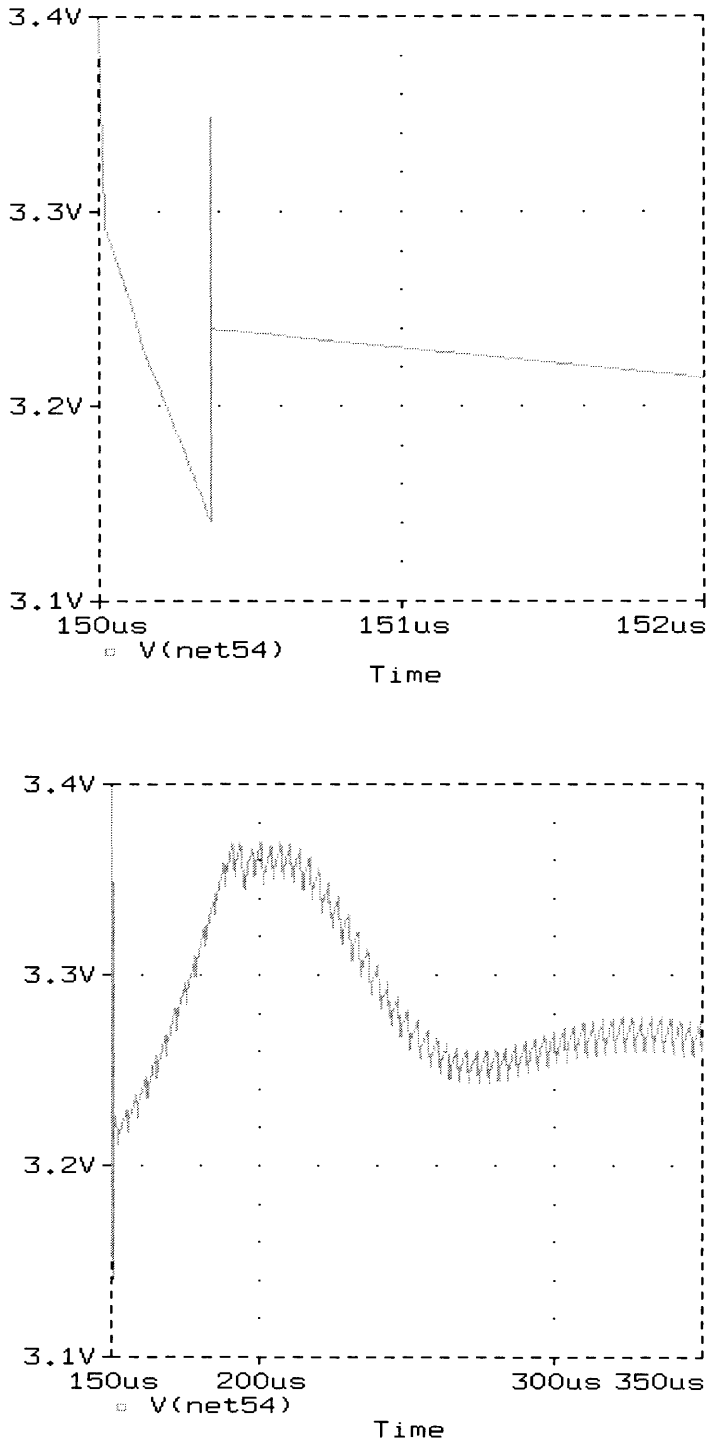


Figure 5-6: Pspice transient response of generated from Figure 5.5 for a) 0 - 2 μs and b) 0 - 200 μs.

```
l=2.8e-6;
c=1.98e-3;
esr=0.0136;
rdson=0.018+0.001;      %the Rdson includes both the Rdson and
                        %ESR of the inductor
% PWM Gain Factor
a=0.9;
a=5/a;
% Phase Margin Computation
g1=[gm];
g2=[1/p1 1];
comp1=[c1*rf*rl rl];
comp2=[c1*c2*rf*rl (rl*c1+rl*c2+c1*rf) 1];
[op1,op2]=series(a*g1,g2,comp1,comp2);
r2=0.6; %average load current
r1=esr;
lco=3.3e-9;
out2=[c*(lco+l) c*(r1+rdson) 1];
out1=[lco*c r1*c 1];
[tot1,tot2]=series(out1,out2,op1,op2);
margin(tot1,tot2)
```

Execution of the script indicates that the phase margin of the regulator is 57°, assuming that the R_{DSon} of M1 is 18 mΩ as measured on the breadboard. The open loop transfer function is shown in Figure 5.7. If the R_{DSon} is assumed to be 13.75 mΩ to match the model of the two FETs in parallel in the PSpice simulation, the phase margin becomes 56°. This phase margin describes the system immediately following a load change from 0 to 11 amps. At this time, the top FETs are on at nearly the maximum duty cycle and the equivalent R_{DSon} is the 18 mΩ described above. However, when the load changes from a maximum to a minimum, the bottom FET is on most of the time as the duty cycle drops to zero. This causes the effective

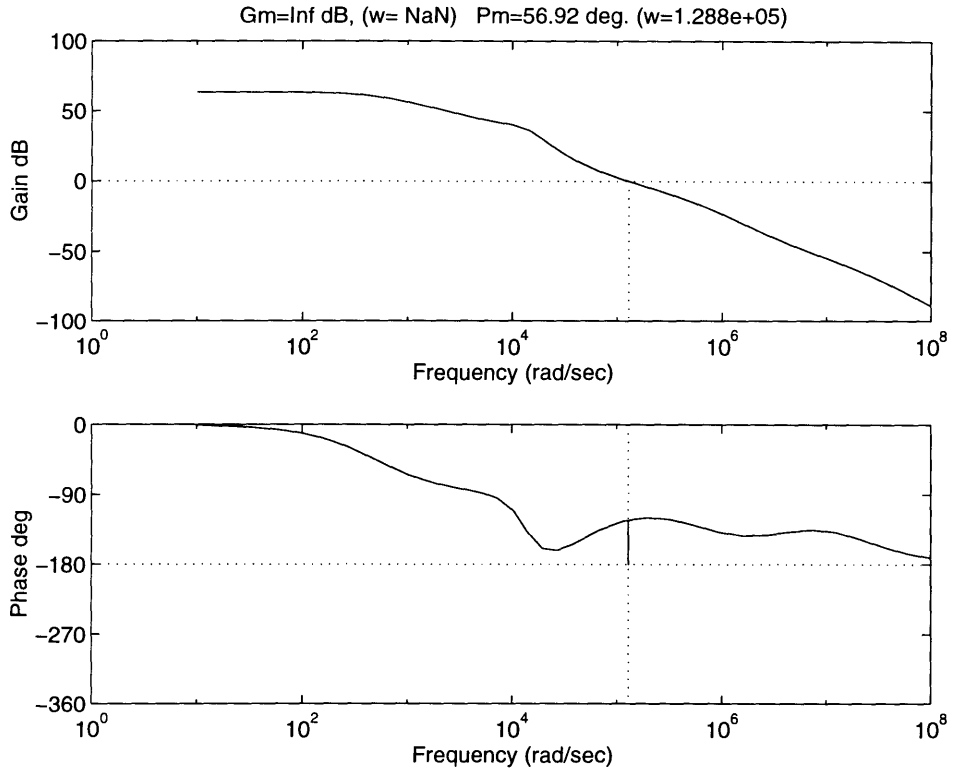


Figure 5-7: Open loop response for breadboard with AVX capacitors generated with Matlab.

R_{DSon} to be twice the value used above. When $R_{DSon} = 36$ mΩ is used, the phase margin of the system becomes 60°.

5.2.2 LTC1430 Breadboard Using Sanyo Oscon Capacitors

The second breadboard schematic using the LTC1430 regulator IC is shown in Figure 5.8. The configuration is similar to the breadboard shown in Figure 5.3 with the exception that Sanyo Oscon capacitors are used in place of the AVX capacitors in the output filter. Three 1 Ω resistors are connected in parallel at the output and in series with a FET causing an 10 amp load to be applied when the regulator is maintaining 3.3 volts at the output. This breadboard uses 7 Sanyo Oscon electrolytic capacitors arranged in parallel. These capacitors, part number 6SA330M, use an organic semiconductor as the electrolyte [28]. Each capacitor has a maximum ESR

rating at 100 kHz of 30 mΩ. The power FETs are Motorola's MTP5ONO6VL with an R_{DSon} of 25 mΩ at $V_{gs} = 5$ volts and $3 < I_d < 15$ amps. This room temperature R_{DSon} value can increase by 50% at 105°C and decreases slightly as V_{gs} increases. For this breadboard, two FETs are used in parallel for both M1 and M2.

A 0 to 10 amp load current square wave was applied to the regulator output at 4 kHz. The rising and falling edges occurred at 30 A/μs. The output waveforms are shown in Figure 5.9. From Figure 5.9a, the ESR, ESL and R_s can be found as with the previous breadboard. First, the parasitic resistance is found as ESR + R_s = 50 mV/10 A = 5 mΩ (Eq. 5.4). The maximum ESR of the each capacitor is 30 mΩ. Therefore, with seven capacitors connected in parallel, the maximum ESR expected is 30 mΩ/7 or 4.3 mΩ (Eq. 5.5). The inductor value was measured to be 2.5 μH with an ESR of 13 mΩ. The ESL is given as 350 mV/(30 A/μs) or 11.7 nH (Eq. 5.6). From Figure 5.9b, the offset voltage is used to find that R_s = 20 mV/10 A = 2 mΩ (Eq. 5.7). Therefore, the ESR is given as 5 mΩ - 2 mΩ = 3 mΩ which is 70% of the maximum ESR value for this type of capacitor.

Figure 5.9b illustrates the regulator response over a longer time scale. It can be seen that the output voltage has a significant overshoot in response to the load current step indicating a less than desirable phase margin was achieved.

The PSpice schematic used to simulate this breadboard is shown in Figure 5.10 and uses the same macromodels that were described in Section 5.2.1. Figure 5.11a is the transient performance of the regulator system for 0 to 2 μs and can be used to correlate the model with the photograph shown in Figure 5.9a. Figure 5.11b extends the time scale to examine the final regulated output voltage under a heavy load. As expected, problems with Spice convergence limited the amount of data that could be obtained. The simulation again required multiple hours of microprocessor dedication and resulted in a 40 Mbyte output file.

The values taken from the breadboard were then entered into a modified Matlab script to find the expected phase margin of the regulator. The script used is shown below.

```
% Matlab Script to Analyze Phase Margin of LTC1430 with Sanyo
```

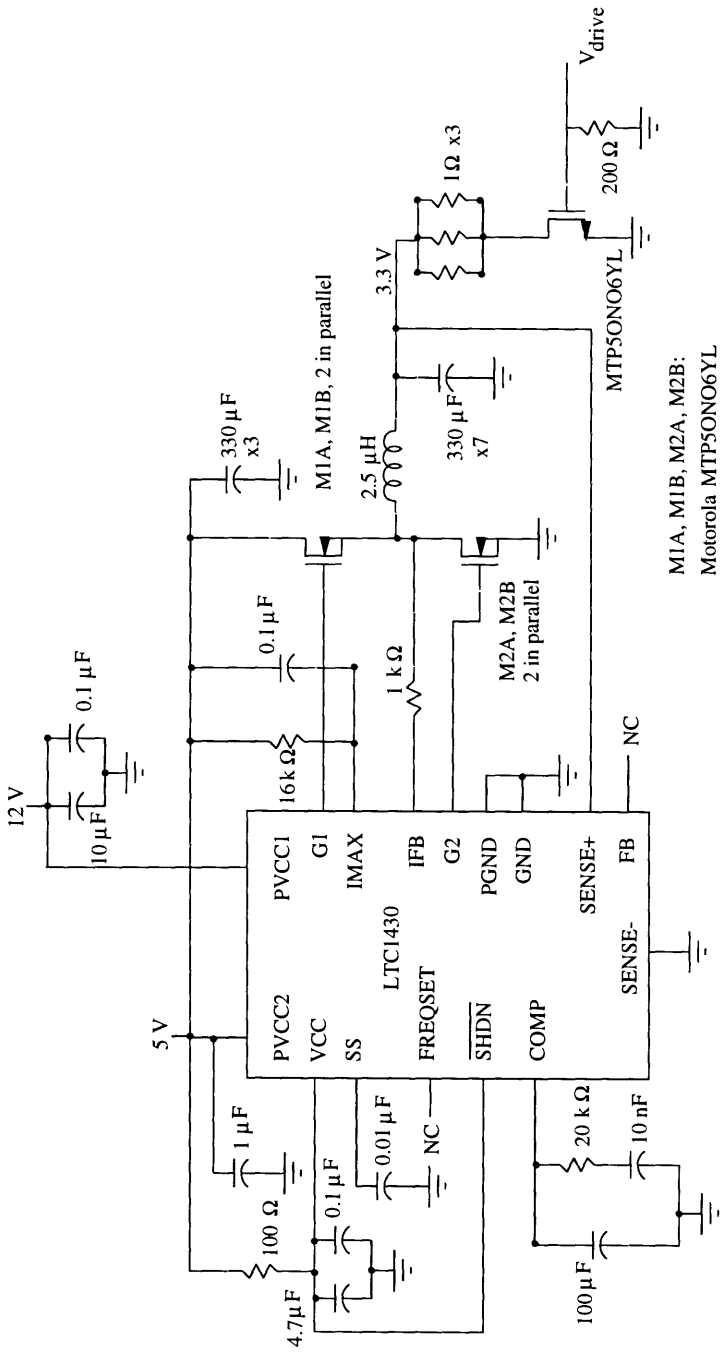
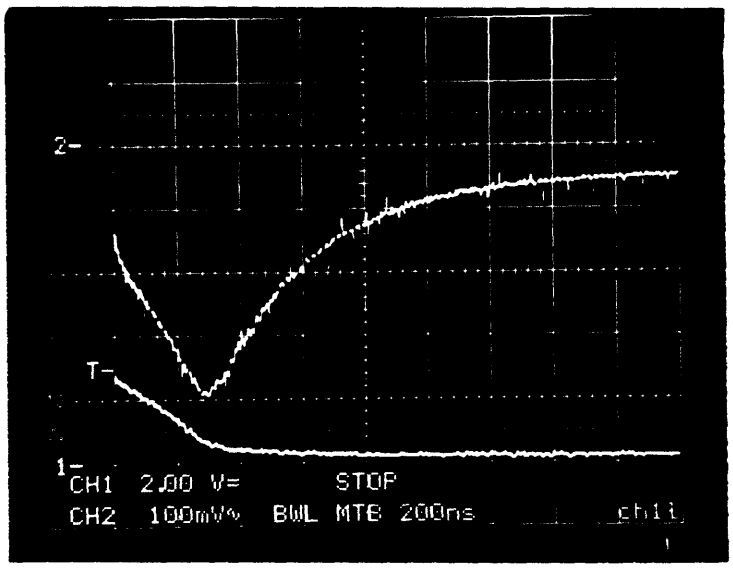


Figure 5-8: LTC1430 breadboard using Sanyo Oscon capacitors.



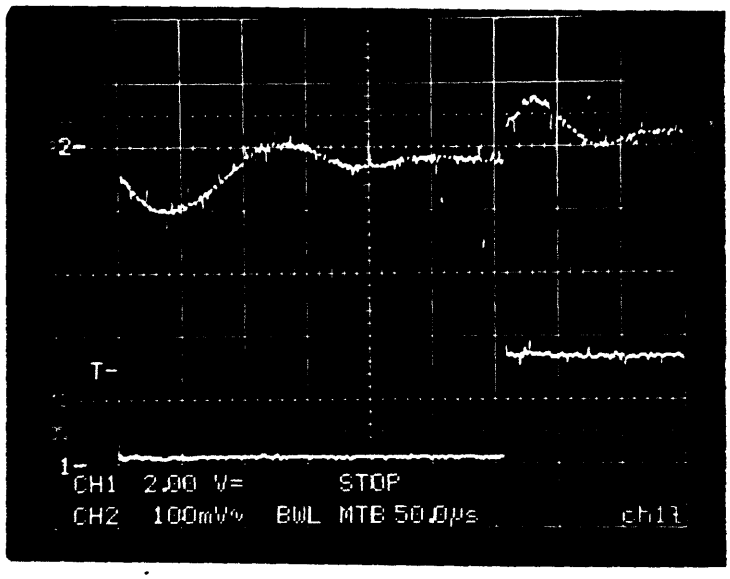


Figure 5-9: Transient response of LTC1430 breadboard with Sanyo Oscon capacitors: a) 0 - 2 μs. b) 0 - 500 μs.

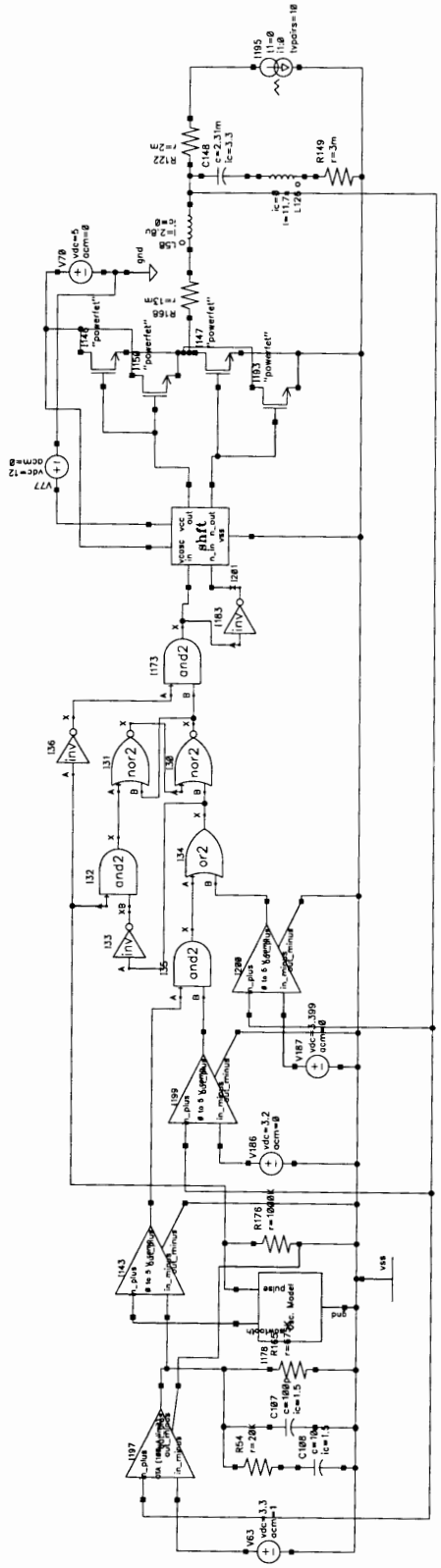
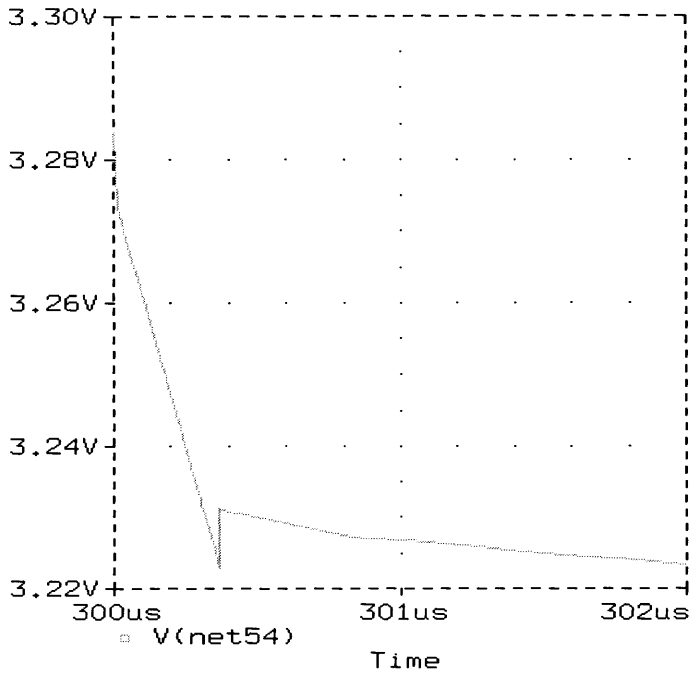


Figure 5-10: Pspice schematic of the LTC1430 breadboard using Sanyo Oscon capacitors.



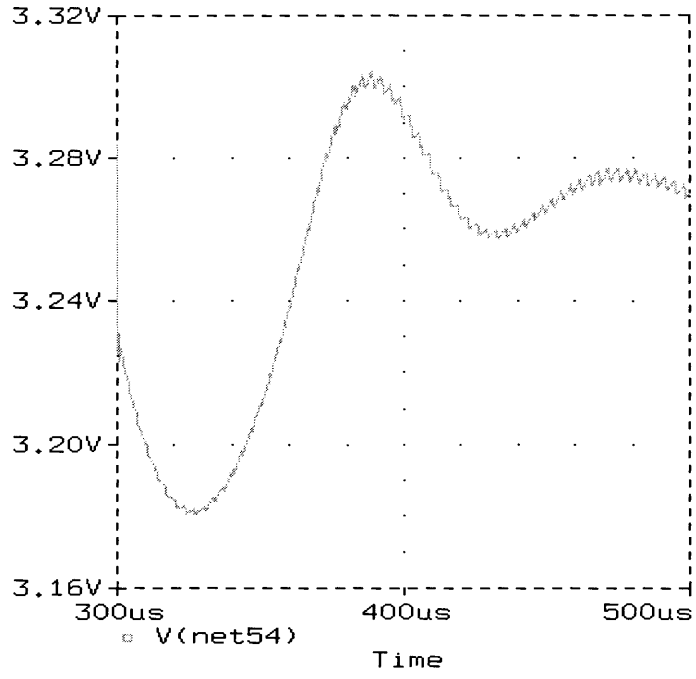


Figure 5-11: Pspice transient response of generated from Figure 5.10 for a) 0 - 2 μs and b) 0 - 200 μs.

```
% Oscon Capacitors & OTA Description including Compensation
gm=.65*1e-3;    %OTA transconductance
p1=1e6;         %OTA dominant pole location
p1=p1*2*pi;
rf=20e3;        %compensation resistor
c1=10e-9;       %compensation capacitor
c2=100e-12;     %compensation capacitor
rl=405e3;       %output impedance of the OTA
% Output Filter Component Values and Parasitics
l=2.5e-6;
c=2.31e-3;
esr=0.003;
rdson=0.0125+0.013      %the Rdson includes both the Rdson and
                        %ESR of the inductor
% PWM Gain Factor
a=0.9;
a=5/a;
% Phase Margin Computation
g1=[gm];
g2=[1/p1 1];
comp1=[c1*rf*rl rl];
comp2=[c1*c2*rf*rl (rl*c1+rl*c2+c1*rf) 1];
[op1,op2]=series(a*g1,g2,comp1,comp2);
r2=0.6;
r1=esr;
lco=11.7e-9;
out1=[lco*c r1*c 1];
out2=[c*(lco+l) c*(r1+rdson) 1];
[tot1,tot2]=series(out1,out2,op1,op2);
margin(tot1,tot2)
```

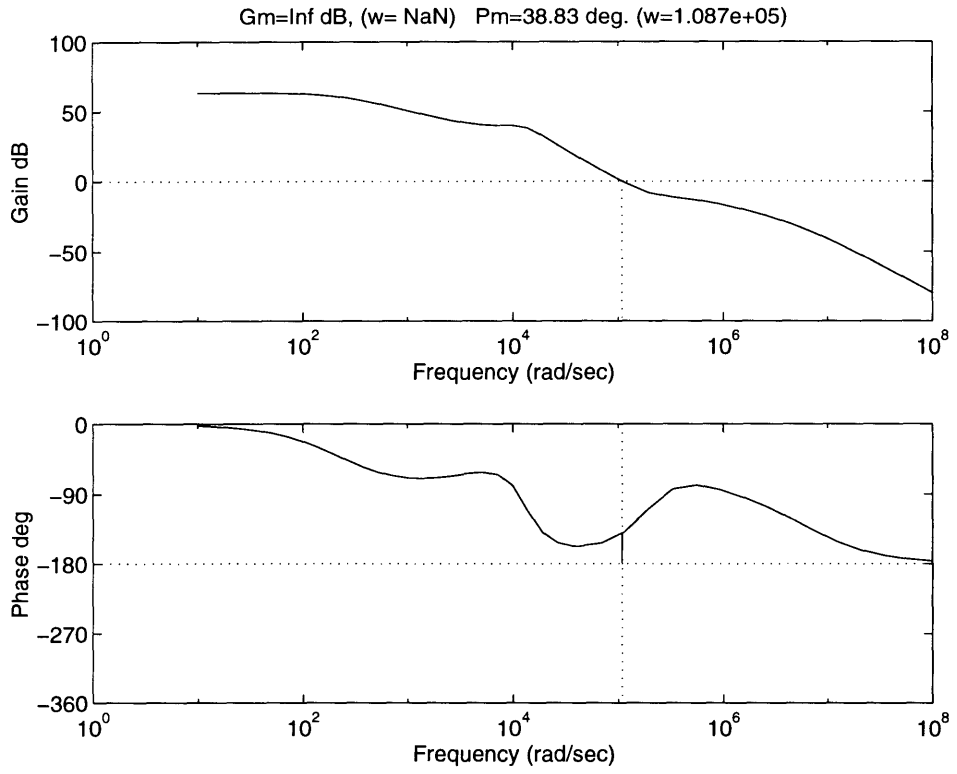


Figure 5-12: Open loop response for breadboard with Sanyo Oscon capacitors generated with Matlab.

Assuming the R_{DSon} is given by 25 mΩ/2 = 12.5 mΩ, the phase margin is found to be 39°, which agrees with the overshoot shown in Figures 5.9 and 5.11. The open loop transfer function of the regulator is shown in Figure 5.12. The dominant complex pole pair was found using Matlab, from which ω_n was found to be $2\pi \cdot 15.8 \cdot 10^3$ rad/sec and $\zeta = 0.29$. Using the following equation,

$$w_d = w_n\sqrt{1-2\zeta^2} \tag{5.8}$$

w_d was calculated to be $2\pi \cdot 14 \cdot 10^3$ rad/sec. From the breadboard results in Figure 5.9, $\omega_d = 2\pi \cdot 9$ rad/sec. The PSpice simulation is used to find that $\omega_d = 2\pi \cdot 11$ rad/sec. The peak-to-peak output voltage for the breadboard was 110 mV while the output voltage of the PSpice simulation had a peak-to-peak value of 120 mV.

5.2.3 LTC1430 Breadboard Using Nichicon Capacitors

The third breadboard schematic using the LTC1430 regulator IC is shown in Figure 5.13. The setup is similar to the previous two breadboards, with three 1 Ω resistors connected in parallel at the output and in series with a FET causing an 10 amp load to be applied when the regulator is maintaining 3.3 volts at the output. Four Nichicon aluminum electrolytic capacitors, part number UPL1A182MHH, are connected in parallel for the output filter capacitor. Each of these capacitors has a maximum ESR at 100 kHz of 39 mΩ. The power FETs used are Motorola's MTP5ONO6VL, as were used in the previous breadboard.

A 0 to 10 amp load current square wave was applied to the regulator output at a rate of 4 kHz with the rising and falling edges occurring at 30 A/μs. The output waveforms are shown in Figure 5.14. The ESR, ESL and R_s values were determined as before: ESR + R_s = 150 mV/10 A = 15 mΩ (Eq. 5.4). With four capacitors connected in parallel, the maximum ESR expected is 39 mΩ/4=9.75 mΩ (Eq. 5.5). The inductor was the same type as that used in the breadboard with Oscon capacitors. The ESL is given by 250 mV/(30 A/μs) = 8.3 nH (Eq. 5.6). From Figure 5.14b, the offset voltage is used to find the value of R_s. R_s = 30 mV/10 A = 3 mΩ (Eq. 5.7). Therefore, the ESR = 15 mΩ - 3 mΩ = 12 mΩ which is more than the maximum ESR value expected for this type of capacitor.

The PSpice schematic used to simulate this breadboard is shown in Figure 5.15 and uses the same macromodels that were described in Section 5.2.1. Figure 5.16a is the transient performance of the regulator system for 0 to 2 μs and can be used to correlate the model with the photograph shown in Figure 5.14a. Figure 5.14b extends the time scale to examine the final regulated output voltage under a heavy load. As expected, problems with Spice convergence limited the amount of data that could be obtained. The simulation again required multiple hours of simulation and resulted in a 45 Mbyte output file.

The values taken from the breadboard were then entered into a modified Matlab script to find the expected phase margin of the regulator. The script used is shown

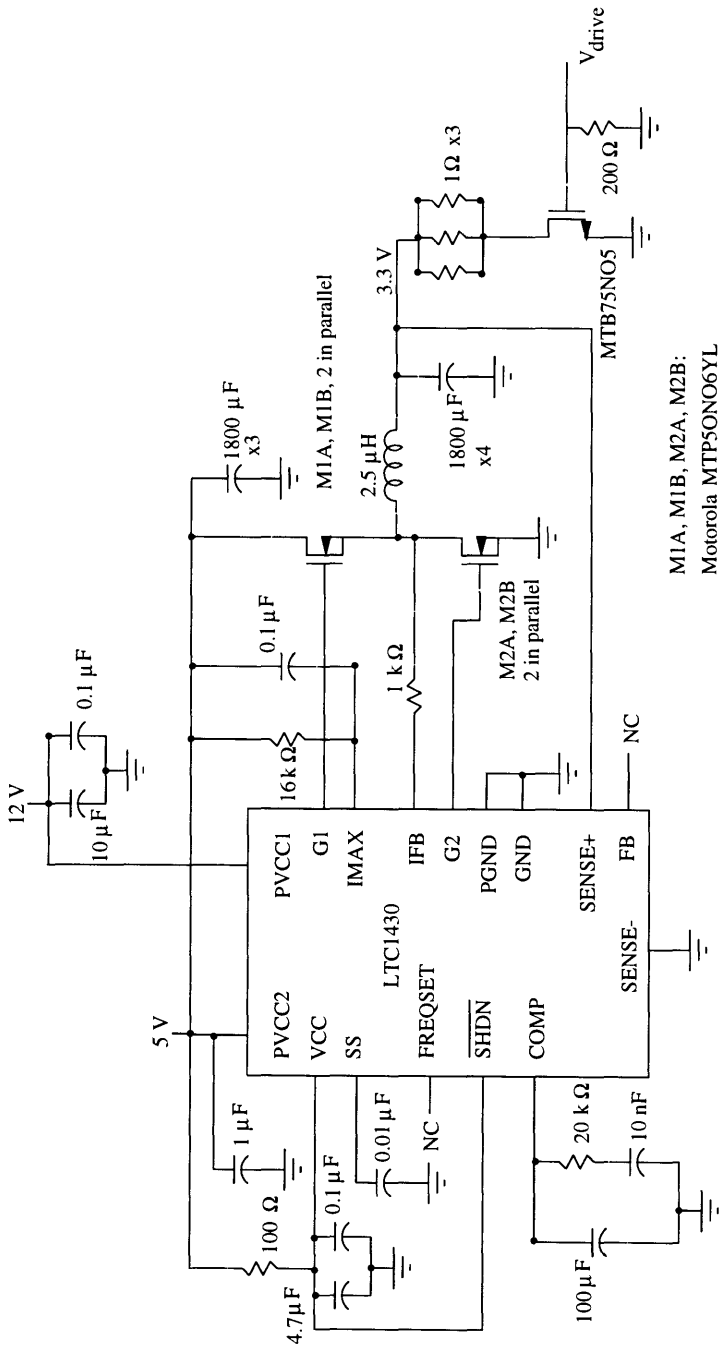
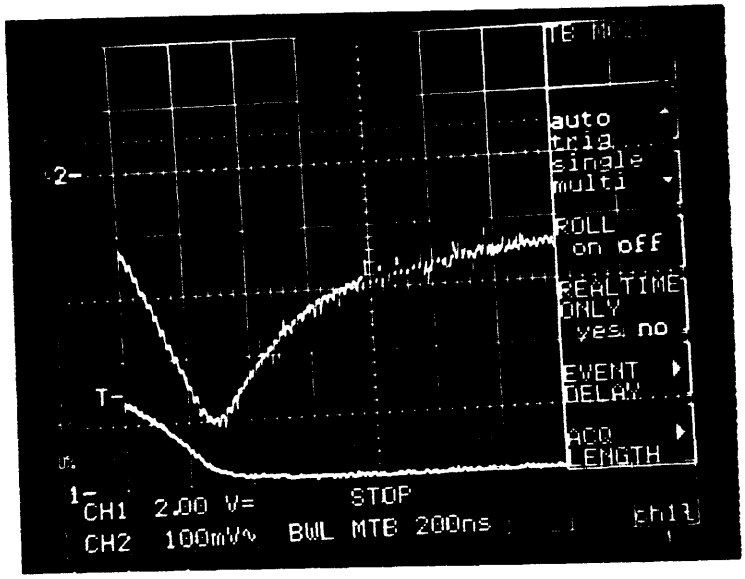


Figure 5-13: LTC1430 breadboard using Nichicon capacitors.



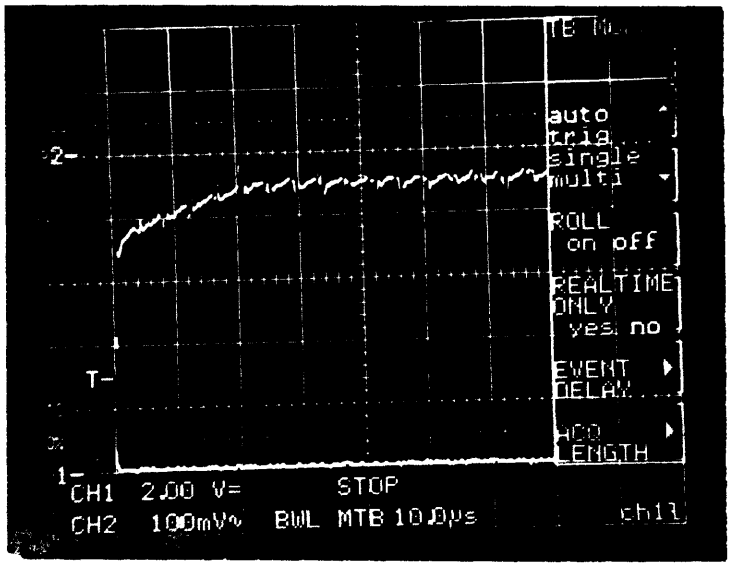


Figure 5-14: Transient response of LTC1430 breadboard with Nichicon capacitors: a) 0 - 2 μs. b) 0 - 200 μs.

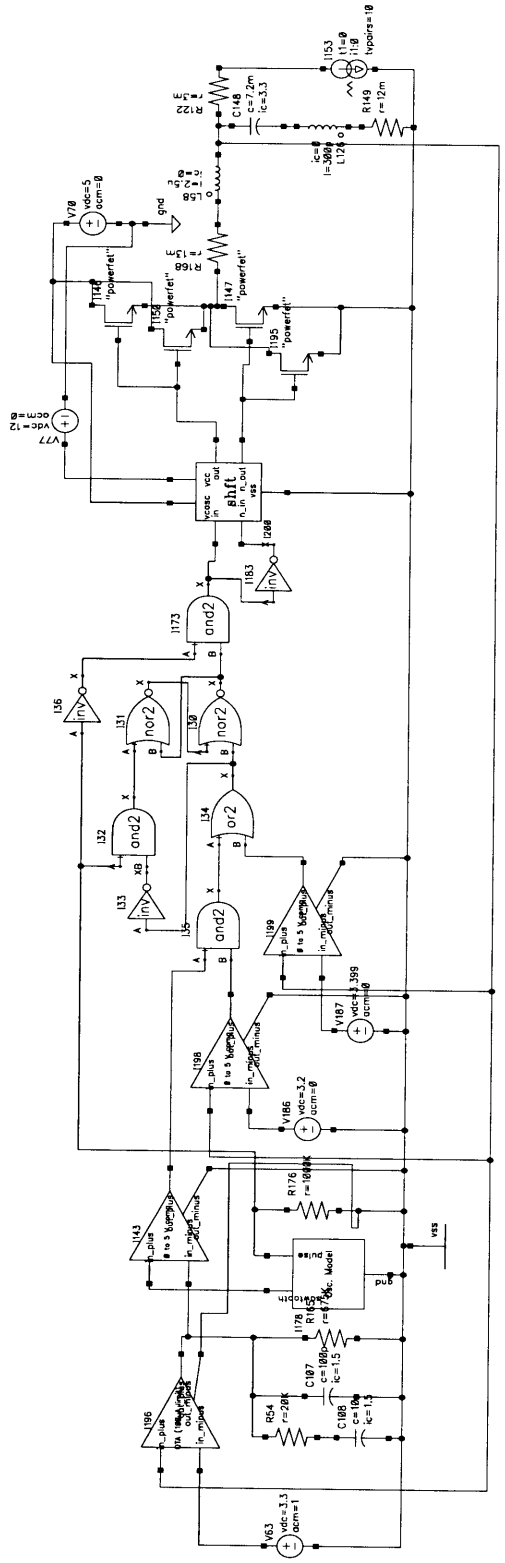


Figure 5-15: Pspice schematic of the LTC1430 breadboard using Nichicon capacitors.

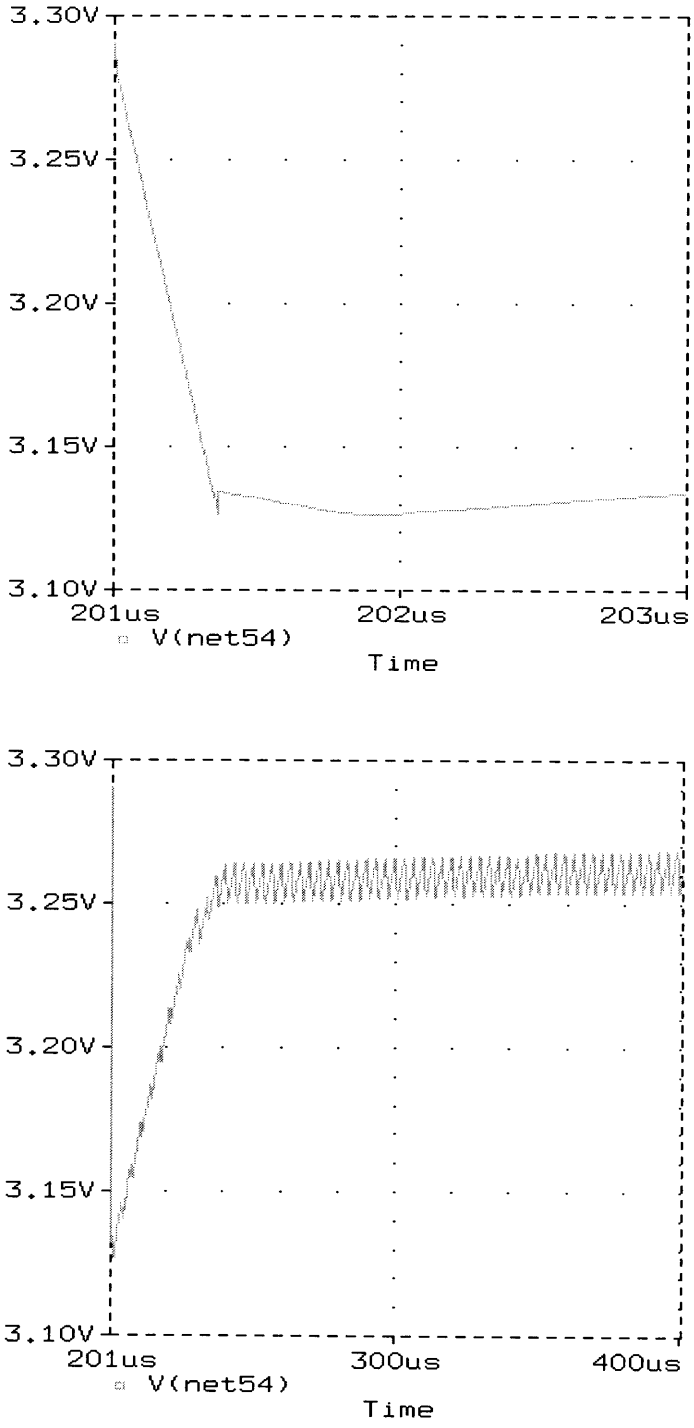


Figure 5-16: Pspice transient response of generated from Figure 5.15 for a) 0 - 2 μs and b) 0 - 200 μs.

below.

```
% Matlab Script to Analyze Phase Margin of LTC1430 with Nichicon
% Capacitors & OTA Description including Compensation
gm=.65e-3;      %OTA transconductance
p1=1e6;         %OTA dominant pole location
p1=p1*2*pi;
rf=20e3;        %compensation resistor
c1=10e-9;       %compensation capacitor
c2=100e-12;     %compensation capacitor
rl=406e3;       %output impedance of the OTA
% Output Filter Component Values and Parasitics
l=2.5e-6;
c=18e-3*4;
esr=0.012;
rdson=0.0125+0.013;     %the Rdson includes both the Rdson and
                        %ESR of the inductor
% PWM Gain Factor
a=0.9;
a=5/a;
% Phase Margin Computation
g1=[gm];
g2=[1/p1 1];
comp1=[c1*rf*rl rl];
comp2=[c1*c2*rf*rl (rl*c1+rl*c2+c1*rf) 1];
[op1,op2]=series(a*g1,g2,comp1,comp2);
r2=0.6;
r1=esr;
lco=8.3e-9;
out1=[lco*c r1*c 1];
out2=[c*(lco+l) c*(r1+rdson) 1];
```

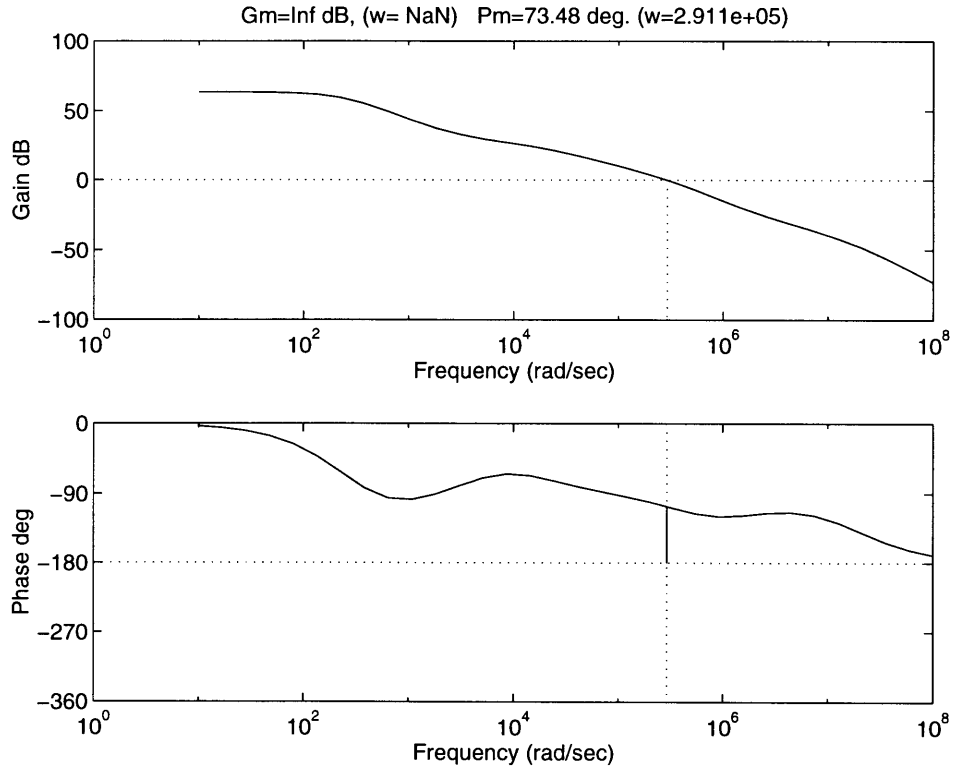


Figure 5-17: Open loop response for breadboard with Nichicon capacitors generated with Matlab.

```
[tot1,tot2]=series(out1,out2,op1,op2);
margin(tot1,tot2)
```

In both the PSpice simulation in Figure 5.16 and the breadboard results in Figure 5.14, the output voltage does not exhibit any overshoot, indicating that the regulator has a phase margin of greater than 60°. This agrees with the phase margin of 73° in Figure 5.17 found using Matlab. The output voltage of the breadboard takes 25 μs to to recover from the load current step. The PSpice simulation shows that the output voltage recovery time is 40 μs. The recovery time of this breadboard is faster than that of the breadboard using AVX capacitors for two reasons. First, the bandwidth of the breadboard using Nichicon capacitors is slightly more than twice that of the breadboard using AVX capacitors. Secondly, the inductor value is slightly smaller, allowing the current to change more quickly.

In this chapter, three breadboards have been correlated with both the Matlab script and PSpice. The phase margin predicted by Matlab agrees with the output voltage transient characteristics of the breadboards and PSpice. The Matlab script can be used to study the effect of system parameters such as output filter capacitor ESR and MOSFET R_{DSon} on system stability without needing to run long, memory intensive PSpice simulations.

Chapter 6

Regulator Design: MC33470

In Chapter 4, a Matlab script was written and models were developed to aid in the design of a buck switching regulator. This design method will be illustrated by using it in the design of the MC33470, a buck switching regulator control IC developed in the Amplifier and Power Group of Motorola's Logic and Analog Technologies Group.

6.1 Specifications

A simplified block diagram of the proposed MC33470 is shown in Figure 6.1. The reference voltage is set with a 5-bit DAC. A soft start function controls how quickly the output voltage can ramp up during startup. The output voltage is regulated with an error amplifier, PWM comparator, and logic so that the duty cycle can be limited to a maximum and minimum value. Two comparators are also used to force the duty cycle of the external MOSFETs to either its maximum or minimum value when the output voltage is outside of a specified regulation window. A power good indicator provides a signal indicating when the output voltage is within the desired regulation window. Also, logic is provided to shut off the regulator if the input voltage is too low or an "OFF" signal is received. Finally, the output current is sensed to provide short circuit protection.

The input voltage will be 5 V $\pm$ 5%. As described in Chapter 1, the output voltage will be between 1.8 and 3.5 volts, programmed by an integrated 5-bit DAC

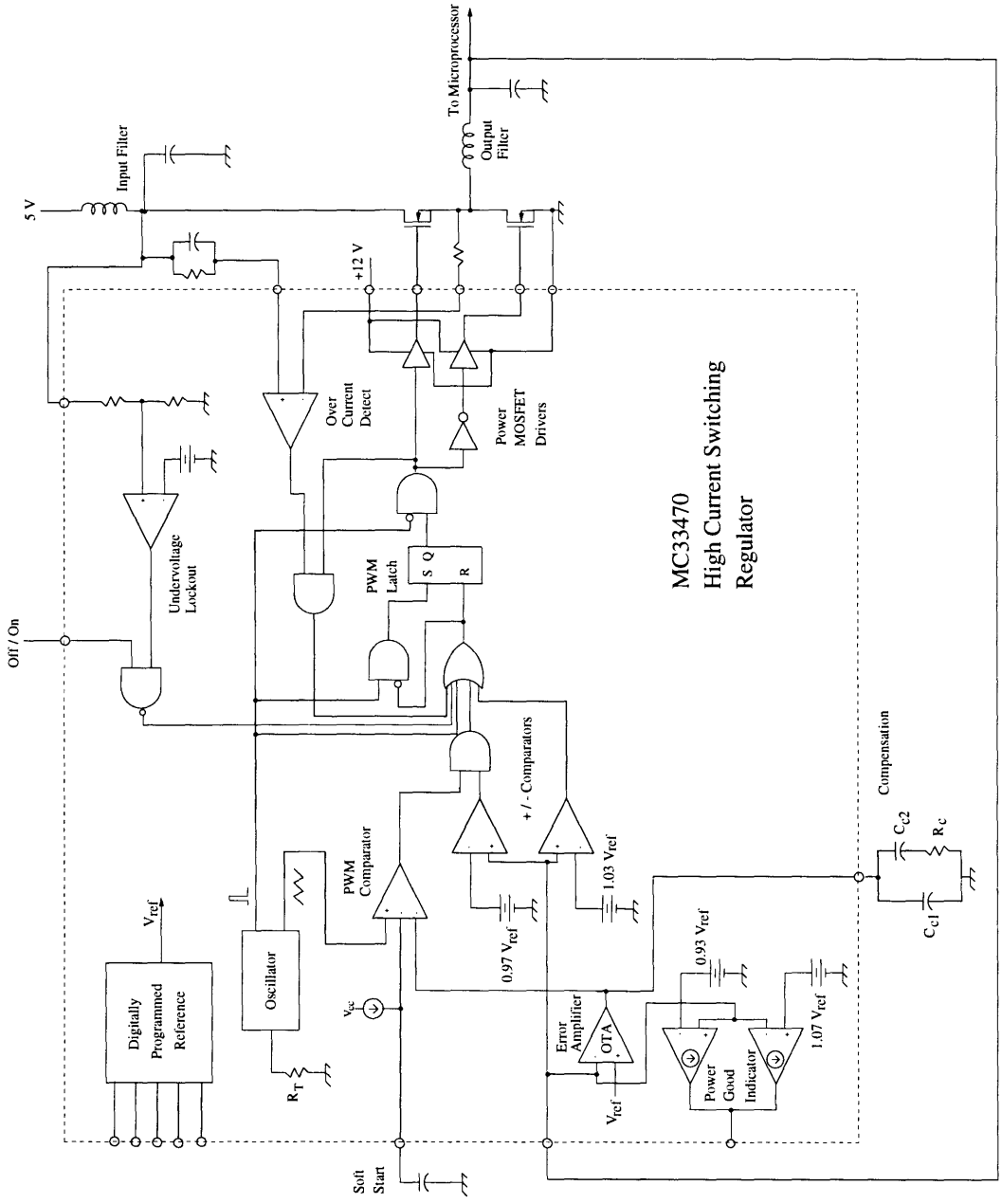


Figure 6-1: Block diagram of the MC33470.

in 50 mV increments below 2.1 V and 100 mV increments above 2.1 V. The regulator needs to maintain the output voltage to within ± 5% of its nominal value during a transition between no load and maximum load of 11 to 14 amps. The maximum load current depends on the supply voltage and is 11 amps at 1.8 volts and 14 amps at 3.5 volts. This ±5% window includes variations due to both the voltage reference and the load transient effects. The regulator IC has to be able to slew at 30 A/μs at the output pins which drive the external power devices due to current load changes of up to 1 A/ns at the pins of the microprocessor. Due to the large current demands, the regulator must have an efficiency rating of at least 80% at the maximum rated load and a minimum of 40% at low load conditions.

During startup, the error amplifier must begin to function as soon as the input voltage reaches 4.25 V, which places a tight requirement on the input stage of the amplifier when the IC is used to regulate 3.5 volts. The OTA has to have an input common mode voltage range of 1.6 - 3.9 V to handle all possible output voltages. The oscillator waveform used with the PWM has a 1.5 V swing from 1.5 to 3.0 V. The OTA must have an output signal swing of at least this range to be able to regulate the duty cycle between its maximum and minimum values.

The regulator switching frequency should be chosen to be between 200 kHz and 300 kHz. The MC33470 IC has to have an operation temperature range of 0 to 70°C. Therefore, the error amplifier and comparators have to be functional up to 125°C due to heating caused by the FET driver circuitry with will use 2 amps to drive the external power MOSFETs. Also, the mean time between failures (MTBF) for the voltage regulator module should be 500,000 hours of continuous operation at 55°C while meeting all specifications listed above. The total cost of the external components needed to meet the specifications should be minimized. Ideally, it should be less than 5 dollars, with 10 dollars as an upper bound.

6.2 Regulator Parameter Extraction from Specifications

From the specifications in the previous section, several parameter values can be determined and used with the Matlab script to find the rest of the design parameters.

6.2.1 Parasitic Values

The output filter capacitors tend to make up the largest fraction of the total cost of the external components. Their cost increases with decreasing ESR. Therefore, it is important to know the maximum ESR that is acceptable in this regulator design to minimize the cost.

At 2.8 V and a load current of 13 A, a voltage drop of

$$\Delta V = 13\ A \cdot (R_{co} + R_s) \tag{6.1}$$

occurs as described in Section 5.2.1. If ΔV is 5% of the output voltage, $(R_{co} + R_s) < 10.8\ \text{m}\Omega$ to meet the transient specification, assuming the capacitor has no ESL. However, as was shown in Chapter 5, the ESL cannot be neglected. The voltage change due to the ESL is given by $V = L\frac{di}{dt}$. With $\frac{di}{dt} = 30\ A/\mu s$ and L = 1 nH, the voltage change is 30 mV, which constrains $(R_{co} + R_s) < 8.5\ \text{m}\Omega$.

In future generation processors, the regulator output voltage will be lower and therefore have a smaller absolute tolerance window. The calculations in the preceeding paragraph would then no longer be valid, and the external components would need to be changed. For this reason, the ESR and ESL values (R_{co} and L_{co}) should be minimized below the maximum values given above for a given total cost. For example, to meet the 5% window at 1.8 V with an ESL of 1 nH, $(R_{co} + R_s) < 5.5\ \text{m}\Omega$. Finally, if the cost restriction does not allow capacitors with a sufficiently low ESR to be used, the reference voltage can be adjusted by 1 or 2% of the output voltage to account for the drop due to R_s.

The equations for static power dissipation in the external MOSFETs are given

below [16].

$$R_{DSon(M1)} = \frac{V_{in}P_{max(M1)}}{V_{out}I_{max}^2} \tag{6.2}$$

$$R_{DSon(M2)} = \frac{V_{in}P_{max(M2)}}{(V_{in}-V_{out})I_{max}^2} \tag{6.3}$$

Assuming 1 W of dissipation as P_{max}, $V_{in} = 5$ V, $V_{out} = 2.8$ V, and $I_{max} = 13$ A, the maximum values are given below.

$$R_{DSon(M1)} = 10.5\ m\Omega \tag{6.4}$$

$$R_{DSon(M2)} = 13.4\ m\Omega \tag{6.5}$$

For applications in which V_{out} is around 3.5 V, the required R_{DSon} of M1 is half of the R_{DSon} of M2. This implies that in such situations, one MOSFET could be used for M2 while two in parallel are needed for M1. For this regulator design, two are needed for both M1 and M2 to achieve a low enough R_{DSon} over all possible ranges of V_{out}. These calculations were done neglecting the switch losses which are given as

$$P_{sw} = CV^2 f_{sw} \tag{6.6}$$

where C is the effective capacitance due to the power MOSFETs, V is the applied gate drive signal (12 V in the MC33470), and f_{sw} is the switching frequency. With $f_{sw} = 300$ kHz and $C = 10{,}000$ pF, the switching losses are $P_{sw} = 0.43$ W which must be included in the efficiency calculations to ensure that the specifications are met.

6.2.2 Output Filter Components

Inductor values in this application are typically in the 1 μH to 5 μH range [16]. If a larger value of inductor is used, the required capacitance also increases. This occurs because a larger inductor restricts how quickly the regulator can provide the desired output current. During the time the inductor current is changing, the difference between the inductor current and output current is provided by the capacitor, causing

a droop in the output voltage.

If a 1 μH inductor is used, the inductor current rate of change for V_{out} = 2.8 V is 2.2 A/μs. This means it takes 5.9 μs for the regulator to respond to a 13 A load change. To keep the output voltage variation due to the droop in the capacitor voltage less than 1%, $C_o > 5.9\ \mu s \cdot 13\ A/28\ mV = 2743\ \mu F$. This calculation neglects the output voltage ripple at the switching frequency. The magnitude of this ripple is given by

$$\Delta v = \frac{I_{load}(1-D)T}{C} \tag{6.7}$$

where D is the steady state duty cycle and T is the switching period. Using the worst case values of I_{load} = 15 A and D = 1.8/5 = 0.36, and the value of C found above, the ripple is found to be 13 mV. Therefore, with this value of capacitance used in the output filter, the output voltage variation will be about 1.5%. If this is deemed to be too large, the output filter capacitance should be increased.

Another important consideration in choosing a filter capacitor is its expected lifetime. The voltage regulator module is required to have a MTBF of 500,000 hours as stated above, while many capacitors have specifications based on a 2000 hour lifetime. To ensure proper operation, higher reliabilty capacitors have to be used so that the parameters such as ESR are constant over their lifetime [21].

6.2.3 Regulator Output Impedance Requirement

The Pentium® Pro microprocessor has a load current rate of change of 1000 A/μs. If the minimum current is zero amps, and the maximum is 13 A, the load current risetime is 13 ns. The rise time is used in determining what range of the regulator output impedance is acceptable and has an effect on what the bandwidth of the loop needs to be.

If the regulator output current changes by 13 A at V_{out} = 2.8 V, the low frequency output impedance must be less than 10.8 mΩ for the output voltage to remain within the 5% window as described in Section 4.1.2. Likewise, the output impedance must be less than 8 mΩ if the output voltage is 1.8 V. The load current rate of change, the

maximum output impedance, and the filter capacitor value will be used to determine the loop bandwidth specification.

6.2.4 Desired Phase Margin

The desired phase margin is generally between 45° and 60°. A lower phase margin can be acceptable as long as the percent overshoot is not large enough to exceed the 5% transient window. If a higher phase margin is used, the loop response to a load current change will be slower, which requires that the output filter capacitor be larger.

6.2.5 Operational Amplifier versus Operational Transconductance Amplifier (OTA)

For the MC33470 design, an OTA will be used rather than an operational amplifier for two reasons. First, the OTA requires fewer external compensation components to provide the same phase boost. Secondly, the OTA configuration is more common among competitors' ICs. The external component connections for an OTA and operational amplifier are different. Because of this, even if the pin outs are the same between an IC using an operational amplifier and one using an OTA, the ICs cannot be interchanged without altering the board layout. An OTA is chosen so that the MC33470 will be able to minimize the system cost and maximize compatibility.

The nominal transconductance of the OTA design can be found in the following equation

$$g_m = \frac{120\ \mu A}{2.8\ V \cdot 0.05} = 850\ \mu\Omega^{-1} \tag{6.8}$$

where the output drive capability of the OTA is assumed to be 120 μA and the output voltage is 2.8 V. This value of transconductance causes the maximum OTA output current to be provided to the compensation when the output voltage is $\pm 5\%$ of the reference voltage.

The OTA gain should be large enough that the OTA is capable of driving the

duty cycle to either extreme with only a few millivolts of error in the output voltage.

$$\begin{aligned} \frac{\Delta V_o}{\Delta V_{in}} &= g_m \cdot R_{out} \\ \frac{1.5\ V}{0.002\ V} &= 850\ \mu\Omega^{-1} \cdot R_{out} \\ \Rightarrow R_{out} &\geq 890\ k\Omega \end{aligned} \tag{6.9}$$

6.2.6 Compensation Technique

Three compensation types were described for an operational transconductance amplifier in Chapter 4. The first, a type I, has already been determined to create a bandwidth that is too low for this regulator application. The second, a type II configuration, uses three components to create a phase boost of up to 90°. The third, a series RC connection, also provides up to 90° of phase shift, but does not roll off as much as a type II, leaving the system more suseptible to high frequency noise. For these reasons, a type II configuration will be used unless it is determined that a series RC is more cost effective compensation scheme.

6.3 Matlab Script for MC33470 Design

The script used in the MC33470 design is shown in Appendix A. Using the values derived in the previous section, the script will be used to find the acceptable variation in the OTA specification, the compensation values, and other regulator design parameters.

```
     Switching Regulator Design and Optimization
     Enter the output filter inductor value (uH): -->  1
     Capacitor data is currently as shown below.

    Value      ESR        Factor     Cost
ctype =
    0.0003     0.0300     1.0000     1.0000  %Sanyo Oscon 330 uF
```

```
    0.0150    0.0200    5.0000    0.1200  %Panasonic 15,000 uF
    0.0018    0.0390    2.0000    0.1200  %Nichicon 1800 uF
    0.0003    0.1000    1.0000    0.5000  %AVX 330 uF
    0.0003    0.0180    1.0000    1.0000  %Low ESR Oscon 330 uF
```

In the capacitor types matrix shown above, the variable `factor` is used to measure relative capacitor size. It could also be used as a reliability factor or any other parameter the user wishes.

```
      1:  Use this data.
      2:  Enter data for a new capacitor.
      3:  Delete data for a capacitor.

    Enter your choice: -->  1
    Enter maximum ESR for acceptable transient response
    (ohms): -->  0.007
    Enter maximum allowable cost of filter capacitors
    ($): -->  4
```

A maximum ESR of 0.007 mΩ is chosen because it is recognized that the value of R_s will not be zero.

```
    Number    Cost      Acceptable
cdesign =
    5.0000    5.0000   -1.0000
    3.0000    0.3600    1.0000
    6.0000    0.7200    1.0000
   15.0000    7.5000   -1.0000
    3.0000    3.0000    1.0000

    3 type(s) of capacitors will meet both the ESR and price spec.
```

```
Output Impedance vs. Frequency analysis
  1:  Use power distribution network values from the
  Unitrode paper.
  2:  Enter new power distribution network values.
  3:  Use values from preliminary board layout.
```

Choice number 3 contains the most current information about the board layout available from the customer.

```
Enter your choice for power distribution network values: -->  3

Maximum allowable output impedance is maximum allowable output
voltage variation (according to error budget calculations)
divided by maximum load current step.

Enter maximum output impedance in mid-frequency
region (ohms): -->  0.008
Enter the load current rise time (s).  (Pentium Pro is
10-15 ns): -->  13e-9

1.  Draw impedance curves.
2.  Do not draw impedance curves.
Enter your choice: -->  1
```

The impedance curves for capacitor type 3 are shown in Figure 6.2.

```
Enter the regulator switching frequency (Hz): -->  300e3
Enter Vpp of oscillator sawtooth waveform: -->  1.5
Enter the input supply voltage: -->  5
Enter ESR of inductor (ohms): -->  0.01
Enter Rdson of FETs (ohms): -->  0.01
Enter desired phase margin (positive degrees): -->  60
Select Error Amplifier Type:
```

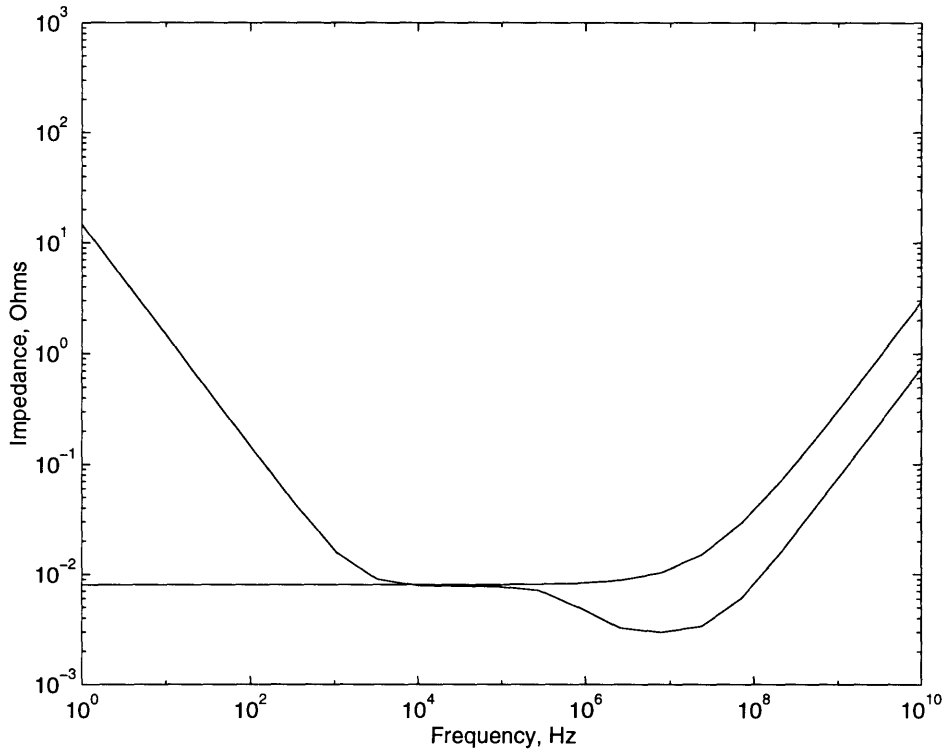


Figure 6-2: Impedance curves for the regulator using Nichicon 1800 μF capacitors.

```
  1:  Op Amp
  2:  OTA
Enter your choice: -->  2

Select Amplifier Compensation Type:
  1:  Type 2
  2:  Type 3 (for op amp only)
  3:  Series RC
  4:  Enter compensation values manually (type 2).
Enter your choice: -->  1

Enter low frequency value of gm (mmhos): -->  0.85
Enter OTA dominant pole location (Hz): -->  500e3
Enter OTA output impedance (ohms): -->  3e6
```

The OTA output impedance is set significantly higher than the minimium calculated value to give a margin for process variation.

```
    R1        R2        C1        C2        PM        Loop BW
cresults =
   1.0e+04 *
         0         0         0         0         0         0
    2.0000         0    0.0000    0.0000    0.0075    5.8187
    2.0000         0    0.0000    0.0000    0.0073    5.6650
         0         0         0         0         0         0
    2.0000         0    0.0000    0.0000    0.0045    5.6794

     The following capacitor type is recommended.
ctc =
     3
```

The values of $\alpha 1$ and $\alpha 2$ used are 1 and 0.4 respectively. If a different combination had been used to give more weight to either size or cost as a design parameter, the suggested capacitor type may change.

```
    Value     ESR       Factor    Cost
ans =
    0.0018    0.0390    2.0000    0.1200

    Number    Cost      Accept?   Req BW    Tot Cap.
ans =
   1.0e+03 *
    0.0060    0.0007    0.0010    9.8838    0.0000

       1:  Use this capacitor type.
       2:  Use a different capacitor than the one suggested.
     Enter your choice: -->  1
```

```
Enter minimum load (in Amps): -->  0
Enter maximum load (in Amps): -->  14
Enter minimum programmable output voltage: -->  1.8
Enter maximum programmable output voltage: -->  3.5

 The transconductance and output impedance of the OTA can
 vary by the following factor without becoming unstable.
tampvar =
    2.0000

   1:  Use an input filter.
   2:  Do not use an input filter.
Enter your choice: -->  1
Enter the number of capacitors used in the filter: -->  3
Enter the capacitance of each capacitor (F): -->  1800e-6
Enter the ESR of each capacitor (ohms): -->  0.039
Enter the inductor value used in the filter (H): -->  2.2e-6
Enter the ESR of the inductor plus the source resistance
(ohms): -->  0.01
   This input filter will work.
```

The plot of the output impedance of the input filter and the input impedance of the regulator is shown in Figure 6.3.

```
   The change in phase margin and cross over frequency
   with changing load current is shown below. (degrees & hz)
dpm =
    1.1069
dwc =
   2.3624e+03
```

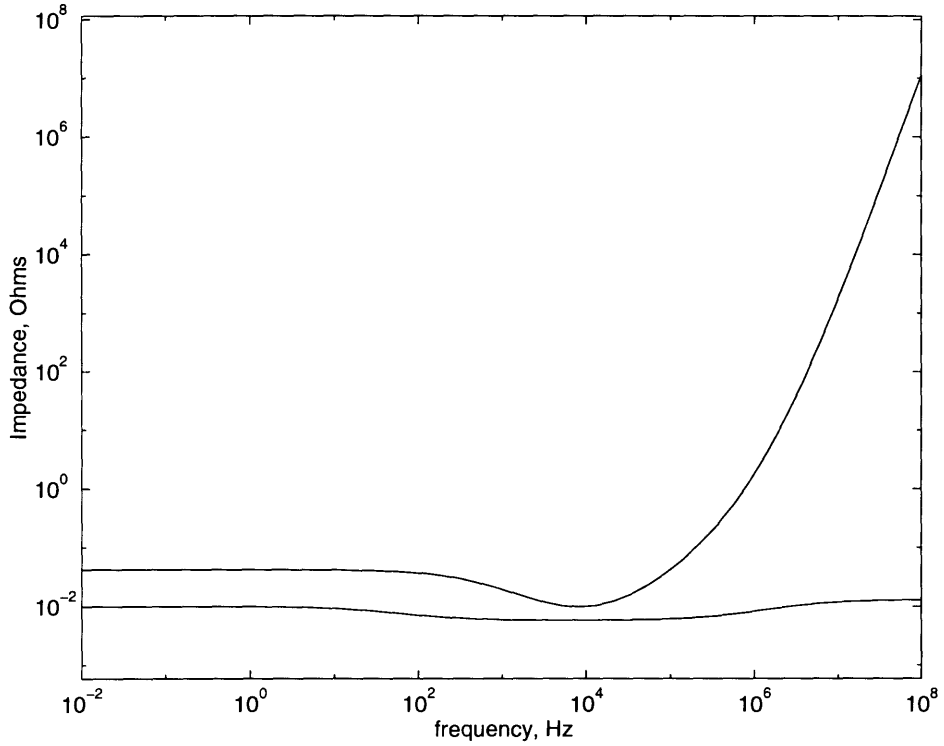


Figure 6-3: Impedance curves to determine if the proposed input filter is acceptable.

These variables `dpm` and `dwc` show that the load variation does not significantly affect the frequency characteristics of the regulator.

```
1:  Simulate regulator using these values with comparators.
2:  Simulate regulator using these values without comparators.
3:  Quit.

Enter your choice: -->  1
```

6.4 Simulation Results and Design Requirements

Running the Matlab script in the previous section gave the open loop system response shown in Figure 6.4. The compensation values are $R_c = 20$ kΩ, $C_{c1} = 22$ pF, and $C_{c2} = 2.2$ nF.

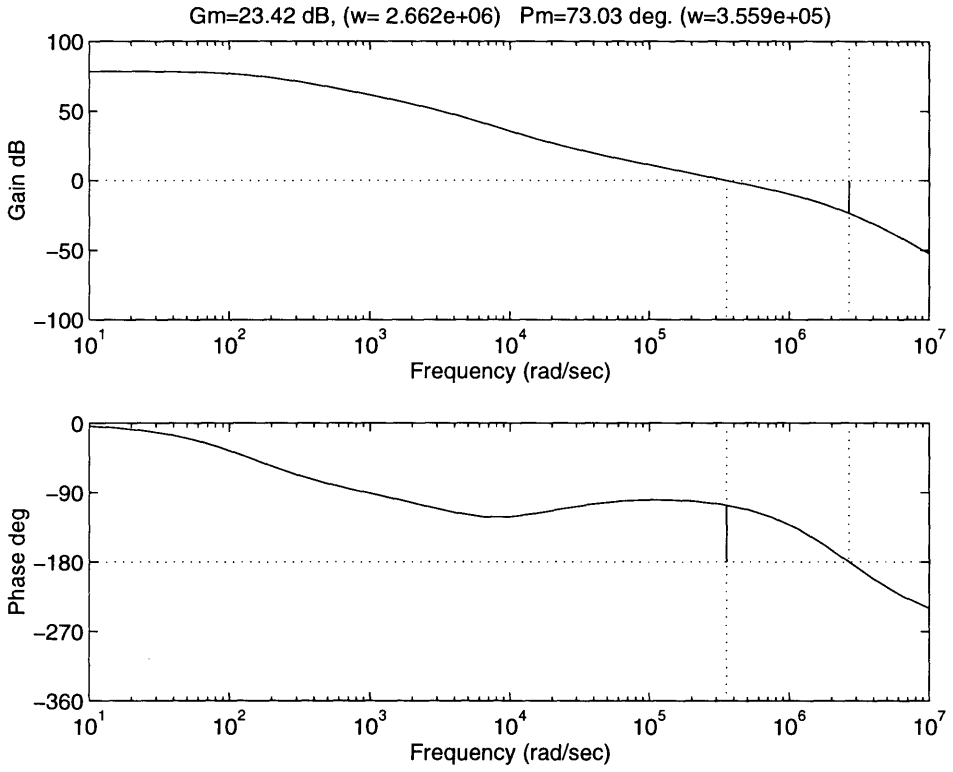


Figure 6-4: Open loop system response generated by running the Matlab script.

6.4.1 Phase Margin and Stability

From Figure 6.4, the phase margin of the system is seen to be 73°. However, this does not include the phase delay due to the PWM. This can be included using Eq. 3.3. With a switching frequency of 300 kHz, the phase margin is 34° lower than expected, or 39°. Usually the phase delay associated with the PWM is only around 10° to 15° when the loop bandwidth of the regulator is approximately 1/10 of the switching frequency. In this particular case, the loop bandwidth is 1/5 of the switching frequency. Therefore, the phase delay at crossover is higher, causing a lower phase margin than expected.

These problems can be avoided by finding new compensation values that cause the crossover to be at a lower frequency. With compensation values of $R_c = 10.5$ kΩ, $C_{c1} = 100$ pF, and $C_{c2} = 10$ nF, the nominal phase margin is 78° and the crossover frequency is 30 kHz. Once the phase delay associated with the PWM is included, the phase margin becomes 60°. The Bode plot is shown in Figure 6.5.

6.4.2 Compensation Values and Slew Rate Requirement

In Section 6.2.5, the OTA was assumed to be able to source and sink 120 μA. With this current, the time it takes to slew the output voltage of the OTA from the minimum to maximum value of the oscillator sawtooth waveform is given by

$$t = \frac{\Delta V_{oscil} \cdot C}{120\mu A} = \frac{1.5V \cdot C}{120\mu A} \tag{6.10}$$

Assuming that C_{c2} dominates, the OTA will require 125 μs to fully respond to the input change. However, during the first part of the transient response, the output voltage slew rate of the OTA is dominated by C_{c1} and is much faster than it is after C_{c2} becomes more important, thereby reducing the total transistion time between minimum and maximum duty ratio.

Figure 6.6 shows the output voltage of the OTA in the transition between minimum and maximum values of the oscillator voltage waveform with the compensation values given above. The total transition time for the output of the OTA to slew 1.5

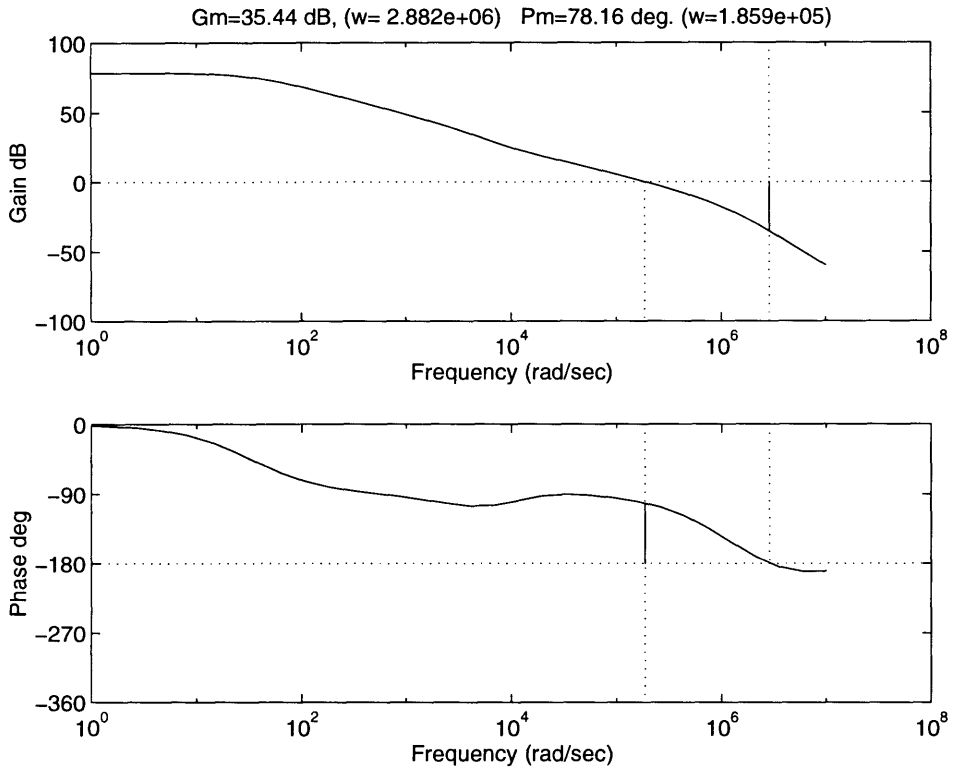


Figure 6-5: Open loop system response with improved compensation values.

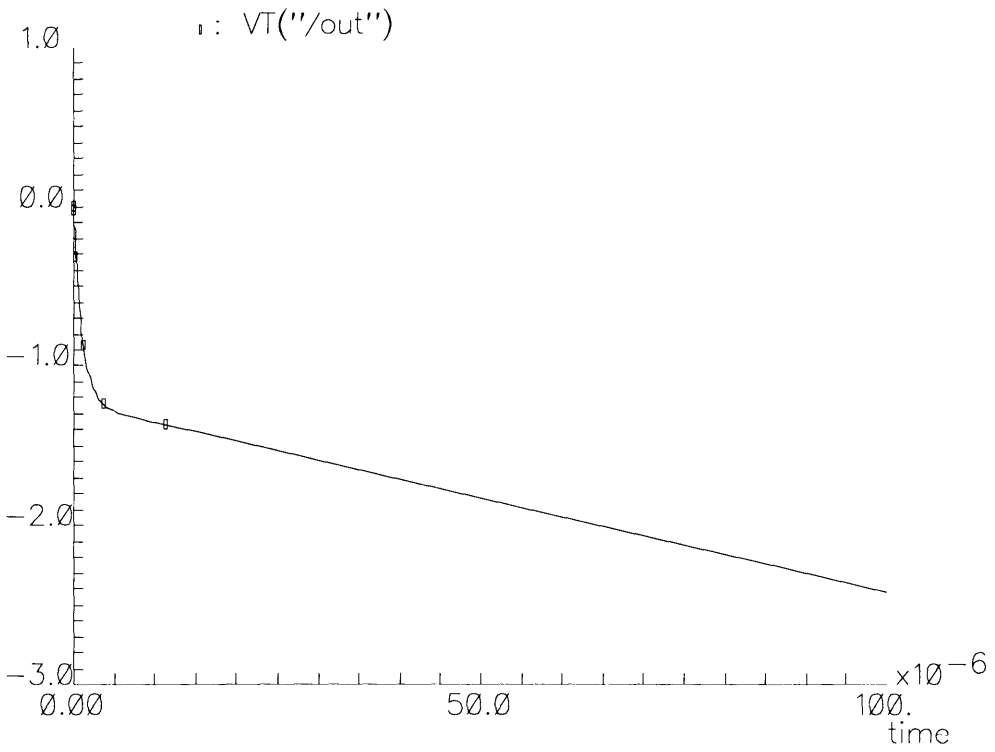


Figure 6-6: Slew rate characteristic of the OTA output voltage with $C_{c1} = 100$ pF, and $C_{c2} = 10$ nF.

V is 25 μs, which is significantly less than the 125 μs predicted above as expected. If this transition time is deemed to be too long, the compensation values can be scaled to use smaller capacitors while achieving the same phase boost.

The Matlab script has been used to design the MC33470 switching regulator and determine the necessary characteristics of the OTA. The OTA and comparator designs are presented in the following two chapters.

Chapter 7

Error Amplifier Design

7.1 Amplifier Topology and Technology Description

An operational transconductance amplifier, or OTA, provides an output current proportional to a differential input voltage. A single high impedance node, defined by the differential to single ended conversion, is compensated in this design by a two pole, single zero network referenced to ground. The performance requirements of this OTA were derived in Chapter 6 and are summarized in Table 7.1.

Folded cascode architectures have been used in the industry to provide reasonably high stage gain and high speed. However, due to the use of coarse comparators for this design to provide large load transient regulation, a high bandwidth error amplifier is not necessary. A simplified schematic of the OTA architecture used in this design is shown in Figure 7.1. This topology has the advantage of increased drain current efficiency since the current mirrors can be exploited to provide high output drive while minimizing the overall current drain of the amplifier. The output current is given as:

$$I_{out} = G_m V_{id} \tag{7.1}$$

where G_m is the effective transconductance of the stage including the g_m due to the input differential pair and the g_m due to current mirror stage's ratio, and will be

Input Stage Transconductance	$> 850\ \mu\Omega^{-1}$
Output Impedance	$> 1\ \mathrm{M}\Omega$
Input Common Mode Voltage Range	1.6 - 3.9 V
Output Drive Capability	$\pm 120\ \mu\mathrm{A}$
Supply Voltage	4.2 - 5.5 V
Output Voltage Range	1.5 - 3.0
Operating Temperature Range	$0°C - 100°C$

Table 7.1: Performance requirements of the OTA needed for the MC33470.

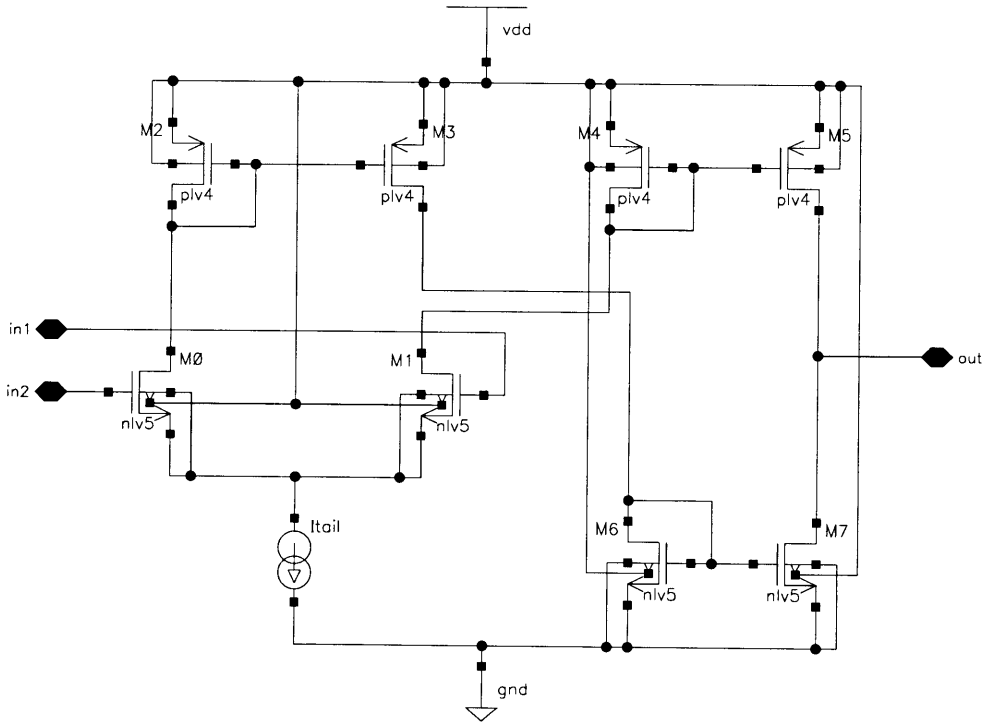


Figure 7-1: Simplified architecture of the proposed OTA.

dependent on the type of devices used, either bipolar or MOSFET. Assuming an output load resistance greater than the output impedance of the current sources that drive the high impedance gain node, the open loop gain of the OTA is therefore given as:

$$A_v = G_m R_{out} \tag{7.2}$$

Achieving high stage voltage gain is therefore driven by the ability to optimize the $G_m R_{out}$ product. The g_m of a MOS transistor, assuming strong inversion operation and V_{ds} greater than $(V_{gs} - V_T)$, is approximated by the following expression.

$$g_m = \mu C_{ox} \frac{W}{L}(V_{gs} - V_T) = \sqrt{2 I_d \mu C_{ox} \frac{W}{L}} \tag{7.3}$$

The output impedance is defined by the following relationship.

$$R_{out} = \frac{1}{\lambda I_d} \tag{7.4}$$

where λ is the channel length modulation parameter and is represented by:

$$\lambda = \frac{1}{L_{eff}} \frac{dX_d}{dV_{ds}} \tag{7.5}$$

Note that while g_m is directly proportional to the drain current of the device, the output impedance varies inversely with it. Matlab provides the designer with a specification for each that satisfies both the open loop gain requirement as well as the output impedance of the OTA. This is necessary to ensure successful interaction with the current to voltage conversion network formed by the two pole, single zero compensation network. Note also that lowering the drain current to increase the output impedance of the current mirrors could lower the current density in the input differential pair to an extent that it forces the device to operate in the moderate inversion region. In this case, Eq. 7.3 becomes invalid and Level 3 models, as will discussed in Section 7.3, fail to accurately predict transistor behavior.

The MC33470 high current switching regulator is fabricated on a SMARTMOS^{TM}[1] process with complimentary bipolar and CMOS transistors. The technology supports two layers of metal routing and two poly thicknesses for both low voltage CMOS and medium voltage power MOSFETs, including LDMOS and TMOS. The process is built using p-well tubs in a lightly doped n-epitaxial layer with an n+ buried layer above a p-substrate. Device selection includes a 5 GHz npn, a 0.8 GHz vertical pnp, a -1 V V_T enhancement mode PMOS and a +0.7 V V_T enhancement mode NMOS transistor. MOSFET sizing is based on Motorola's Universal Design Rule (UDR) system where the size of the device on silicon in microns is equivalent to a UDR times a scaling factor. The UDR convention will be used throughout the circuit design summary.

7.2 Input Differential Pair

7.2.1 Common Mode Voltage Range

The common mode range requirement of 1.6 V to 3.9 V, in combination with a requirement that the system be fully operational at a supply voltage of 4.2 V, precludes the use of an enhancement mode PMOS or pnp transistor for the input differential pair. Referring to Figure 7.2, the common mode voltage range of this architecture is given by the following relationships:

$$\begin{aligned} V_{icm(n)} &> V_{ss} + V_{gs(n)} + V_{dsat(M0)} - V_{gs(M0)} \\ &> V_{ss} + V_{gs(n)} + (V_{gs(M0)} - V_T) - V_{gs(M0)} \\ &> V_{ss} + V_{gs(n)} - V_{T(p)} \qquad (7.6) \\ V_{icm(p)} &< V_{dd} - V_{dsat(I_{tail})} - V_{gs(M0)} \qquad (7.7) \end{aligned}$$

This input stage can easily be designed to sense common mode voltages of 1.6 V or less. However, when V_{dd} = 4.2 V, assume a $V_{dsat(p)}$ of 300 mV and a $V_{gs(pM0)}$ of 1.3

[1]SMARTMOS is a registered trademark of Motorola.

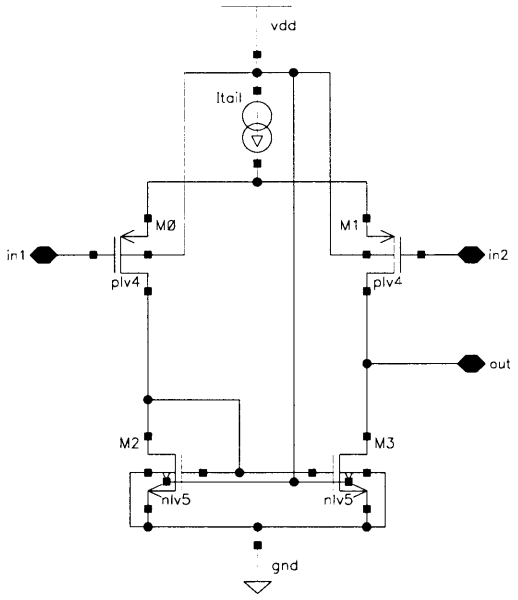


Figure 7-2: Common mode range limitations of a PMOS differential pair.

V, then Eq. 7.7 becomes:

$$V_{icm(p)} < 4.2\ V - 0.3\ V - 1.3\ V = 2.6\ V \tag{7.8}$$

This does not meet the common mode voltage requirement specified in Table 7.1 of 3.9 V. If we perform the same analysis for the OTA illustrated in Figure 7.1, we observe the following relationships:

$$V_{icm(n)} > V_{ss} + V_{dsat(I_{tail})} + V_{gs(M0)} \tag{7.9}$$

$$\begin{aligned} V_{icm(p)} &< V_{dd} - V_{gs(M2)} - V_{dsat(M0)} + V_{gs(M0)} \\ &< V_{dd} - V_{gs(M2)} - (V_{gs(M0)} - V_{T(M0)}) + V_{gs(M0)} \\ &< V_{dd} - V_{gs(M2)} + V_{T(M0)} \end{aligned} \tag{7.10}$$

If we assume a $V_{dsat(n)}$ of 300 mV and a $V_{gs(pM0)}$ of 1.0 V, then Eqs. 7.9 and 7.10 reduce to the following:

$$V_{icm(n)} > 0\ V + 0.3\ V + 1.0\ V = 1.3\ V$$

$$V_{icm(p)} < 4.2\ V - 1.0\ V + 1.0\ V = 4.2\ V$$

which meets the requirement. Equally suitable and possibly functional at lower supply voltages would be the use of an npn differential pair. However, the common mode range requirement is not the only consideration.

7.2.2 Transconductance and Input Impedance

The transconductance of a bipolar device is given as I_C/V_T, where V_T is defined as KT/q and is approximately 26 mV at room temperature. For a MOSFET differential pair biased in strong inversion, the transconductance of each device is given as $I_{tail}/(V_{gs}-V_T)$. To bias a MOSFET in strong inversion can require a $(V_{gs}-V_T)$ on the order of 500 mV. Therefore, for a given tail current bias value, bipolar input stages can provide significantly higher g_m than their MOSFET counterparts. However, one concern associated with bipolar inputs for this design is the effective input impedance that it will present to the internal, precision voltage reference. Errors associated with the input base current requirement of the OTA needs to be considered.

For most switching regulator designs, including the MC33470, the OTA monitors the output regulated voltage and compares its value with that of an internal precision voltage reference, such as a bandgap. For this design, a curvature corrected bandgap current reference is derived using a patented circuit topology that mixes both a thermal and a negative temperature coefficient current [30]. This current is then used to bias the programmable resistor string DAC which generates the internal voltage reference. Errors associated with this voltage reference can make it difficult to guarantee the $\pm 5\%$ tolerance window for the regulated output, particularly with transistor β that vary with temperature and process conditions. A MOSFET input differential pair would ensure that there would be no i_b loading of the voltage reference. The MOS OTA architecture also provides a means to increase stage gain by the use of current mirror ratioing from the input pair to the output drive circuitry. MOSFET architectures also have the added benefit of much higher integration levels than bipolar transistors.

7.3 MOSFET Modeling

Historically, MOSFET models have, to some degree, addressed the problems associated with digital CMOS designs where moderate and weak inversion operation, bulk effects, and small signal conductance are of secondary or lesser concern. However, as more analog and mixed mode ICs have turned to CMOS to capitalize on size efficiency and shrink compatability, the use of existing models have been found to be severely lacking [32, 33]. A good analog MOS model is critical to help ensure initial project success.

7.3.1 Level 3 versus SSIM

Possibly the two most important device characteristics with regard to the performance of the operational transconductance amplifier are the output impedance and moderate inversion characteristics of a MOS transistor. As has been described previously, the $G_m R_{out}$ product determines the open loop gain of the OTA, which, if incorrectly determined, could lead to severe control loop stability problems. As such, a thorough understanding of the model performance available to the designer is necessary if one is to have any faith in the results provided by the simulation tool.

PSpice supports a Level 3 MOSFET model which has been found to be sufficiently adequate for most LSI digital circuit designs. However, this model makes several highly inaccurate assumptions that can lead an analog designer astray. Of particular note is that this model relies on a discrete (i.e., if - then) algorithm which completely ignores device operation in moderate inversion, a typical bias point for low voltage, low current analog circuits. Moderate inversion refers to the region where $I_d(V_{gs})$ is neither exponential nor a first degree polynomial. Using either weak or strong inversion relationships to analyze the performance of the device with regard to g_m and drain current in this region leads to a significant error. For example, when a MOSFET is biased with a low drain to source potential and the gate bias is increased, the device moves from a region where the drain current is exponentially related to the gate bias (weak inversion or subthreshold) to a region where the drain current is

linearly dependent on V_{gs} (strong inversion). Neither of these relationships is valid for the moderate inversion region. The Level 3 MOS model treats the transition between these two regions as a step change in device performance which can appear to the designer as a sudden large drop in transconductance when the current density in the device is lowered. Likewise, a high g_m indicative of strong inversion operation may present itself to the designer even though the device is biased in the intermediate region.

The Level 6 model, or SSIM, used by MCSpice, treats the transition through the various inversion regions as a continuous phenomenon. A comparison of the input characteristic of the Level 3 and SSIM models is shown in Figure 7.3 for a low voltage, enhancement mode NMOS device measuring 80/8 UDR and biased with a V_{ds} of 100 mV. Note that for V_{gs} just above V_T, severe discrepancies exist between the two models.

The Level 3 model also suffers from an inadequate representation of MOSFET output impedance which is not limited to short channel effects alone. This can lead to a highly inaccurate prediction of the voltage gain of the OTA. Figure 7.4 illustrates a comparison of the output impedance of a 4 UDR (minimum supported by the technology) channel length. Figure 7.5 illustrates the output impedance characteristic of the same device if the channel length is increased to 12 UDR.

This modeling inadequacy presents a problem for the design of complex, high performance mixed mode ICs. PSpice adequately supports macro and behavioral modeling but cannot be used to verify transistor level circuit performance. MCSpice provides excellent analog MOS characterization, but fails to provide the designer with a robust simulation capability at the system top level. This discussion further reinforces the need to have additional simulation aids, in this case, the Matlab design methodology as presented in previous chapters, to assist the designer in providing information not readily available by other means.

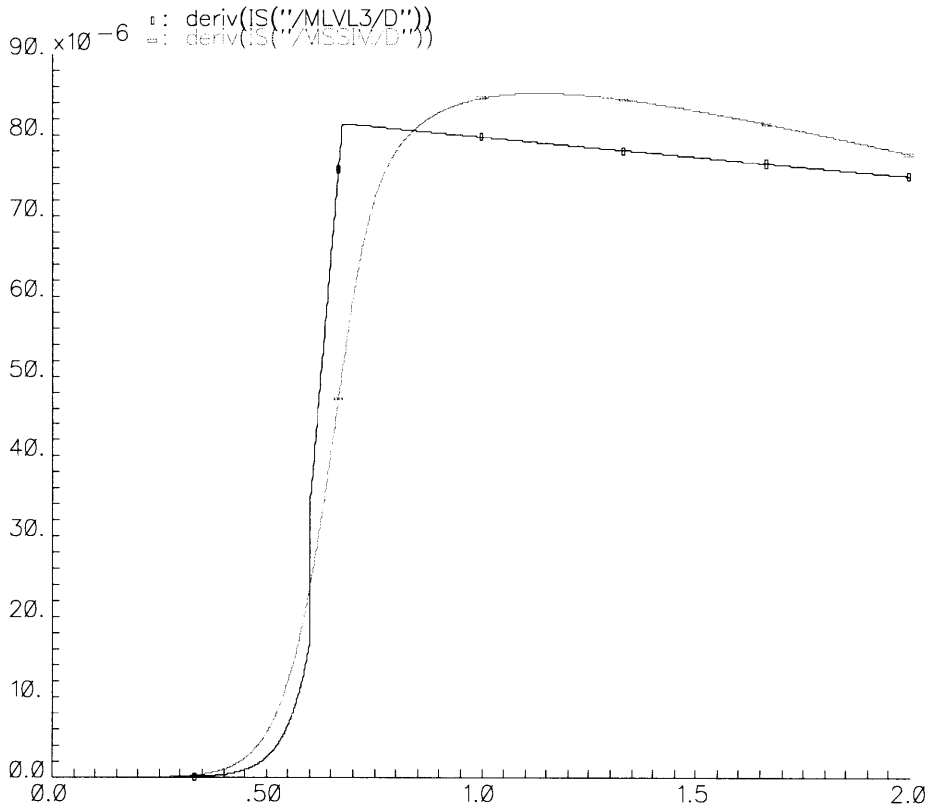


Figure 7-3: Input characteristic comparison (NMOS, V_{ds} = 100 mV): SSIM Level 6 versus Level 3.

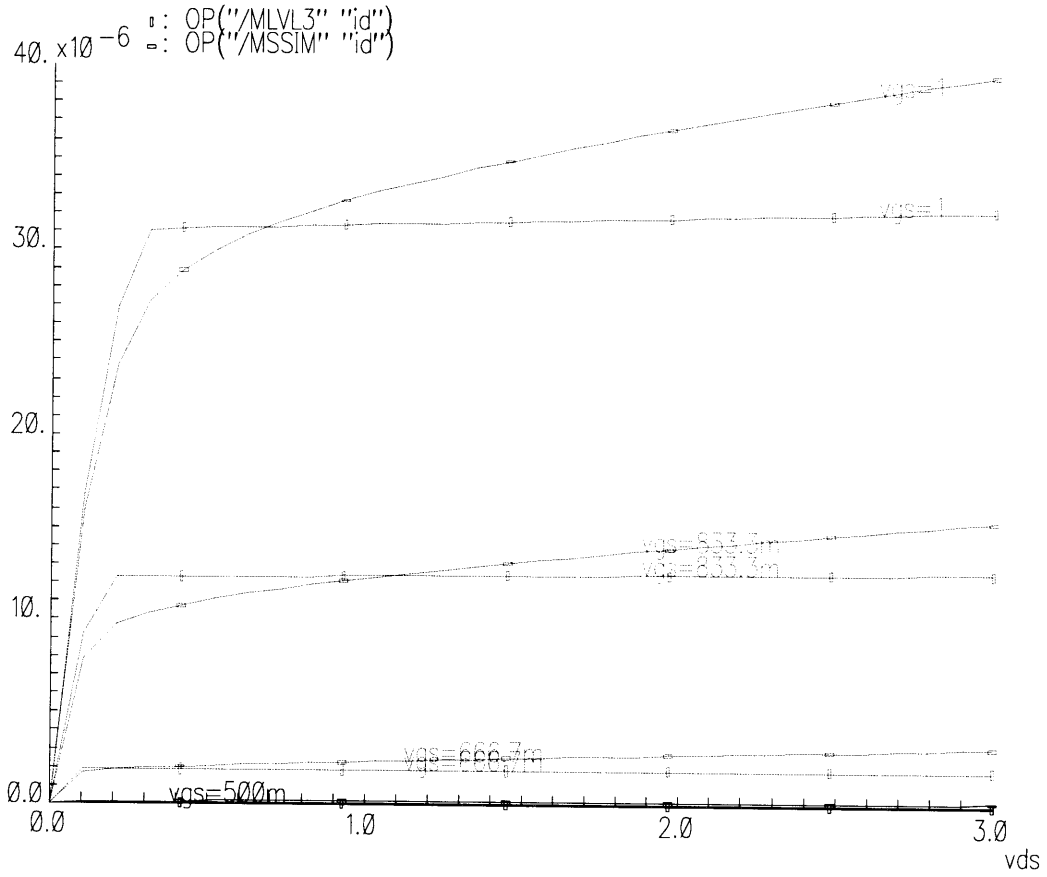


Figure 7-4: Output characteristic comparison (NMOS, L = 4 UDR): SSIM Level 6 versus Level 3.

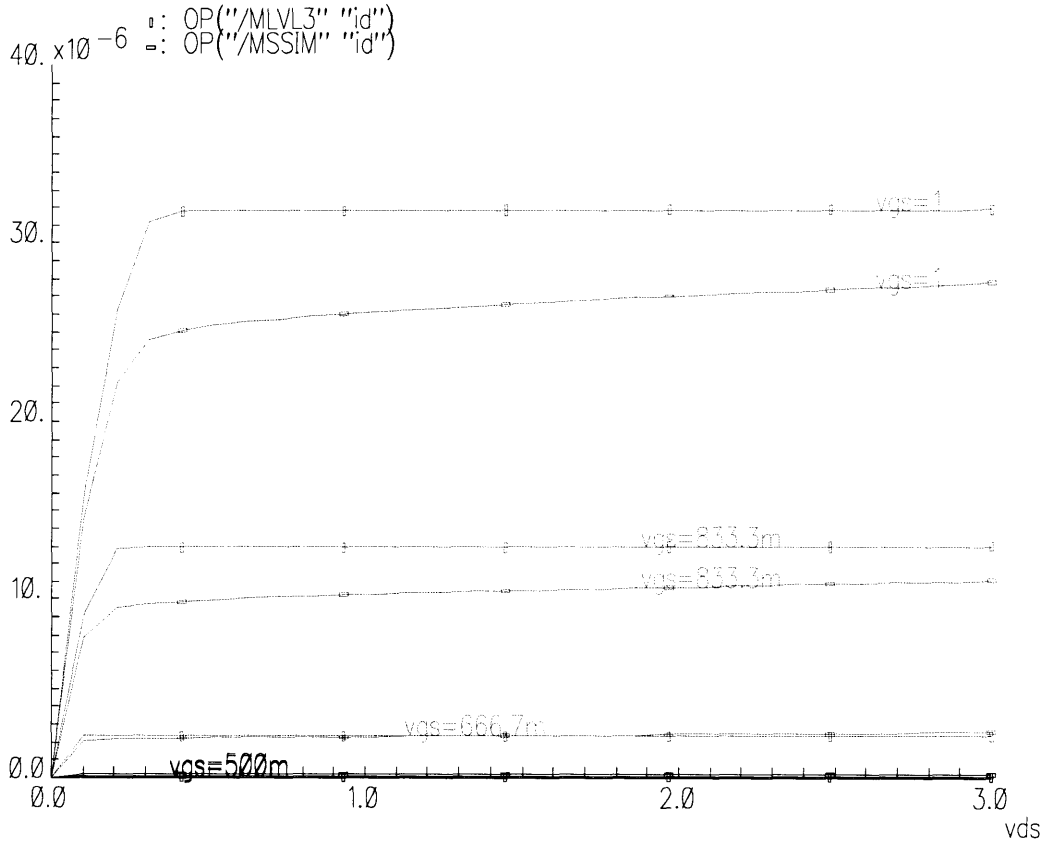


Figure 7-5: Output characteristic comparison (NMOS, L = 12 UDR): SSIM Level 6 versus Level 3.

7.4 Current Mirrors

The performance of the operational transconductance amplifier shown in Figure 7.1 relies extensively on current mirrors to provide output current drive, transconductance multiplication and high stage voltage gain defined at the output high impedance node. As mentioned in Section 7.1, consideration must be given to the tradeoffs behind establishing an optimum value for the $g_m R_{out}$ product and requires an understanding of the slew rate given a certain compensation network and the open loop gain requirement.

7.4.1 Drive Requirement and Slew Rate

The compensation network performs a current to voltage conversion, defines the frequency response, and establishes the limit on slew rate capability. Sufficient drive from the OTA must be available to change the voltage present on the compensation capacitors to ensure adequate response to a change in regulated output voltage is presented to the PWM comparator. The use of coarse comparators for this design in the control loop, as will be described in Chapter 8, greatly alleviates the need for a high performance, high bandwidth error amplifier. The principle requirement of the OTA using the topology presented is to maintain steady state voltage regulation and control loop stability.

During transient situations where the output load is insufficient to activate the coarse comparators (i.e., voltage regulation does not move outside the ±3% tolerance window), but is sufficient to divert the full effective tail current bias for charging the compensation capacitors, the slew rate is given as:

$$(\frac{dV_{out}}{dt})_{max} = \frac{I_{tail(eff)}}{C_c} \tag{7.11}$$

Therefore, assuming a ±3% transition requirement for a regulated voltage setting of 3.5 V and an effective load capacitance of 100 pF, one half of one clock cycle results

in an effective drive capability shown below.

$$\begin{aligned}
\Delta V_{out} &= 1.5\ V \\
\Delta t &= 1.5\ \mu s \\
\Rightarrow SR &= \frac{1.5\ V}{1.5\ \mu s} = \frac{1\ V}{\mu s} \\
I_{tail(eff)} &= C_c(\frac{1\ V}{\mu s}) = 100\ pF(\frac{1\ V}{\mu s}) \\
I_{tail(eff)} &= 100\ \mu A
\end{aligned}$$

The time requirement is set by the customer and reflects a margin to allow sufficient time for settling effects due to the finite bandwidth of the amplifier. The use of current mirror ratioing can be exploited in this architecture to reduce the tail current bias value to minimize amplifier drain current yet still provide the output drive capability.

7.4.2 Output Impedance and Cascoding

Simple MOS current mirrors, like the one employed in the OTA shown in Figure 7.1, suffer from poor matching due to the finite output impedance of the MOS transistor. A simple PMOS current mirror is shown in Figure 7.6. The drain of reference diode M1 is held at a V_{gs} below the supply voltage V_{dd} resulting in a fixed drain to source potential equal to $V_{gs(M1)}$. The drain of the mirror transistor M0 is connected to V_{node} whose potential varies depending on circuit architecture and transient requirements. The drain current of a MOS transistor is approximated by the following relationship:

$$I_d = \frac{\mu C_{ox}}{2}\frac{W}{L}(V_{gs} - V_t)^2(1 + \lambda V_{ds}) \tag{7.12}$$

where λ was defined in Section 7.1. When a MOSFET enters saturation, a pinch region, given as X_d in Eq. 7.5, is formed at the drain end of the transistor. As the drain to source voltage is varied, the length of this pinch region also varies and results in a modification of the effective channel length of the device. As such, it has a direct effect on the square law characteristic of the transistor. Lengthening of the channel helps to increase the output impedance by reducing the percentage of the pinch region

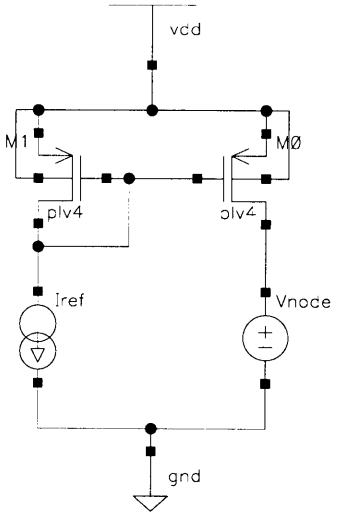


Figure 7-6: Simple PMOS mirror (L = 16 UDR).

relative to the drawn channel length.

Figure 7.7 shows the resultant mirror mismatch due to the finite output impedance for a PMOS transistor with a channel length of 16 UDR as the drain of M0 is moved from ground to within 500 mV of V_{dd}. Note that even without mismatch effects due to short channel phenomenon, errors of up to 50% are observed. Cascode devices can be used to greatly increase the output impedance of a current mirror. However, due to the fairly wide dynamic range requirement of this OTA, standard techniques of cascoding two diode connected MOSFETS would severely impact the common mode range. A cascoded mirror which preserves the common mode range capability of the simple mirror yet provides high output impedance is shown in Figure 7.8. Current mirror transistor M0 is cascoded with M13 to increase the output impedance as illustrated by the following relationship:

$$\begin{aligned} R_{o(eq)} &= r_{o(M3)} + r_{o(M0)} + g_{m(M3)} r_{o(M3)} r_{o(M0)} \\ R_{o(eq)} &= g_{m(M3)} r_{o(M3)} r_{o(M0)} \end{aligned} \tag{7.13}$$

where it is assumed that the cascoded diode comprised of M1 and M2 is assumed to be low impedance. Transistor M2 is added to the conventional Wilson current

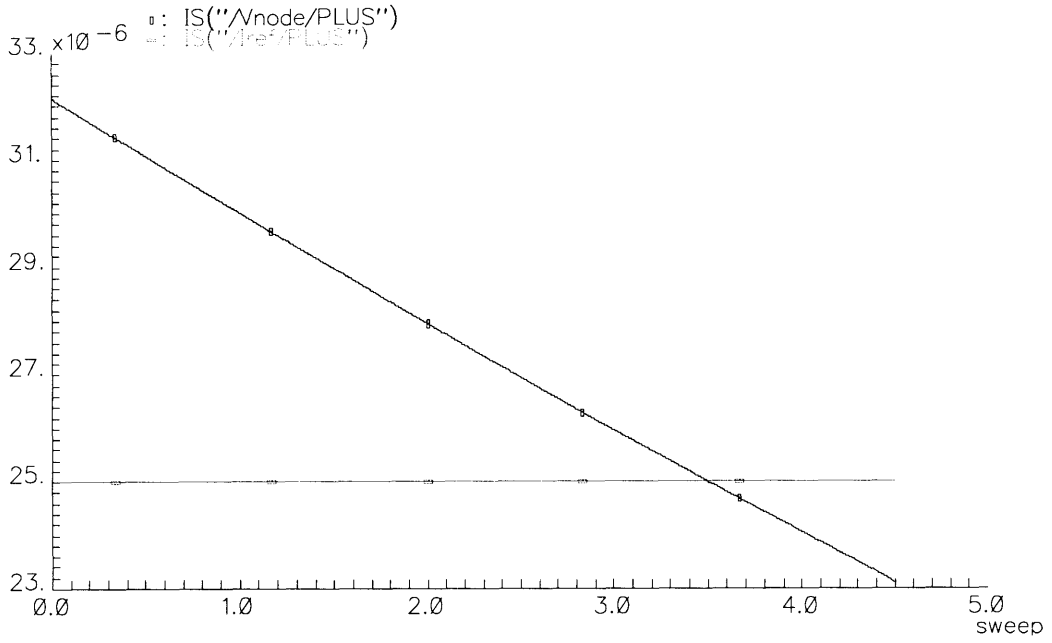


Figure 7-7: Matching of simple mirror.

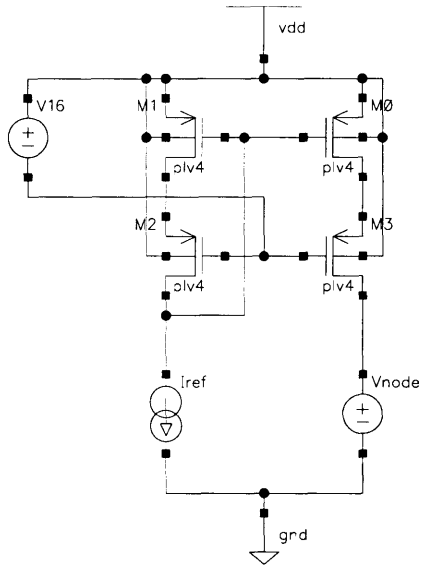


Figure 7-8: Cascoded PMOS mirror with headroom extension (L = 8 UDR).

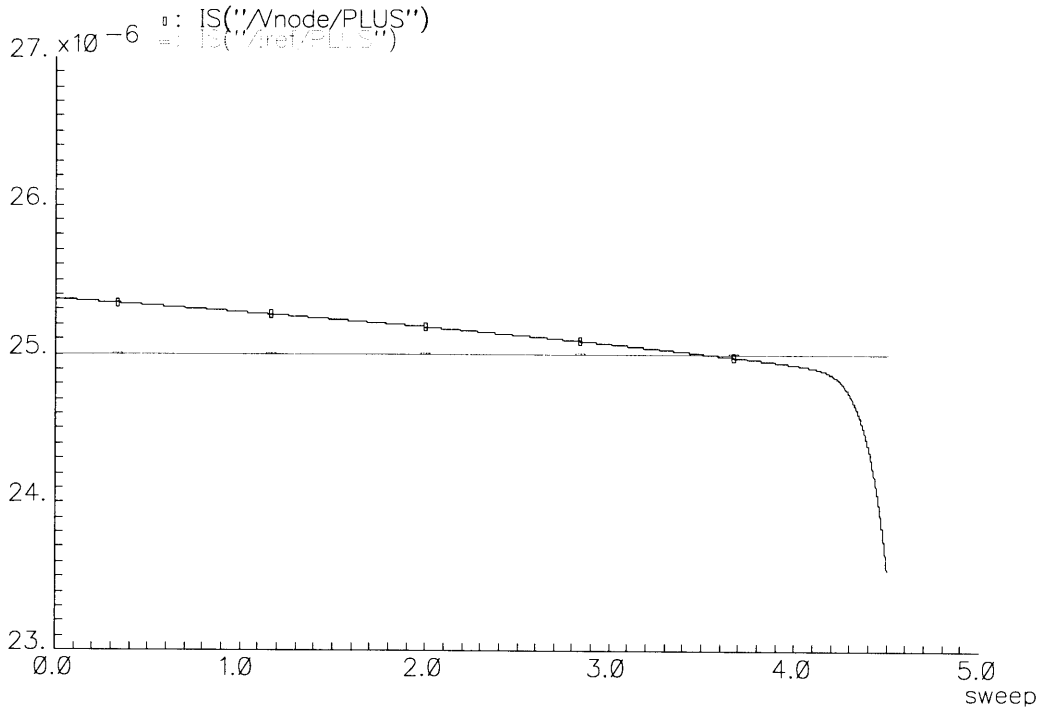


Figure 7-9: Matching of cascoded mirror.

source to balance the drain voltage of M1 with M0. A bias potential V_{bias} is applied to the common gate connection of M2 and M3 and set to a value to ensure that M0 and M1 are held in saturation. Device sizing of M1 and M2 are such that the sum of their V_{dsat} voltages is always less than the gate to source voltage of M1, that is $V_{gs(M1)} > V_{dsat(M1)} + V_{dsat(M2)}$. Figure 7.8 shows the matching of this current mirror, using 8 UDR channel length devices, as the drain of M3 is moved from ground to within 500 mV of V_{dd}. The drop in the mirrored current within 700 mV of the supply rail is due to the cascode device M3 entering the linear region. This phenomenon can be avoided by lowering the V_{dsat} of the affected devices through device sizing. In this way, a well matched current mirror can be designed with limited impact on the common mode range and yet have substantial output voltage swing.

7.4.3 Output Signal Swing

The complete OTA schematic is shown in Figure 7.10. The output voltage is required to swing from 1.5 V to 3 V to match the peak to peak voltage of the internal

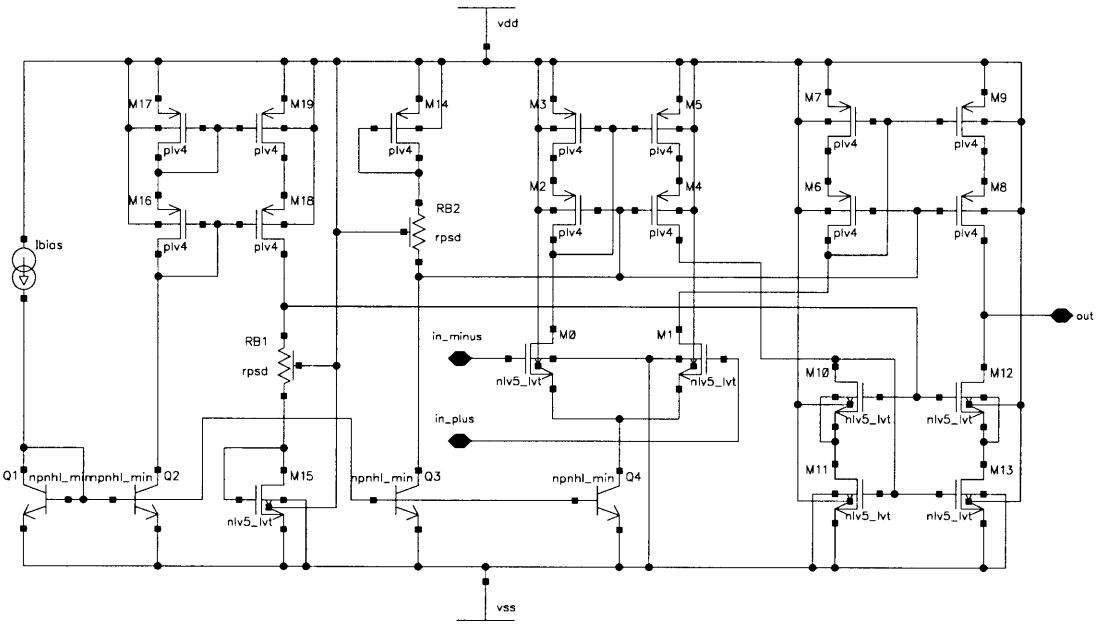


Figure 7-10: Complete OTA schematic.

oscillator. To preserve the open loop gain of the OTA through its output dynamic range, transistors M8, M9, M12 and M13 must be sized to maintain saturated region operation for a given output current. V_{dsat} is given as the difference of V_{gs} and V_T of the device. An additional safety margin is also included to ensure the transistor is operated away from the "soft saturation" condition. For an I_d of 100 μA and a V_{dsat} by design of 400 mV, the size of the NMOS device is given by:

$$\begin{aligned} I_d &= \frac{K}{2}\frac{W}{L}(V_{gs}-V_T)^2 = \frac{K}{2}\frac{W}{L}V_{dsat}^2 \\ \Rightarrow \frac{W}{L} &= \frac{2I_d}{KV_{dsat}^2} \\ \Rightarrow \frac{W}{L} &= \frac{2(100\ \mu A)}{60\ \frac{\mu A}{V^2}\ 0.4^2} = 20 \end{aligned} \tag{7.14}$$

Minimum channel length for this technology is 4 UDR. However, to provide a margin from long term threshold shifts due to hot carrier injection effects, analog MOS transistors are sized at 8 UDR. The transconductance of a PMOS transistor is approximately half that of an NMOS device and the transistor is sized accordingly.

Assuming a safety margin for V_{dsat} of 250 mV, the output dynamic range is given as follows:

$$\begin{aligned} V_{out(dyn)+} &= V_{dd(min)} - 2(V_{dsat}) - 2(V_{dssm}) \\ &= 4.5\ V - 2(0.4\ V) - 2(0.25\ V) = 3.2\ V \\ V_{out(dyn)-} &= V_{ss} + 2(V_{dsat}) + 2(V_{dssm}) \\ &= 0 + 2(0.4\ V) + 2(0.25\ V) = 1.3\ V \end{aligned}$$

which satisfies the output signal swing requirement.

7.4.4 Input Common Mode Range Enhancement

The input common mode range requirement can be further enhanced by extending the gate to source voltage of the input differential pair. This can be accomplished by increasing the intrinsic threshold voltage of transistors M0 and M1 by using the body effect. The body of transistors M0 and M1 are both connected to V_{ss}. As the gates of these transistors are moved toward the positive supply, the source will tend to follow the gate for a fixed bias current. However, as the source voltage increases, the source to body voltage increases, leading to a positive shift in the magnitude of the threshold voltage. This is represented by the following relationship:

$$V_{tact} = V_{t0} + \gamma[\sqrt{2\phi_f + V_{sb}} - \sqrt{2\phi_f}] \tag{7.15}$$

where γ is the body effect coefficient and is dependent on oxide thickness and background doping levels. For a p-well technology, γ for an NMOS device device is approximately 1.0 $\sqrt{V}$. If we assume a V_{sb} of 2 V with the gates near the positive supply, the actual threshold is given as:

$$\begin{aligned} V_{tact} &= 0.6\ V + 1[\sqrt{0.7 + 2.0} - \sqrt{0.7}] \\ V_{tact} &= 0.6 + 0.8 = 1.4\ V \end{aligned}$$

For the same bias current, $V_{dsat} = (V_{gs} - V_T)$ is unaffected. However, the source potential is now up to 2 V below the gate voltage when the gates are biased near the top of the common mode voltage range, moving the device operation far from the linear region.

7.4.5 Transconductance

The total OTA g_m requirement was determined in Chapter 6 to be greater than 850 $\mu\Omega^{-1}$. For the architecture in Figure 7.10, this is equivalent to:

$$G_m = n \cdot g_{m(M0)} \tag{7.16}$$

where n is the current mirror ratio factor. For this design a ratio of 4 was used based on meeting the output drive requirement and saving overall amplifier drain current. This results in a gm(M0) requirement of 850 $\mu\Omega^{-1}/4 = 212.5$ $\mu\Omega^{-1}$. The drive requirement as determined in Section 7.4.1 of 100 μA leads to a tail current bias value of 25 μA. The transconductance of the input stage devices, assuming strong inversion operation, is given as:

$$\begin{aligned} g_m &= \sqrt{2I_d K \frac{W}{L}} \\ \Rightarrow \frac{W}{L} &= \frac{g_m^2}{2I_d K} \end{aligned} \tag{7.17}$$

where $I_d = I_{tail}/2$. Therefore:

$$\frac{W}{L} = \frac{(212.5\ \mu\Omega^{-1})^2}{(25 \cdot 10^{-6} \cdot 60 \cdot 10^{-6})} = 30 \tag{7.18}$$

Initial simulations using the SSIM Level 6 model indicated a lower g_m than predicted due to a relatively low current density. The tail current was increased slightly, to 30 μA, and device sizing was increased to 40×.

7.5 Biasing

Tail current and cascode reference voltage biasing was implemented using npn transistors to provide high output impedance without the need for cascoding. The reference voltage for the cascoded current mirrors are provided by diode connected devices M14 and M15 and bias resistors R_{B1} and R_{B2}. Transistors M14 and M15 are sized to match the current density of cascoding devices M2, M4 and M10, M11 respectively. The voltage drop across bias resistors R_{B1} and R_{B2} maintain sufficient voltage across M3, M5, M11 and M12 to keep these devices operating in the saturated region. An example of the relationship between V_{dsat} and R_B can be represented as follows:

$$\begin{aligned}
V_{ss} + V_{gs(M15)} + V_{(RB1)} - V_{gs(M10)} - V_{dsat(M11)} &= 0 \\
0 + V_{gs(M15)} - V_{gs(M10)} &= V_{dsat(M11)} - V_{(RB1)} \\
V_{gs(M15)} &= V_{gs(M10)} \\
\Rightarrow V_{dsat(M11)} &= V_{(RB1)} \\
V_{dsat(M11)} &= \sqrt{\frac{2I_{d(M11)}}{K(W/L)_{M11}}} \\
V_{(RB1)} &= I_{d(M15)}R_{B1} \\
\Rightarrow \sqrt{\frac{2I_{d(M11)}}{K(W/L)_{M11}}} &= I_{d(M15)}R_{B1} \\
\Rightarrow R_{B1} &= \sqrt{\frac{2I_{d(M11)}}{K(W/L)_{M11}}}\frac{1}{I_{d(M15)}} \\
Let\ I_{d(M11)} &= I_{d(M15)} = I_d \\
R_{B1} &= \sqrt{\frac{2I_d}{(K(W/L)_{M11}I_d^2}} \\
R_{B1} &= \sqrt{\frac{2}{K(W/L)_{M11}I_d}} \qquad (7.19)
\end{aligned}$$

For example, with a W/L ratio of M11 of 10 biased at 30 μA, R_{B1} would be sized at 10.5 kΩ. Cascoded PMOS mirror M16 - M19 was used to provide a well controlled bias current to the voltage reference comprised of M15 and R_{B1}.

7.6 Design Summary

The open loop, uncompensated, frequency response of this OTA over the specified temperature range of 0°C to 100°C is shown in Figure 7.11. Note a nominal open loop gain of 67 dB which remains flat until approximately 200 kHz. The -20 dB/dec rolloff above the corner frequency is indicative of a single dominant pole response caused by the high impedance node formed at the drains of M8 and M12. The frequency response, with a two pole, single zero compensation network included, is shown in Figure 7.12. The peak positive phase addition of 70° occurs at approximately 8 kHz and can be adjusted, if necessary, by a simple iteration of the Matlab script demonstrated in Chapter 6. Figure 7.13 illustrates the large signal slew rate behavior of the OTA when biased from ±2.5 V supplies. Figure 7.14 demonstrates the output signal swing capability. A summary of the simulated performance, conducted using MCSpice, is listed in Table 7.2.

The layout of the OTA is shown in Figure 7.15. The input stage devices and the cascoded mirrors are cross coupled. The die size measures 0.352×0.253 mm^2 (13.8×9.96 $mils^2$).

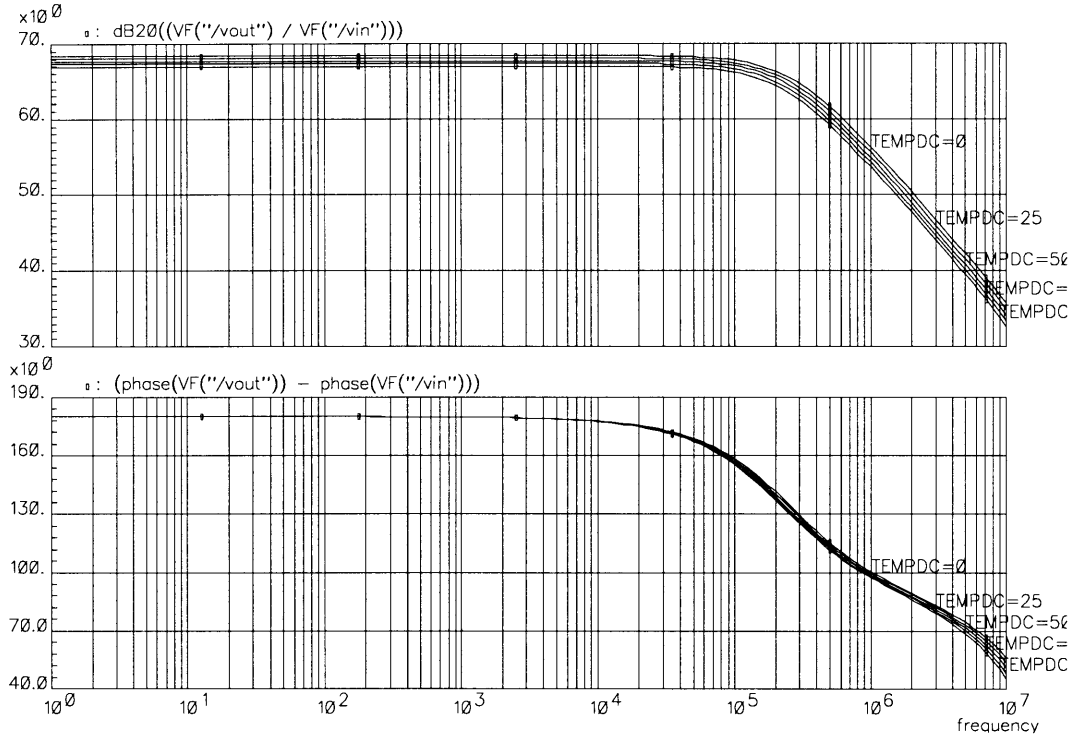


Figure 7-11: Open Loop frequency response: uncompensated.

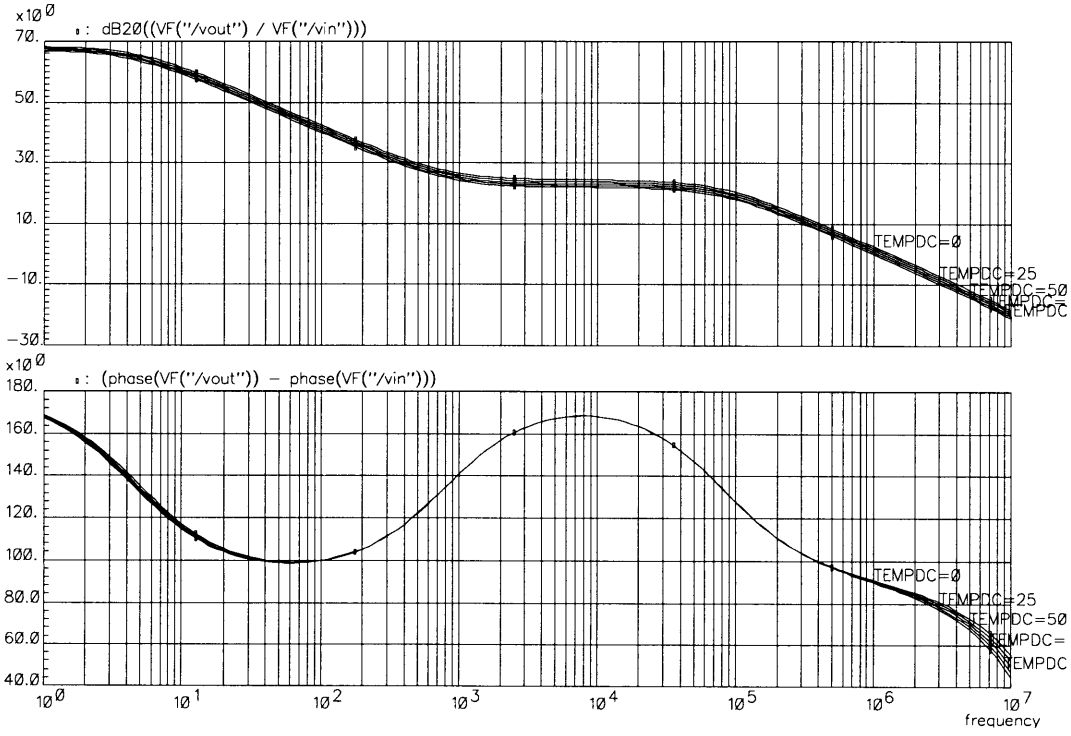


Figure 7-12: Open Loop frequency response: compensated.

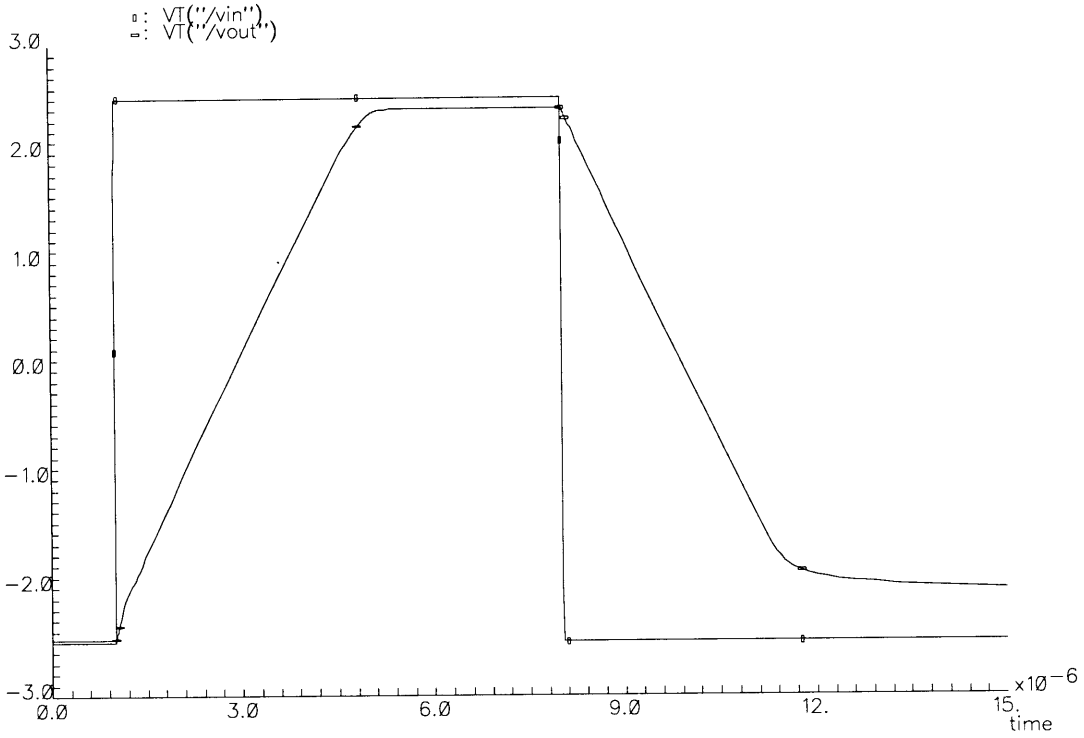


Figure 7-13: Slew rate behavior.

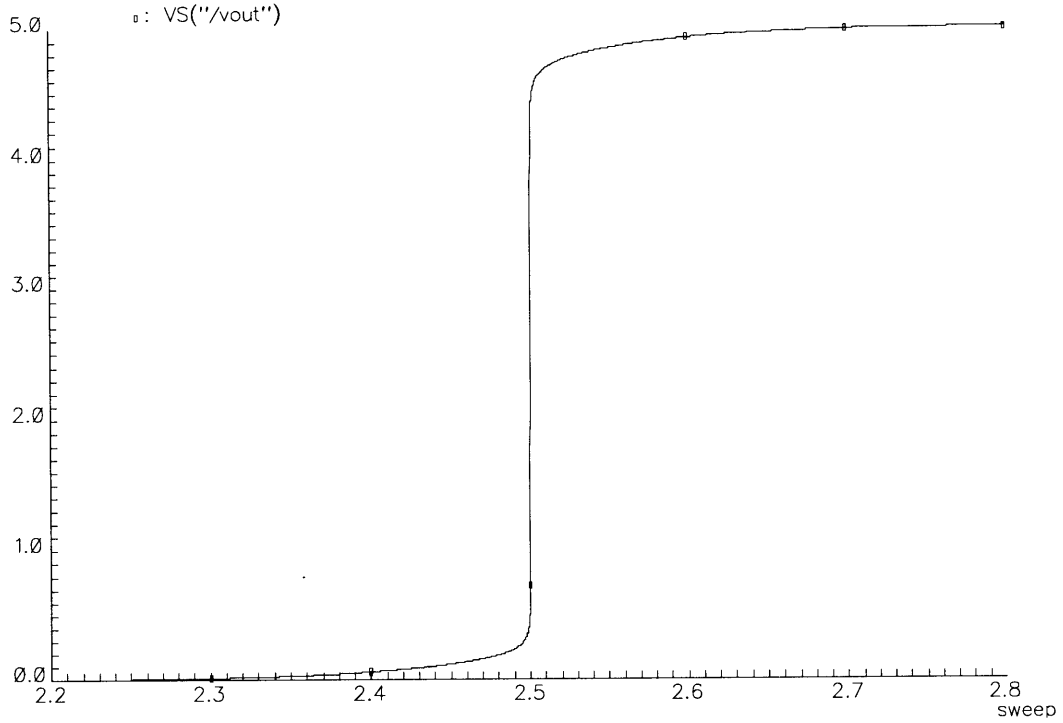


Figure 7-14: Output signal swing capability.

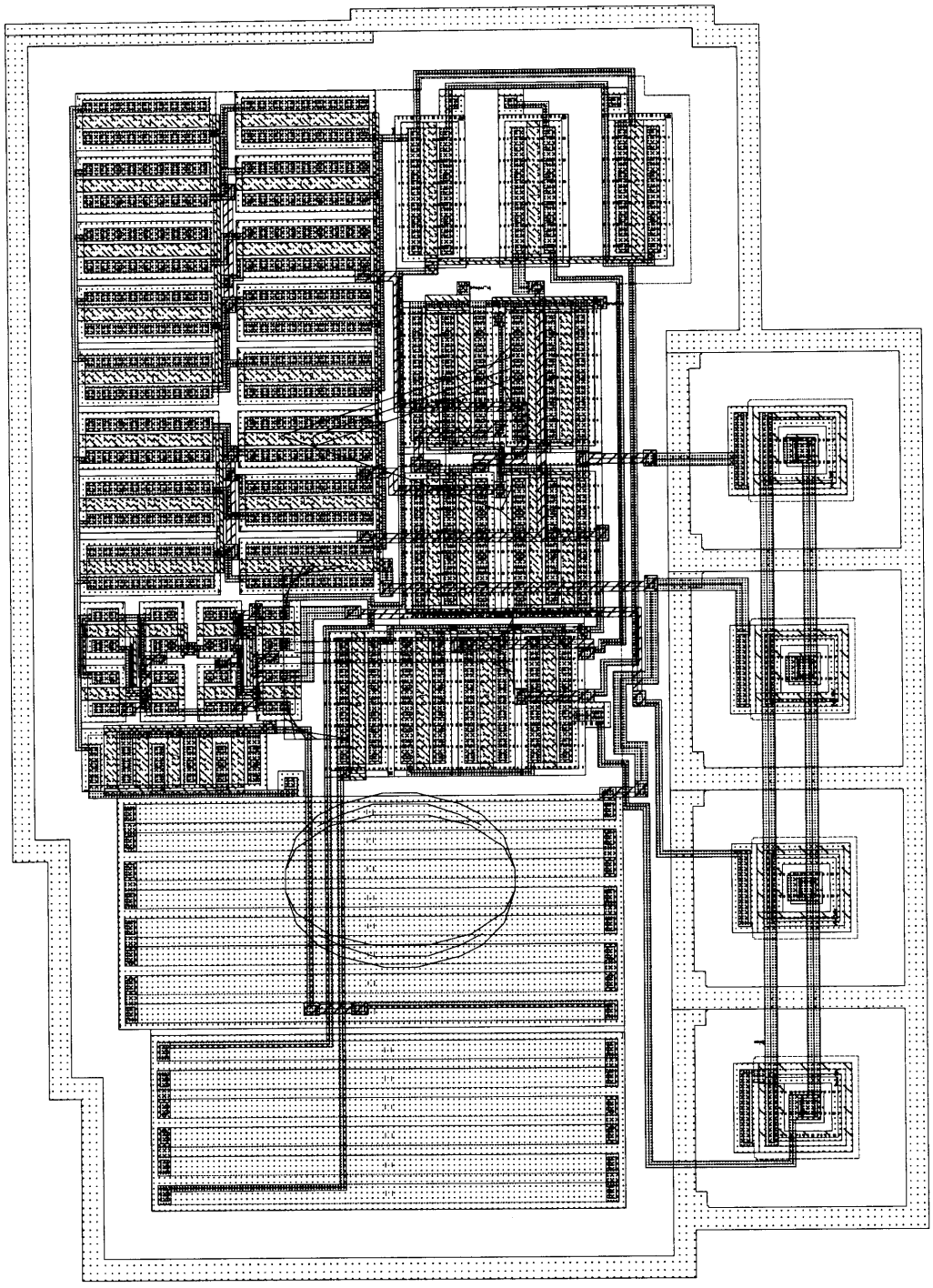


Figure 7-15: Layout of the OTA: die size = 0.352×0.253 mm^2.

Parameter	**Simulated Value**
dc Gain	67 dB
-3 dB Bandwidth	200 kHz
Output Signal Swing ($V_{dd} = 5$ V)	0.5 - 4.5 V
Input Common Mode Range	1.2 - 4.8 V
Output Drive	120 μA
Slew Rate	1.2 V/μs
Power Supply	+5 V / ground
Power Dissipation	1 mW

Table 7.2: Summary of OTA design.

Chapter 8

Coarse Comparator Design

8.1 Functional Description

The MC33470 switching regulator architecture uses coarse comparators to provide output voltage regulation during large load transient situations. A pair of high speed comparators serve to override the correction signal generated by the error amplifier during conditions when the output voltage is pulled either above or below its programmed setpoint by more than 3%. These comparators act to drive the duty cycle of the power FET driver control to either 100% or 0% depending on the direction of the load transient. This serves to simplify the design and control loop stability requirements by alleviating the need for a high bandwidth error amplifier.

8.2 Design Requirements

The common mode voltage range requirement is similar to that needed by the OTA described in Chapter 7 since all share a common sense point to the regulated output voltage. In addition, input impedance requirements follow the same guidelines as those described for the OTA and as such, the comparator input differential pair will utilize low voltage NMOS transistors.

The only load presented to these comparators are the logic gates governing PWM override control, and therefore, minimal output drive capability is needed. The input

stage transconductance is determined by analyzing the amount of differential voltage that is required to divert the entire tail current through one side of the differential pair. If we set the tail current source to 60 μA, then for a DC configuration, the drain current through each device is given as $I_{tail}/2$ or 30 μA. When the actual output voltage varies outside of the $\pm 3\%$ setpoint of the coarse comparators, the g_m of the input stage should be adequate to begin driving the control signal to the power FET drive circuitry to an on or off state. A rapid and complete transition must occur when the output voltage reaches $\pm 5\%$ of the programmed level. The limiting case exists when the selected regulated voltage is at its minimum value, which for this design corresponds to 1.8 V. The transconductance is then determined as follows:

$$\begin{aligned} \Delta V &= 0.05 \cdot 1.8\ V = 90\ mV \\ \Delta I &= 30\ \mu A \\ \Rightarrow g_{m(M0)} &= \frac{\Delta I}{\Delta V} = \frac{30\ \mu A}{90\ mV} = 333\ \mu\Omega^{-1} \end{aligned} \tag{8.1}$$

using Eq. 7.17,

$$\begin{aligned} \frac{W}{L} &= \frac{g_m^2}{2I_d K} \\ &= \frac{(333 \cdot 10^{-6})^2}{(60 \cdot 10^{-6} \cdot 60 \cdot 10^{-6})} \\ &= 30.8 \end{aligned} \tag{8.2}$$

To account for temperature and process variations, as well as effects due to moderate inversion operation observed during simulations using MCSpice, actual input stage device sizing was increased to 40×.

8.3 Comparator Architecture

The comparator architecture used in this design is shown in Figure 8.1. It is comprised of a n-channel differential pair, M0 and M1, whose bulk terminal is connected to V_{ss} to enhance common mode voltage range as the inputs are driven towards the

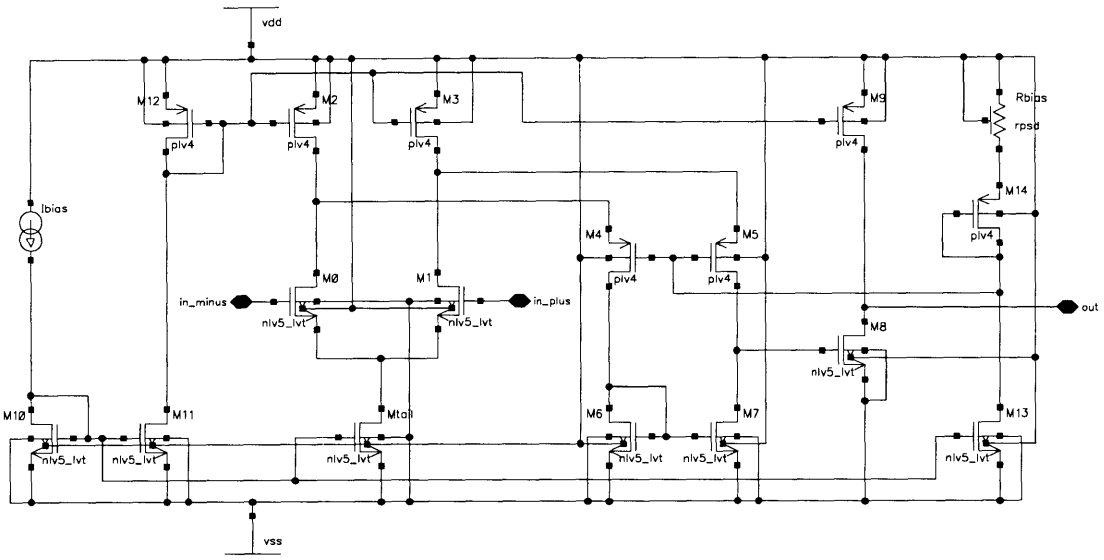


Figure 8-1: Comparator schematic.

positive supply. The folded cascode employs PMOS current source loads, M2 and M3, common gate PMOS transistors, M4 and M5, and a differential to single ended conversion circuit comprised of NMOS transistors M6 and M7. A simple class A output stage is used to provide wide output dynamic range and consists of PMOS current source M9 driving an NMOS inverter M8. Biasing for the common gate transistors is provided by diode connected PMOS device M14 and implant resistor R_{bias}. The voltage across R_{bias}, given as $I_{d(M13)}R_{bias}$ is set to ensure that active load devices M2 and M3 remain in saturation. To preserve full transconductance of the input stage, the output impedance must be kept high impedance relative to the source impedance of M4 and M5. Operation of M2 and M3 out of saturation would result in a loss of gain as some of the input signal would be shunted to small signal ground.

Device sizing and drain current bias value selection in output stage drive transistor M8 is determined such that the gate to source voltage of M8 ($V_{gs(M8)}$) is approximately equal to the gate to source voltage of M6 ($V_{gs(M6)}$). This helps to maintain the drain to source voltage of M6 ($V_{ds(M6)}$) at roughly the same value as the drain to source voltage of M7 ($V_{ds(M7)}$) for static operating situations. This helps to maintain the differential to single ended conversion network, comprised of simple mirror M6 and

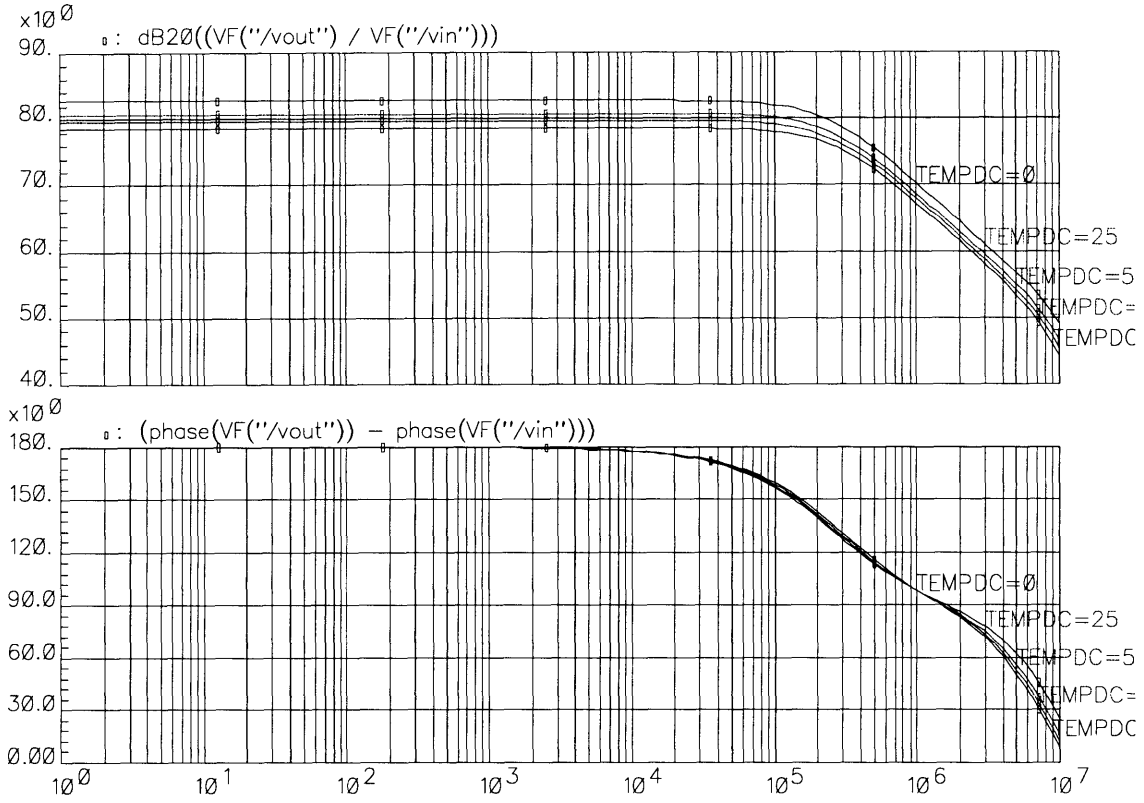


Figure 8-2: Open loop frequency response.

M7, in a balanced condition. Any DC imbalance in this mirror results in uneven bias currents in the folded cascode (i.e., $I_{d(M6)} \neq I_{d(M7)}$) and contributes to the amount of input referred offset voltage of the comparator. In addition, the channel lengths of the NMOS mirror transistors were set to 12 UDR to increase device output impedance.

8.4 Simulation Results

The open loop frequency response of the comparator is shown in Figure 8.2 for a specified temperature range of $0°C$ to $100°C$. The comparator exhibits a flat gain of approximately 82 dB out to a -3 dB frequency of 200 kHz. A characteristic single pole gain attenuation due to the first gain stage is observed. A second pole contributed by the second gain stage occurs above 10 MHz.

Figure 8.3 is the slew rate performance of the comparator when driving a 5 pF

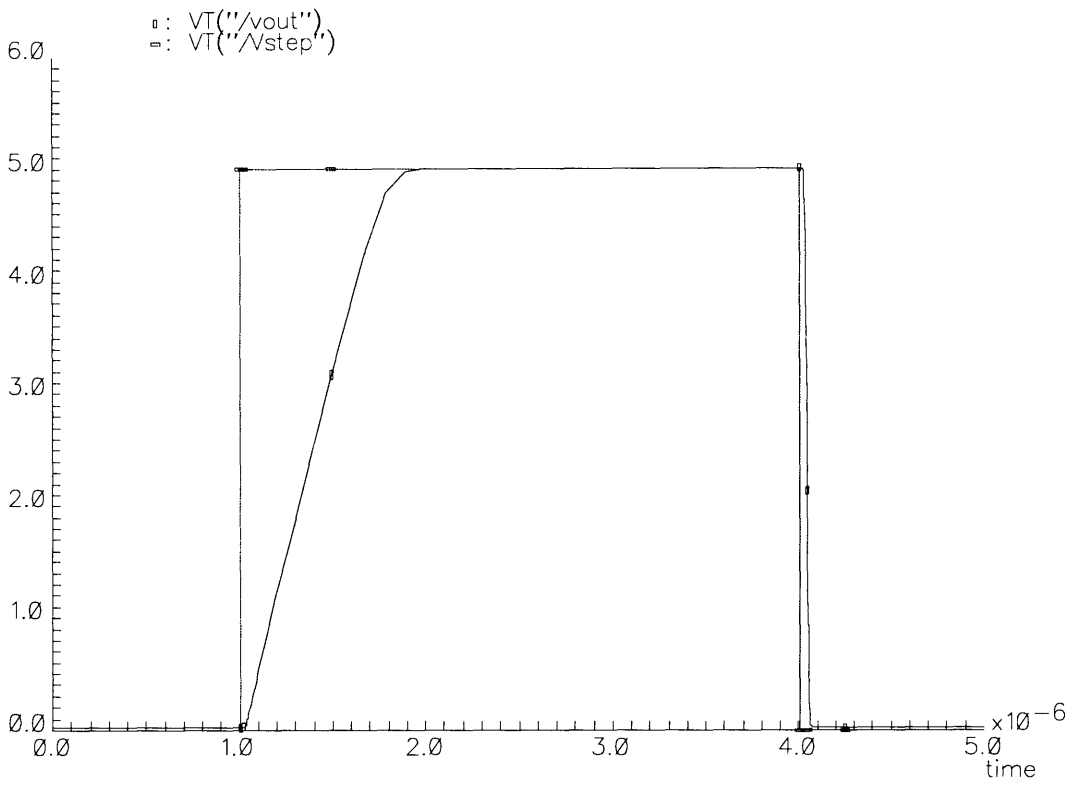


Figure 8-3: Slew rate behavior: C_{load}= 5 pF.

load and biased with a single 5 volt supply. The observed assymetrical behavior is due to the simple class A output stage. During evolutions which require charging of the output load capacitance, NMOS transistor M8 is turned off, leaving fixed PMOS current source M9 to provide sourcing current capability. This limits the positive slew rate to:

$$(\frac{dV_{out}}{dt})_{max(+)} = \frac{I_{d(M9)}}{C_{load}} = \frac{60 \cdot 10^{-6}}{5 \cdot 10^{-12}} = 12\frac{V}{\mu s} \tag{8.3}$$

assuming infinite output impedance of the PMOS current source. Alternatively, removing charge is accomplished by turning on M8 and pulling charge out of the load capacitor. Assuming a $(V_{gs} - V_t)$ of 3 V when the input pair receives a large negative differential signal, then M8 will try to sink a drain current equivalent to:

$$\begin{aligned} I_{d(M8)} &= \frac{K}{2}\frac{W}{L}(V_{gs} - V_t)^2 \\ &= 30 \cdot 10^{-6}(5)(3)^2 = 1.35\ mA \end{aligned} \tag{8.4}$$

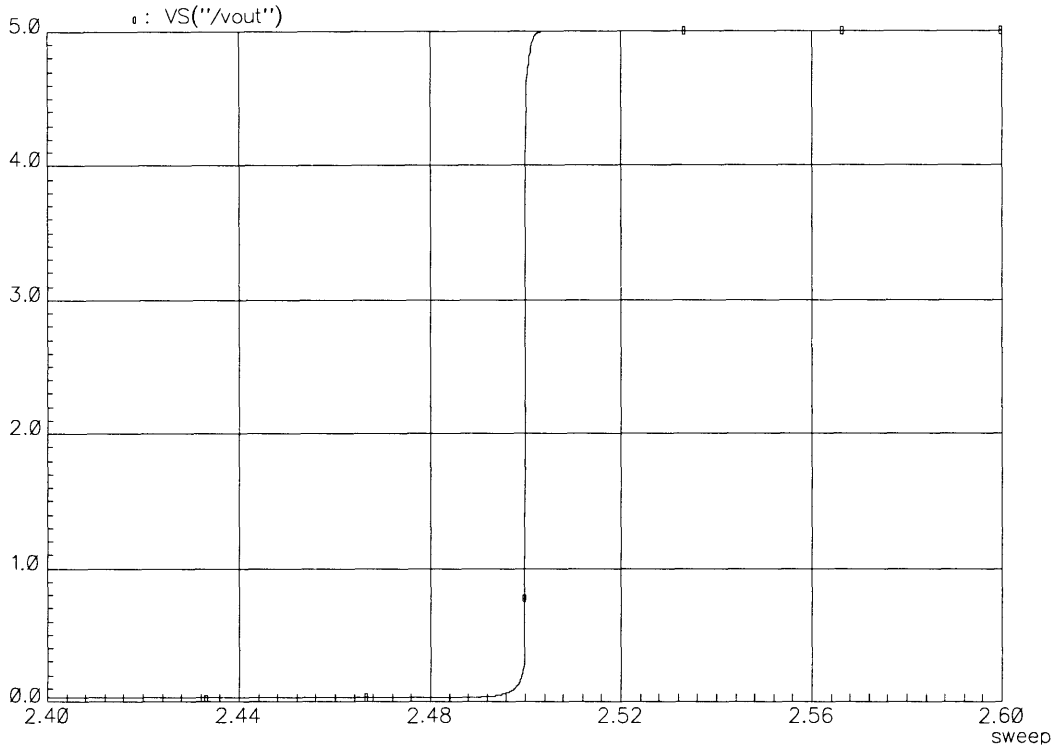


Figure 8-4: Output signal swing capability.

Fixed current source M9 is providing only 60 μA of bias current and as such, M8 will attempt to sink a large amount of charge from the load capacitor leading to a significant negative slew rate capability, as illustrated in Figure 8.3.

The output swing capability of the comparator is shown in Figure 8.4. The MOS class A output stage provides a rail-to-rail signal swing compatible with digital input levels.

The layout of the dual, plus and minus comparators, showing a common biasing configuration, is shown in Figure 8.5. The input stage devices are common centroid and cross coupled to minimize the intrinsic offset. Die size for the dual comparator measures 0.382×0.2 mm^2 (15×8 mils^2).

A summary of the performance features are given in Table 8.1.

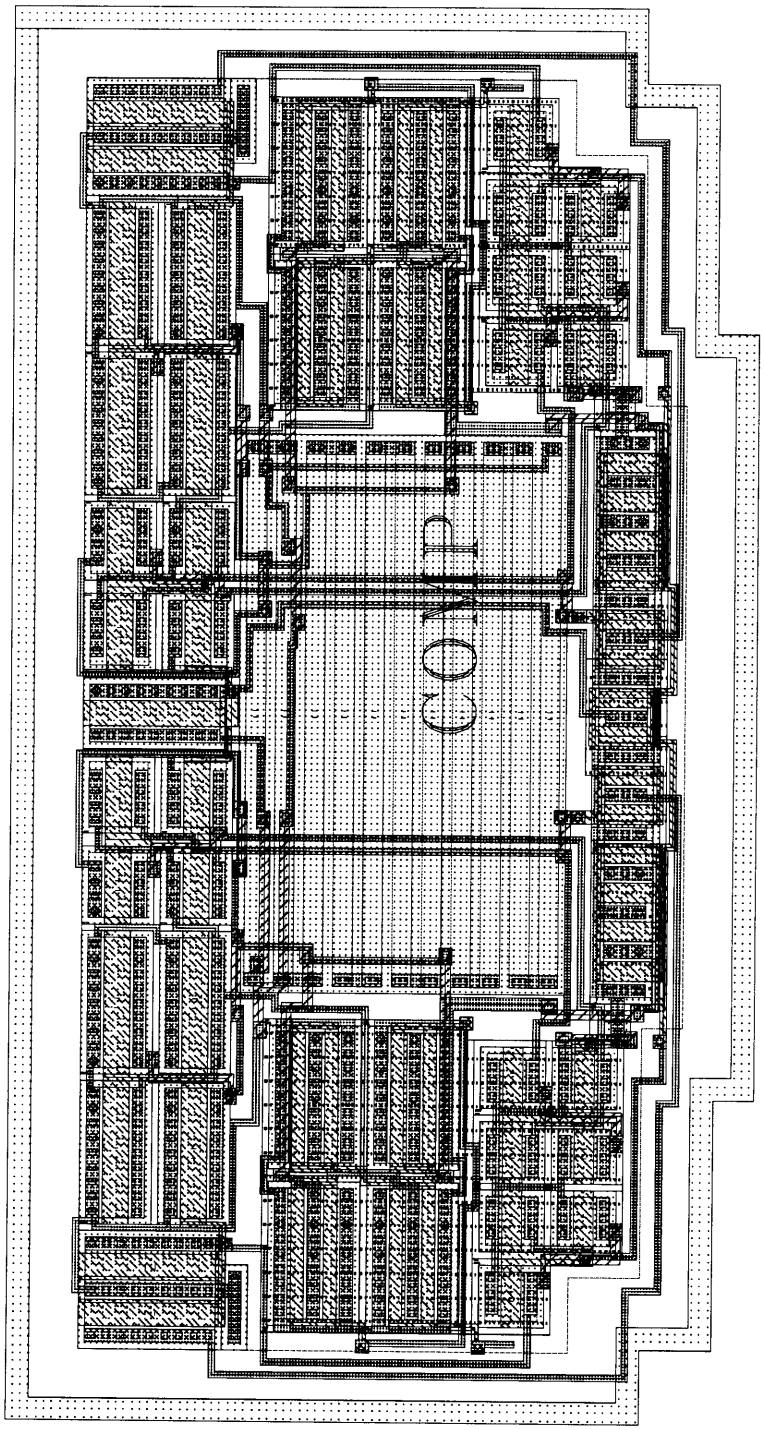


Figure 8-5: Layout of dual comparators: die size = 0.382×0.2 mm^2.

Parameter	**Simulated Value**
dc Gain	82 dB
-3 dB Bandwidth	200 kHz
Output Signal Swing ($V_{dd} = 5$ V)	0 - 5 V
Input Common Mode Range	1.2 - 4.8 V
Slew Rate (+) $C_{load} = 5$ pF	12 V/μsec
Slew Rate (-) $C_{load} = 5$ pF	> 100 V/μsec
Power Supply	+5 V / ground
Power Dissipation (Dual Comparators)	2.5 mW
Operating Temperature Range	$0°C - 100°C$

Table 8.1: Summary of comparator design parameters.

Chapter 9

Conclusion

9.1 Summary

In Chapter 3, a basic model of the buck switching regulator was created using Matlab. This model was used to determine the frequency characteristics and transient response of the regulator. Chapter 4 developed a more detailed regulator model that accounts for parasitic component values and second order effects. A Matlab script was also written that aids in the regulator design process by giving the designer an intuitive feel for how parameter variations affect the performance of the regulator. In Chapter 5, the Matlab script and model were verified using PSpice and three breadboards built with Linear Technology's LTC1430 regulator IC. Then the Matlab script was used in Chapter 6 to aid in the design of the MC33470. Chapters 7 and 8 summarized the design of the operational transconductance amplifier and comparators, respectively.

9.2 Conclusions

The Matlab design methodology avoids many of the typical simulation problems associated with using PSpice. For example, the number of memory intensive PSpice simulations that are needed to complete a regulator design is reduced. Also, it allows the designer to quickly determine the robustness of the regulator by varying parameters and observing the resulting phase margin of the system. Finally, this design

method allows the project goals of a low cost design, a robust regulator system, and a quick design cycle time to be met.

9.3 Future Work

The techniques presented in this thesis can be extended in several ways. First, regulator topologies such as boost and flyback can also be modeled in the frequency domain to allow the phase margin and required compensation to be quickly determined for a set of design parameters. The Matlab script presented could then be expanded to facilitate the design of these regulators. Also, Matlab could be used to perform transient simulations using a set of nonlinear differential equations that describe the regulator. These simulations are generally much less time consuming than similar PSpice simulations while still providing excellent accuracy. Although the equations describing the regulator can be complicated, they could be incorporated directly into a script so that the design is interactive and no prior knowledge of how to use Matlab would be necessary.

Appendix A

Matlab Script

```
clc
clear
disp(' ')
disp([blanks(5) 'Switching Regulator Design and Optimization'])
disp(' ')
l=input([blanks(5) 'Enter the output filter inductor value
(uH): -->  ']);
l=l*1e-6;
disp(' ')
ntimes=0;

%cap types [value, esr, size, cost]
%each row indicates cap value, esr, performance indicator (taking
%size into account), and cost.)

disp([blanks(5) 'Capacitor data is currently as shown below.'])
disp(' ')

disp('    Value     ESR       Factor    Cost')
ctype=[330e-6 0.03 1 1; 15000e-6 0.020 5 0.12; 1800e-6 0.039 2
0.12; 330e-6 0.100 1 0.5; 330e-6 0.018 1 1]

cchange=0;
while cchange==0,
     disp([blanks(5) '  1:  Use this data.'])
     disp([blanks(5) '  2:  Enter data for a new capacitor.'])
     disp([blanks(5) '  3:  Delete data for a capacitor.'])
     disp('  ')
```

```
     ch0=input([blanks(5) 'Enter your choice: -->  ']);

if ch0==3,
     row=input([blanks(5) 'Enter row to delete: -->  ']);
     ctype1=ctype(1:row-1,:);
     ctype2=ctype(row+1:length(ctype(:,1)),:);
     disp(' ')
     disp('    Value     ESR       Factor    Cost')
     ctype=[ctype1;ctype2]
end

if ch0==2,
     ctype(length(ctype(:,1))+1,1)=input([blanks(5) 'Enter new
     capacitor value (F): -->  ']);
     ctype(length(ctype(:,1)),2)=input([blanks(5) 'Enter new
     capacitor ESR (ohms): -->  ']);
     ctype(length(ctype(:,1)),3)=input([blanks(5) 'Enter new
     capacitor size factor: -->  ']);
     ctype(length(ctype(:,1)),4)=input([blanks(5) 'Enter new
     capacitor price ($): -->  ']);
     disp(' ')
     disp('    Value     ESR       Factor    Cost')
     ctype
end

if ch0==1,
     cchange=1;
end

if ch0~=1,
     disp('  ')
     disp([blanks(5) '  0:  Continue editing capacitor data.'])
     disp([blanks(5) '  1:  Quit editing data.'])
     cchange=input([blanks(5) 'Enter your choice: -->  ']);
     disp(' ')
end

end

maxesr=input([blanks(5) 'Enter maximum ESR for acceptable
transient response (ohms): -->  ']);
maxcost=input([blanks(5) 'Enter maximum allowable cost of
filter capacitors ($): -->  ']);
disp(' ')
```

```
for loop1=1:1:length(ctype(:,1)),
     cdesign(loop1,1)=ceil(ctype(loop1,2)/maxesr);
     cdesign(loop1,2)=ctype(loop1,4)*cdesign(loop1,1);
     cdesign(loop1,3)=sign(maxcost+0.001-cdesign(loop1,2));
end

gcaps=0;

for loop2=1:1:length(ctype(:,1)),
     if cdesign(loop2,3)==1,
          gcaps=gcaps+1;
     end
end

disp(' ')
disp('    Number    Cost      Acceptable')
cdesign
gcaps=num2str(gcaps);

disp([blanks(5) gcaps ' type(s) of capacitors will meet
both the ESR and price spec.'] )

%  Output Impedance vs. Frequency analysis
disp(' ')
disp([blanks(5), 'Output Impedance vs. Frequency analysis'])
disp([blanks(5), 'to determine necessary bandwidth of
regulator loop'])
disp(' ')
disp([blanks(5) '  1:  Use power distribution network values
from the Unitrode paper.'])
disp([blanks(5) '  2:  Enter new power distribution
network values.'])
disp([blanks(5) '  3:  Use values from preliminary board layout.'])
disp('  ')
ch1=input([blanks(5) 'Enter your choice for power distribution
network values: -->  ']);

if ch1==1,
     lco=1.25e-9;
     cb=30e-6;
     rcb=4e-3;
     lcb=63e-12;
     rs=3.6e-3;
     lc=1e-9;
     lb=.5e-9;
```

```
end

if ch1==2,
     lco=input([blanks(5) 'Enter Lco value (H): -->  ']);
     rcb=input([blanks(5) 'Enter Rcb value (ohms): -->  ']);
     cb=input([blanks(5) 'Enter Cb value (F): -->  ']);
     lcb=input([blanks(5) 'Enter Lcb value (H): -->  ']);
     rs=input([blanks(5) 'Enter Rs value (in ohms): -->  ']);
     lc=input([blanks(5) 'Enter Lc value (H): -->  ']);
     lb=input([blanks(5) 'Enter Lb value (H): -->  ']);
end

if ch1==3,
     lco=0.34e-9;
     cb=40e-6;
     rcb=3e-3;
     lcb=12e-12;
     rs=1.29e-3;
     lc=.26e-9;
     lb=.5e-9;
end

disp(' ')
disp([blanks(5) 'Maximum allowable output impedance is
maximum allowable output'])
disp([blanks(5) 'voltage variation (according to error
budget calculations) divided by'])
disp([blanks(5) 'maximum load current step.'])
disp(' ')
imp=input([blanks(5) 'Enter maximum output impedance in
mid-frequency region (ohms): -->  ']);

tr=input([blanks(5) 'Enter the load current rise time (s).
(Pentium Pro is 10-15 ns): -->  ']);
fc=0.35/tr;

disp(' ')
disp([blanks(5) '1.  Draw impedance curves.'])
disp([blanks(5) '2.  Do not draw impedance curves.'])
disp(' ')
curves=input([blanks(5) 'Enter your choice: -->  ']);

for loop3=1:1:length(ctype(:,1)),
     if cdesign(loop3,3)==1,
     co=cdesign(loop3,1)*ctype(loop3,1);
```

```
     rco=ctype(loop3,2)/cdesign(loop3,1);

num=[co*cb*lcb*(lc+lb+lco) (lc+lb+lco)*rcb*co*cb
+lcb*co*cb*(rco+rs) lcb*cb+co*(lc+lb+lco)+rcb*co*(rco+rs)*cb
(rco+rs)*co+rcb*cb 1];
den=[cb*co*(lc+lb+lcb+lco) cb*co*(rco+rcb+rs) (co+cb) 0];
temp=[1e-12 1];
den2=conv(temp,den);
[mag,phase,w]=bode(num,den2);
f=w/2/pi;
[mag1]=freqresp([1/fc*imp imp],[1e-12 1],f);

if curves==1,
     figure
     loglog(f,mag)
     v=axis;
     axis([1 1e10 v(3) v(4) ])
     hold on
     loglog(f,mag1,'r')
     xlabel('Frequency, Hz')
     ylabel('Impedance, Ohms')
     hold off
end

loop4=length(f);
while mag(loop4) < mag1(loop4), loopbw=f(loop4);,
loop4=loop4-1; end

if mag(length(f)) > mag1(length(f)),
     loopbw=0;
end

cdesign(loop3,4)=loopbw;
cdesign(loop3,5)=cdesign(loop3,1)*ctype(loop3,1);

end
end   %  end of imp. vs. freq. loop

%  Now necessary loop bandwidth for each capacitor
%  type has been found.  It is checked again switching
%  frequency to see if it is reasonable.

swf=input([blanks(5) 'Enter the regulator switching
frequency (Hz): -->  ']);
for loop1=1:1:length(cdesign(:,1)),
```

```
     if cdesign(loop1,4)>swf,
          cdesign(loop1,3)=-1;
     end
end

if max(cdesign(:,3))<1,
     disp(' ')
     disp([blanks(5) 'The power distribution network chosen
     requires too high of a loop'])
     disp([blanks(5) 'bandwidth given the switching frequency
     and filter capacitors.'])
     disp([blanks(5) 'Please increase the maximum output impedance
     or change the power'])
     disp([blanks(5) 'distribution network.  Decreasing maximum
     ESR may also help.'])
break
end

%finding gain due to PWM and oscillator.
vpp=input([blanks(5) 'Enter Vpp of oscillator sawtooth
waveform: -->  ']);
supply=input([blanks(5) 'Enter the input supply
voltage: -->  ']);
a=supply/vpp;

%finding ESR of inductor.
lesr=input([blanks(5) 'Enter ESR of inductor (ohms): -->  ']);

%finding Rdson of FETs.
rdson=input([blanks(5) 'Enter Rdson of FETs (ohms): -->  ']);

pm=input([blanks(5) 'Enter desired phase margin (positive
degrees): -->  ']);
minpm=30;
minpml=10;
if minpm>pm,
     minpm=pm-5;
end
if minpm<minpml,
     minpml=minpm;
end

disp(' ')
disp([blanks(5) 'Select Error Amplifier Type:'])
disp([blanks(5) '  1:  Op Amp'])
```

```
disp([blanks(5) '  2:  OTA'])
disp('')
eat=input([blanks(5) 'Enter your choice: -->  ']);
disp(' ')
disp([blanks(5) 'Select Amplifier Compensation Type:'])
disp([blanks(5) '  1:  Type 2'])
disp([blanks(5) '  2:  Type 3 (for op amp only)'])
disp([blanks(5) '  3:  Series RC'])
disp([blanks(5) '  4:  Enter compensation values manually
(type 2).'])
disp('')
cv=input([blanks(5) 'Enter your choice: -->  ']);

if eat==1,
     % ask for op amp characteristics.
     av=input([blanks(5) 'Enter low frequency open loop
     gain: -->  ']);
     p1=input([blanks(5) 'Enter dominant pole location
     (Hz): -->  ']);
     p1=p1*2*pi;
     p2=input([blanks(5) 'Enter high frequency pole location
     (Hz): -->  ']);
     p2=p2*2*pi;
end

if eat==2,
     % ask for ota characteristics.
     disp('')
     gm=input([blanks(5) 'Enter low frequency value of gm
     (mmhos): -->  ']);
     gm=gm*1e-3;
     p1=input([blanks(5) 'Enter OTA dominant pole location
     (Hz): -->  ']);
     p1=p1*2*pi;
     rl=input([blanks(5) 'Enter OTA output impedance
     (ohms): -->  ']);
end

% given error amplifier characteristics, compensation type,
% inductor value find compensation values for each capacitor
% type that works.  after values are found, collect info on
% phase margin and loop bandwidth.

for loopc=1:1:length(cdesign(:,1)),
```

```
loopc

% cycle through each capacitor type.
% check to see if capacitor type is worth trying to use.

%make sure cresults is the right size.

if cdesign(loopc,3)~=1,
     cresults(loopc,1)=0;
     crobust(loopc,1)=0;
end

if cdesign(loopc,3)==1,

esr=ctype(loopc,2)/cdesign(loopc,1);
c=cdesign(loopc,5);
out1=[a*c*esr a];
out2=[1*c c*(esr+rdson+lesr) 1];

if eat==1,

% while loop to check robustness of compensation.
n2=0;
rbst=0;

while (n2<2 & rbst==0),
     n2=n2+1;

g1=[av];
g2=[1/p1 1];
h1=[1];
h2=[1/p2 1];
[opamp1,opamp2]=series(g1,g2,h1,h2);

% find op amp compensation here.
% decide which compensation scheme to use.

if cv==1,
     if cdesign(loopc,4)>swf/10 & cdesign(loopc,4)<swf/5
          fco=swf/5;
     end
     if cdesign(loopc,4)<swf/10,
          fco=swf/10;
     end
     if cdesign(loopc,4)>swf/5,
```

```
          fco=cdesign(loopc,4)*1.2
     end

if n2==2,
fco=1.2*cdesign(loopc,4);
end
     wc=fco*2*pi;
     p=angle(out1(1)*(wc*i)+out1(2))-angle(out2(1)*(wc*i)^2
     +out2(2)*(i*wc)+out2(3));
     p=p*180/pi;
     m=pm;

if (n2==2 & m<60),
     m=60;
end

     boost=m-p-90;
     if boost>90,
          k=10;
     end
     k=tan((boost/2+45)*pi/180);
     if k>10,
          k=10;
     end
     g=1/abs((out1(1)*(wc*i)+out1(2))/(out2(1)*(wc*i)^2
     +out2(2)*(i*wc)+out2(3)));
     r1=1e3;
     c1=1/(wc*g*k*r1);
     c2=c1*(k^2-1);
     r2=k/(wc*c2);
     comp1=[r1*r2*c1*c2 r1*(c1+c2) 0];
     comp2=[r2*c2*1e-9 r2*c2+1e-9 1];
     [eamp1,eamp2]=feedback(opamp1,opamp2,comp1,comp2);
     [tot1,tot2]=series(eamp1,eamp2,out1,out2);
     cresults(loopc,1)=r1;
     cresults(loopc,2)=r2;
     cresults(loopc,3)=c1;
     cresults(loopc,4)=c2;
     [temp1,cresults(loopc,5),temp2,cresults(loopc,6)]
     =margin(tot1,tot2);
     cresults(loopc,6)=cresults(loopc,6)/2/pi;
     figure
     margin(tot1,tot2);
end
```

```
if cv==2,

     if cdesign(loopc,4)>swf/10 & cdesign(loopc,4)<swf/5
          fco=swf/5;
     end
     if cdesign(loopc,4)<swf/10,
          fco=swf/10;
     end
     if cdesign(loopc,4)>swf/5,
          fco=cdesign(loopc,4)*1.2
     end

if n2==2,
     fco=1.2*cdesign(loopc,4);
end

     wc=fco*2*pi;
     p=angle(out1(1)*(wc*i)+out1(2))-angle(out2(1)*(wc*i)^2
     +out2(2)*(i*wc)+out2(3));
     p=p*180/pi;
     m=pm;

if (n2==2 & m<60),
     m=60;
end

     boost=m-p-90;
     if boost>180,
          k=100;
     end
     k=(tan((boost/4+45)*pi/180))^2;
     if k>100,
          k=100;
     end
     g=1/abs((out1(1)*(wc*i)+out1(2))/(out2(1)*(wc*i)^2
     +out2(2)*(i*wc)+out2(3)));
     r1=1e3;
     c1=1/(wc*g*r1);
     c2=c1*(k-1);
     r2=k^0.5/(wc*c2);
     r3=r1/(k-1);
     c3=1/(wc*k^0.5*r3);
     comp1=[r1*r2*c1*c2*r3*c3 r3*c3*r1*(c1+c2)+r1*r2*c1*c2
     r1*(c1+c2) 0];
     comp2=[r2*c2*(r1*c3+r3*c3)*1e-9 (r1*c3+r3*c3+c2*r2)*1e-9
```

```
      +(r2*c2)*(r1*c3+r3*c3) 1e-9+r1*c3+r3*c3+c2*r2 1];
      [eamp1,eamp2]=feedback(opamp1,opamp2,comp1,comp2);
      [tot1,tot2]=series(eamp1,eamp2,out1,out2);
      cresults(loopc,1)=r1;
      cresults(loopc,2)=r2;
      cresults(loopc,3)=c1;
      cresults(loopc,4)=c2;
      [temp1,cresults(loopc,5),temp2,cresults(loopc,6)]
      =margin(tot1,tot2);
      cresults(loopc,6)=cresults(loopc,6)/2/pi;
      cresults(loopc,7)=r3;
      cresults(loopc,8)=c3;
      figure
      margin(tot1,tot2);
end

if cv==3,
       if cdesign(loopc,4)>swf/10 & cdesign(loopc,4)<swf/5
            fco=swf/5;
       end
       if cdesign(loopc,4)<swf/10,
            fco=swf/10;
       end
       if cdesign(loopc,4)>swf/5,
            fco=cdesign(loopc,4)*1.2
       end

if n2==2,
       fco=1.2*cdesign(loopc,4);
end

       wc=fco*2*pi;
       p=angle(out1(1)*(wc*i)+out1(2))-angle(out2(1)*(wc*i)^2
       +out2(2)*(i*wc)+out2(3));
       p=p*180/pi;
       m=pm;

if (n2==2 & m<60),
       m=60;
end

       boost=m-p-90;
       if boost>90,
            boost=90;
       end
```

```
      g=1/abs((out1(1)*(wc*i)+out1(2))/(out2(1)*(wc*i)^2
      +out2(2)*(i*wc)+out2(3)));
      r1=1e3;
      x=-tan((-boost)*pi/180)/wc;
      r2=g*r1;
      c1=x/r2;
      comp1=[r1*c1 0];
      comp2=[r2*c1 1];
      [eamp1,eamp2]=feedback(opamp1,opamp2,comp1,comp2);
      [tot1,tot2]=series(eamp1,eamp2,out1,out2);
      cresults(loopc,1)=r1;
      cresults(loopc,2)=r2;
      cresults(loopc,3)=c1;
      [temp1,cresults(loopc,5),temp2,cresults(loopc,6)]
      =margin(tot1,tot2);
      cresults(loopc,6)=cresults(loopc,6)/2/pi;
      figure
      margin(tot1,tot2);
end

ntimes=ntimes+1;

if cv==4,

if ntimes==1,

      r1=input([blanks(5) 'Enter R1 value (ohms): -->  ']);
      r2=input([blanks(5) 'Enter R2 value (ohms): -->  ']);
      c1=input([blanks(5) 'Enter C1 value (F): -->  ']);
      c2=input([blanks(5) 'Enter C2 value (F): -->  ']);
      comp1=[r1*r2*c1*c2 r1*(c1+c2) 0];
      comp2=[r2*c2*1e-9 r2*c2+1e-9 1];
      [eamp1,eamp2]=feedback(opamp1,opamp2,comp1,comp2);
end

      [tot1,tot2]=series(eamp1,eamp2,out1,out2);
      cresults(loopc,1)=r1;
      cresults(loopc,2)=r2;
      cresults(loopc,3)=c1;
      cresults(loopc,4)=c2;
      [temp1,cresults(loopc,5),temp2,cresults(loopc,6)]
      =margin(tot1,tot2);
      cresults(loopc,6)=cresults(loopc,6)/2/pi;
      figure
      margin(tot1,tot2);
```

```
      rbst=1;
end

[j1,j2,j3,wca]=margin(tot1,tot2);
[mag2,phase2,w2]=bode(tot1,tot2);
w3=find(w2<wca);
phase2=phase2(1:length(w3));
crobust(loopc,1)=min(phase2);

if (crobust(loopc,1)+180)>minpml,
      rbst=1;
end

end  %end of robustness while loop.

end %end of if eat==1 loop.

      if eat==2,

% find ota compensation here.
% decide which compensation type to use.

% while loop to check robustness of compensation.
n2=0;
rbst=0;

while (n2<2 & rbst==0),
       n2=n2+1;

g1=[gm];
g2=[1/p1 1];
ota1=g1;
ota2=g2;

if cv==2,
     disp(' ')
     disp([blanks(5) 'This choice is only for op amps.
     Using type 2 instead.'])
     cv=1;
end

if cv==1,
     if cdesign(loopc,4)>swf/10 & cdesign(loopc,4)<swf/5
          fco=swf/5;
     end
```

```
     if cdesign(loopc,4)<swf/10,
          fco=swf/10;
     end
     if cdesign(loopc,4)>swf/5,
          fco=cdesign(loopc,4)*1.2
     end

wc=fco*2*pi;

p=angle(out1(1)*(wc*i)+out1(2))-angle(out2(1)*(wc*i)^2
+out2(2)*(i*wc)+out2(3));
g=1/abs((out1(1)*(wc*i)+out1(2))/(out2(1)*(wc*i)^2
+out2(2)*(i*wc)+out2(3)));

gnuf=0;
tries=0;

while gnuf==0,

r1=20e3;

if n2==2,
      r1=100e3;
end

c1=1/r1/(abs(min(real(roots(out2))))+tries*100);
c2=c1/100;

if n2==2,
      c2=c1/1000;
end

tries=tries+1;

comp1=[c1*r1*rl rl];
comp2=[c1*c2*r1*rl (rl*c1+rl*c2+c1*r1) 1];
[eamp1,eamp2]=series(ota1,ota2,comp1,comp2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);

cresults(loopc,1)=r1;
cresults(loopc,3)=c1;
cresults(loopc,4)=c2;

[temp1,cresults(loopc,5),temp2,cresults(loopc,6)]=margin(tot1,tot2);
cresults(loopc,6)=cresults(loopc,6)/2/pi;
```

```
     if (cresults(loopc,5)>30 | tries>100),
          gnuf=1;
     end
end   % end of while loop
figure
margin(tot1,tot2);
end

if cv==3,
     if cdesign(loopc,4)>swf/10 & cdesign(loopc,4)<swf/5
          fco=swf/5;
     end
     if cdesign(loopc,4)<swf/10,
          fco=swf/10;
     end
     if cdesign(loopc,4)>swf/5,
          fco=cdesign(loopc,4)*1.2
     end
wc=fco*2*pi;

p=angle(out1(1)*(wc*i)+out1(2))-angle(out2(1)*(wc*i)^2
+out2(2)*(i*wc)+out2(3));
g=1/abs((out1(1)*(wc*i)+out1(2))/(out2(1)*(wc*i)^2
+out2(2)*(i*wc)+out2(3)));

gnuf=0;
tries=0;

while gnuf==0,

r1=20e3;

if n2==2,
      r1=100e3;
end

c1=1/r1/(abs(min(real(roots(out2))))+tries*100);
tries=tries+1;

comp1=[c1*r1*rl rl];
comp2=[c1*(r1+rl)  1];
[eamp1,eamp2]=series(ota1,ota2,comp1,comp2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);
```

```
cresults(loopc,1)=r1;
cresults(loopc,3)=c1;
[temp1,cresults(loopc,5),temp2,cresults(loopc,6)]=margin(tot1,tot2);
cresults(loopc,6)=cresults(loopc,6)/2/pi;

if (cresults(loopc,5)>minpm | tries>100),
     gnuf=1;
end

end % end of while loop

figure
margin(tot1,tot2);
end

ntimes=ntimes+1;

if cv==4,

if ntimes==1,
     r1=input([blanks(5) 'Enter R1 value (ohms): -->  ']);
     c1=input([blanks(5) 'Enter C1 value (F): -->  ']);
     c2=input([blanks(5) 'Enter C2 value (F): -->  ']);
     comp1=[c1*r1*rl rl];
     comp2=[c1*c2*r1*rl (rl*c1+rl*c2+c1*r1) 1];
     [eamp1,eamp2]=series(ota1,ota2,comp1,comp2);
end

[tot1,tot2]=series(eamp1,eamp2,out1,out2);
cresults(loopc,1)=r1;
cresults(loopc,3)=c1;
cresults(loopc,4)=c2;
[temp1,cresults(loopc,5),temp2,cresults(loopc,6)]=margin(tot1,tot2);
cresults(loopc,6)=cresults(loopc,6)/2/pi;
figure
margin(tot1,tot2);
rbst=1;
end

[j1,j2,j3,wca]=margin(tot1,tot2);
[mag2,phase2,w2]=bode(tot1,tot2);
w3=find(w2<wca);
phase2=phase2(1:length(w3));
crobust(loopc,1)=min(phase2);
```

```
if (crobust(loopc,1)+180)>minpml,
     rbst=1;
end

end     % end of robustness while loop.

end     % end of eat==2 loop.

end     % end of if loop for each cap type that works.

end     % end of for loop for checking each type of cap.

disp(' ')
disp('    R1        R2        C1        C2        PM        Loop BW
    R3     C3')
cresults

%remember to create a crobust and minpml in compensation scheme.

%screen out cap types that won't work here.

for loop3=1:1:length(cdesign(:,1)),

%check to see if bandwidth obtained is high enough.
     if cresults(loop3,6)<cdesign(loop3,4),
          cdesign(loop3,3)=-1;
     end

%check to see if phase margin obtained is large enough.
     if cresults(loop3,5)<minpm,
          cdesign(loop3,3)=-1;
     end

% check to see if phase curve gets too close to -180 before crossover.
     if crobust(loop3,1)+180<minpml,
          cdesign(loop3,3)=-1;
     end
end

gcaps=0;
for loop2=1:1:length(ctype(:,1)),
     if cdesign(loop2,3)==1,
           gcaps=gcaps+1;
     end
end
```

```
if gcaps==0,
     disp([blanks(5) 'No capacitor types will work.
     Try changing the power distribution'])
     disp([blanks(5) 'network parameters, number of
     capacitors used, desired phase margin,'])
     disp([blanks(5) 'or error amplifier characteristics.'])
     break
end

%with caps that are acceptable, pick the best one.

for p=1:1:length(cdesign(:,1)),

if cdesign(p,3)==1,
% adjust alpha1 and alpha2 according to relative
% importance of size and cost.
     alpha1=1;
     alpha2=0.4;
     whichcap(p,1)=alpha1*cdesign(p,2)
     +alpha2*cdesign(p,1)*ctype(p,3);
end

if cdesign(p,3)~=1,
     whichcap(p,1)=1e3;
end

end

[minscore,ctc]=min(whichcap);

disp(' ')
disp([blanks(5) 'The following capacitor type is recommended.'])
ctc
disp(' ')
disp('    Value     ESR       Factor    Cost')
ctype(ctc,:)
disp(' ')
disp('    Number    Cost      Accept?   Req BW    Tot Cap.')
cdesign(ctc,:)
disp(' ')
disp('    R1        R2        C1        C2        PM        Loop BW
    R3     C3')
cresults(ctc,:)
disp(' ')
disp([blanks(5) '  1:  Use this capacitor type.'])
```

```
disp([blanks(5) '  2:  Use a different capacitor than the
one suggested.'])
disp('  ')
ch2=input([blanks(5) 'Enter your choice: -->  ']);

if ch2==2,

disp(' ')
disp('    Value     ESR       Factor    Cost')
ctype
disp(' ')
disp('    Number    Cost      Accept?   Req BW    Tot Cap.')
cdesign
disp(' ')
disp('    R1        R2        C1        C2        PM        Loop BW
    R3     C3')
cresults
disp(' ')
ctc=input([blanks(5) 'Enter the row number of your choice: -->  ']);

end

%only considering one type of cap from here on.

c=cdesign(ctc,5);
esr=ctype(ctc,2)/cdesign(ctc,1);
out1=[a*c*esr a];
out2=[1*c c*(esr+rdson+lesr) 1];

mini=input([blanks(5) 'Enter minimum load (in Amps): -->  ']);
maxi=input([blanks(5) 'Enter maximum load (in Amps): -->  ']);
minv=input([blanks(5) 'Enter minimum
programmable output voltage: -->  ']);
maxv=input([blanks(5) 'Enter maximum
programmable output voltage: -->  ']);
if mini==0,
mini=1e-6;
end
rloadmin=minv/maxi;
rloadmax=maxv/mini;
rload=rloadmin;

if eat==1,

if (cv==1 | cv==4),
```

```
     r1=cresults(ctc,1);
     r2=cresults(ctc,2);
     c1=cresults(ctc,3);
     c2=cresults(ctc,4);
     comp1=[0 r1*r2*c1*c2 r1*(c1+c2) 0];
     comp2=[0 r2*c2*1e-9 r2*c2+1e-9 1];
end

if cv==2,
     r1=cresults(ctc,1);
     r2=cresults(ctc,2);
     c1=cresults(ctc,3);
     c2=cresults(ctc,4);
     r3=cresults(ctc,7);
     c3=cresults(ctc,8);
     comp1=[r1*r2*c1*c2*r3*c3 r3*c3*r1*(c1+c2)+r1*r2*c1*c2
     r1*(c1+c2) 0];
     comp2=[r2*c2*(r1*c3+r3*c3)*1e-9 (r1*c3+r3*c3+c2*r2)*1e-9
     +(r2*c2)*(r1*c3+r3*c3) 1e-9+r1*c3+r3*c3+c2*r2 1];
end

if cv==3,
     r1=cresults(ctc,1);
     r2=cresults(ctc,2);
     c1=cresults(ctc,3);
     comp1=[0 0 r1*c1 0];
     comp2=[0 0 r2*c1 1];
end

%now find how much op amp parameters can vary.
typpm=cresults(ctc,5);
typbw=cresults(ctc,6);

brokeyet=0;
n1=1;
while (brokeyet==0 & n1<11),
n1=n1+1;

g1=[av*n1];
g2=[1/p1*n1 1];
h1=[1];
h2=[1/p2*n1 1];
[vampchar1,vampchar2]=series(g1,g2,h1,h2);
[eamp1,eamp2]=feedback(vampchar1,vampchar2,comp1,comp2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);
```

```
[j1,varypm,j2,varywc]=margin(tot1,tot2);
varywc=varywc/2/pi;

totvary=0;
if (varypm>minpm | varypm>typpm & varywc>typbw*0.5),
     totvary=totvary+1;
end

[j1,j2,j3,wca]=margin(tot1,tot2);
[mag2,phase2,w2]=bode(tot1,tot2);
w3=find(w2<wca);
phase2=phase2(1:length(w3));
rbstv=min(phase2);
if (rbstv+180)>minpml,
     totvary=totvary+1;
end

g1=[av*n1];
g2=[1/p1/n1 1];
h1=[1];
h2=[1/p2/n1 1];
[vampchar1,vampchar2]=series(g1,g2,h1,h2);
[eamp1,eamp2]=feedback(vampchar1,vampchar2,comp1,comp2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);
[j1,varypm,j2,varywc]=margin(tot1,tot2);
varywc=varywc/2/pi;

if (varypm>minpm | varypm>typpm & varywc>typbw*0.5),
     totvary=totvary+1;
end

[j1,j2,j3,wca]=margin(tot1,tot2);
[mag2,phase2,w2]=bode(tot1,tot2);
w3=find(w2<wca);
phase2=phase2(1:length(w3));
rbstv=min(phase2);
if (rbstv+180)>minpml,
     totvary=totvary+1;
end

g1=[av/n1];
g2=[1/p1*n1 1];
h1=[1];
h2=[1/p2*n1 1];
[vampchar1,vampchar2]=series(g1,g2,h1,h2);
```

```
[eamp1,eamp2]=feedback(vampchar1,vampchar2,comp1,comp2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);
[j1,varypm,j2,varywc]=margin(tot1,tot2);
varywc=varywc/2/pi;

if (varypm>minpm | varypm>typpm & varywc>typbw*0.5),
     totvary=totvary+1;
end

[j1,j2,j3,wca]=margin(tot1,tot2);
[mag2,phase2,w2]=bode(tot1,tot2);
w3=find(w2<wca);
phase2=phase2(1:length(w3));
rbstv=min(phase2);
if (rbstv+180)>minpml,
     totvary=totvary+1;
end

g1=[av/n1];
g2=[1/p1/n1 1];
h1=[1];
h2=[1/p2/n1 1];
[vampchar1,vampchar2]=series(g1,g2,h1,h2);
[eamp1,eamp2]=feedback(vampchar1,vampchar2,comp1,comp2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);
[j1,varypm,j2,varywc]=margin(tot1,tot2);
varywc=varywc/2/pi;

if (varypm>minpm | varypm>typpm & varywc>typbw*0.5),
     totvary=totvary+1;
end

[j1,j2,j3,wca]=margin(tot1,tot2);
[mag2,phase2,w2]=bode(tot1,tot2);
w3=find(w2<wca);
phase2=phase2(1:length(w3));
rbstv=min(phase2);
if (rbstv+180)>minpml,
     totvary=totvary+1;
end

if totvary<8,
     brokeyet=1;
end
```

```
end     %end of while loop

disp(' ')
disp([blanks(5) ' The open loop gain and pole location of
the op amp can vary by the'])
disp([blanks(5) ' following factor without becoming unstable.'])
opampvar=n1-1

%reset variable here and continue

[eamp1,eamp2]=feedback(opamp1,opamp2,comp1,comp2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);

end

if eat==2,

typpm=cresults(ctc,5);
typbw=cresults(ctc,6);

if (cv==1 | cv==4),
     r1=cresults(ctc,1);
     c1=cresults(ctc,3);
     c2=cresults(ctc,4);
     comp1=[0 c1*r1*rl rl];
     comp2=[c1*c2*r1*rl (rl*c1+rl*c2+c1*r1) 1];

%now find how much ota parameters can vary.

brokeyet=0;
n1=1;
while (brokeyet==0 & n1<2.1),
n1=n1+0.1;

rlv=rl*n1;
compv1=[0 c1*r1*rlv rlv];
compv2=[c1*c2*r1*rlv (rlv*c1+rlv*c2+c1*r1) 1];
g1=[gm*n1];
g2=[1/p1 1];
tampchar1=g1;
tampchar2=g2;
[eamp1,eamp2]=series(tampchar1,tampchar2,compv1,compv2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);
[j1,varypm,j2,varywc]=margin(tot1,tot2);
varywc=varywc/2/pi;
```

```
totvary=0;
if (varypm>minpm | varypm>typpm & varywc>typbw*0.5),
     totvary=totvary+1;
end
[j1,j2,j3,wca]=margin(tot1,tot2);
[mag2,phase2,w2]=bode(tot1,tot2);
w3=find(w2<wca);
phase2=phase2(1:length(w3));
rbstv=min(phase2);
if (rbstv+180)>minpml,
     totvary=totvary+1;
end

rlv=rl*n1;
compv1=[0 c1*r1*rlv rlv];
compv2=[c1*c2*r1*rlv (rlv*c1+rlv*c2+c1*r1) 1];
g1=[gm/n1];
g2=[1/p1 1];
tampchar1=g1;
tampchar2=g2;
[eamp1,eamp2]=series(tampchar1,tampchar2,compv1,compv2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);
[j1,varypm,j2,varywc]=margin(tot1,tot2);
varywc=varywc/2/pi;

if (varypm>minpm | varypm>typpm & varywc>typbw*0.5),
     totvary=totvary+1;
end
[j1,j2,j3,wca]=margin(tot1,tot2);
[mag2,phase2,w2]=bode(tot1,tot2);
w3=find(w2<wca);
phase2=phase2(1:length(w3));
rbstv=min(phase2);
if (rbstv+180)>minpml,
     totvary=totvary+1;
end

rlv=rl/n1;
compv1=[0 c1*r1*rlv rlv];
compv2=[c1*c2*r1*rlv (rlv*c1+rlv*c2+c1*r1) 1];
g1=[gm*n1];
g2=[1/p1 1];
tampchar1=g1;
tampchar2=g2;
```

```
[eamp1,eamp2]=series(tampchar1,tampchar2,compv1,compv2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);
[j1,varypm,j2,varywc]=margin(tot1,tot2);
varywc=varywc/2/pi;

if (varypm>minpm | varypm>typpm & varywc>typbw*0.5),
     totvary=totvary+1;
end
[j1,j2,j3,wca]=margin(tot1,tot2);
[mag2,phase2,w2]=bode(tot1,tot2);
w3=find(w2<wca);
phase2=phase2(1:length(w3));
rbstv=min(phase2);
if (rbstv+180)>minpml,
      totvary=totvary+1;
end

rlv=rl/n1;
compv1=[0 c1*r1*rlv rlv];
compv2=[c1*c2*r1*rlv (rlv*c1+rlv*c2+c1*r1) 1];
g1=[gm/n1];
g2=[1/p1 1];
tampchar1=g1;
tampchar2=g2;
[eamp1,eamp2]=series(tampchar1,tampchar2,compv1,compv2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);
[j1,varypm,j2,varywc]=margin(tot1,tot2);
varywc=varywc/2/pi;

if (varypm>minpm | varypm>typpm & varywc>typbw*0.5),
     totvary=totvary+1;
end
[j1,j2,j3,wca]=margin(tot1,tot2);
[mag2,phase2,w2]=bode(tot1,tot2);
w3=find(w2<wca);
phase2=phase2(1:length(w3));
rbstv=min(phase2);
if (rbstv+180)>minpml,
     totvary=totvary+1;
end

if totvary<8,
     brokeyet=1;
end
```

```
end     %end while loop

end     %end of cv==1 or 4 loop

if cv==3,
     r1=cresults(ctc,1);
     c1=cresults(ctc,3);
     comp1=[0 c1*r1*rl rl];
     comp2=[0 c1*(r1+rl)  1];

brokeyet=0;
n1=1;
while (brokeyet==0 & n1<2.1),
n1=n1+0.1;

rlv=rl*n1;
compv1=[0 c1*r1*rlv rlv];
compv2=[0 c1*(r1+rlv)  1];
g1=[gm*n1];
g2=[1/p1 1];
tampchar1=g1;
tampchar2=g2;
[eamp1,eamp2]=series(tampchar1,tampchar2,compv1,compv2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);
[j1,varypm,j2,varywc]=margin(tot1,tot2);
varywc=varywc/2/pi;

totvary=0;
if (varypm>minpm | varypm>typpm & varywc>typbw*0.5),
     totvary=totvary+1;
end
[j1,j2,j3,wca]=margin(tot1,tot2);
[mag2,phase2,w2]=bode(tot1,tot2);
w3=find(w2<wca);
phase2=phase2(1:length(w3));
rbstv=min(phase2);
if (rbstv+180)>minpml,
     totvary=totvary+1;
end

rlv=rl/n1;
compv1=[0 c1*r1*rlv rlv];
compv2=[0 c1*(r1+rlv)  1];
g1=[gm*n1];
g2=[1/p1 1];
```

```
tampchar1=g1;
tampchar2=g2;
[eamp1,eamp2]=series(tampchar1,tampchar2,compv1,compv2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);
[j1,varypm,j2,varywc]=margin(tot1,tot2);
varywc=varywc/2/pi;

if (varypm>minpm | varypm>typpm & varywc>typbw*0.5),
     totvary=totvary+1;
end
[j1,j2,j3,wca]=margin(tot1,tot2);
[mag2,phase2,w2]=bode(tot1,tot2);
w3=find(w2<wca);
phase2=phase2(1:length(w3));
rbstv=min(phase2);
if (rbstv+180)>minpml,
     totvary=totvary+1;
end

rlv=rl*n1;
compv1=[0 c1*r1*rlv rlv];
compv2=[0 c1*(r1+rlv)  1];
g1=[gm/n1];
g2=[1/p1 1];
tampchar1=g1;
tampchar2=g2;
[eamp1,eamp2]=series(tampchar1,tampchar2,compv1,compv2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);
[j1,varypm,j2,varywc]=margin(tot1,tot2);
varywc=varywc/2/pi;

if (varypm>minpm | varypm>typpm & varywc>typbw*0.5),
     totvary=totvary+1;
end
[j1,j2,j3,wca]=margin(tot1,tot2);
[mag2,phase2,w2]=bode(tot1,tot2);
w3=find(w2<wca);
phase2=phase2(1:length(w3));
rbstv=min(phase2);
if (rbstv+180)>minpml,
     totvary=totvary+1;
end

rlv=rl/n1;
compv1=[0 c1*r1*rlv rlv];
```

```
compv2=[0 c1*(r1+rlv)  1];
g1=[gm/n1];
g2=[1/p1 1];
tampchar1=g1;
tampchar2=g2;
[eamp1,eamp2]=series(tampchar1,tampchar2,compv1,compv2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);
[j1,varypm,j2,varywc]=margin(tot1,tot2);
varywc=varywc/2/pi;

if (varypm>minpm | varypm>typpm & varywc>typbw*0.5),
     totvary=totvary+1;
end
[j1,j2,j3,wca]=margin(tot1,tot2);
[mag2,phase2,w2]=bode(tot1,tot2);
w3=find(w2<wca);
phase2=phase2(1:length(w3));
rbstv=min(phase2);
if (rbstv+180)>minpml,
     totvary=totvary+1;
end

if totvary<8,
     brokeyet=1;
end

end     %end of while loop

end     %end of if cv3 loop.

disp(' ')
disp([blanks(5) ' The transconductance and output
impedance of the OTA can vary by the'])
disp([blanks(5) ' following factor without becoming unstable.'])
tampvar=n1-0.1

[eamp1,eamp2]=series(ota1,ota2,comp1,comp2);
[tot1,tot2]=series(eamp1,eamp2,out1,out2);

end

fdone=0;
while fdone==0,
disp(' ')
disp([blanks(5) '  1:  Use an input filter.'])
```

```
disp([blanks(5) '  2:  Do not use an input filter.'])
uf=input([blanks(5) 'Enter your choice: -->  ']);
if uf==2,
     fdone=1;
end

if uf==1,

nci=input([blanks(5) 'Enter the number of capacitors used
in the filter: -->  ']);
ci=input([blanks(5) 'Enter the capacitance of each capacitor
(F): -->  ']);
ci=ci*nci;
rci=input([blanks(5) 'Enter the ESR of each capacitor
(ohms): -->  ']);
rci=rci/nci;
li=input([blanks(5) 'Enter the inductor value used in the
filter (H): -->  ']);
rli=input([blanks(5) 'Enter the ESR of the inductor plus the
source resistance (ohms): -->  ']);
d2=(maxv/supply)^2;

zei1=[l*c*(esr+rload) l+rdson*c*(esr+rload)+esr*rload*c rdson+rload];
zei2=[0 (esr+rload)*c 1];
[t1,t2]=series(out1,out2,eamp1,eamp2);
[z2,z1]=series(t1,t2*rload,zei2,zei1);
zi1=conv((t1+t2),z1);
zi2=conv(t2*d2,z2);
f=logspace(-2,8);
[zi]=freqresp(zi1,zi2,f);
figure
loglog(f,zi)
zs1=[li*ci*rci (rci*rli+li) rli];
zs2=[li*ci (rci+rli) 1];
[zs]=freqresp(zs1,zs2,f);
hold on
loglog(f,zs,'r')
%red line should be lower than yellow line for
%stability.
axis([1e-2 1e8 0.1*min([min(zs) min(zi)]) 10*max([max(zs) max(zi)])])

filtergood=1;
n=1;
for n=1:1:length(f);
```

```
if zs(n)>zi(n),
     filtergood=filtergood-1;
end

n=n+1;
end
if filtergood<1,
     filtergood=0;
end
wi=1/sqrt(l*c);
ws=1/sqrt(li*ci);
filtergood2=1;
if ws>wi,
     filtergood2=0;
end
hold off
disp(' ')

if (filtergood==1 & filtergood2==1),
     disp([blanks(5) '  This input filter will work.'])
     fdone=1;
end

if (filtergood==1 & filtergood2==0),
     disp([blanks(5) '  This input filter will probably work
     but the input filter'])
     disp([blanks(5) '  capacitor or inductor value should
     be increased.'])
     fdone=1;
end

if filtergood==0,
     disp([blanks(5) '  This input filter will not work because
     the output impedance is too high.'])
end

end %end of uf=1 loop

end %end of while loop

co=c;
rco=esr;
rload=rloadmin;
C=[(lb+lc) rs];
A=[1 rdson+lesr];
```

```
B2=[co 0];
B1=[lco*co rco*co 1];
D2=[cb*lcb (rcb*cb+rload*cb) 1];
D1=[lcb*cb*rload rcb*cb*rload rload];
s1=[lcb*cb rcb*cb 1];
s2=[(lb+lc)*cb+lcb*cb (rs*cb+rcb*cb) 1];
[num1,den1]=parallel(C,D1,[1],D2);
out3=conv(B1,num1);
temp1=out3;
[num2,den2]=parallel(B2,B1,D2,num1);
temp2=conv(num2,A);
temp3=[0 out3];
out4=temp2+temp3;
[out5,out6]=series(out3,out4,s1,s2);
out5=out5*a;

rload=rloadmax;
C=[(lb+lc) rs];
A=[l rdson+lesr];
B2=[co 0];
B1=[lco*co rco*co 1];
D2=[cb*lcb (rcb*cb+rload*cb) 1];
D1=[lcb*cb*rload rcb*cb*rload rload];
s1=[lcb*cb rcb*cb 1];
s2=[(lb+lc)*cb+lcb*cb (rs*cb+rcb*cb) 1];
[num1,den1]=parallel(C,D1,[1],D2);
out3=conv(B1,num1);
temp1=out3;
[num2,den2]=parallel(B2,B1,D2,num1);
temp2=conv(num2,A);
temp3=[0 out3];
out4=temp2+temp3;
[out7,out8]=series(out3,out4,s1,s2);
out7=a*out7;

rload=(maxv+minv)/(maxi+mini);

out3=[a*rload*esr*c a*rload];
out4=[l*c*(esr+rload) l+esr*rload*c+(rdson+lesr)*(esr+rload)*c
rload+(rdson+lesr)];

[smin1,smin2]=series(eamp1,eamp2,out5,out6);
[smax1,smax2]=series(eamp1,eamp2,out7,out8);
[j1,pm1,j2,wc1]=margin(smin1,smin2);
[j1,pm2,j2,wc2]=margin(smax1,smax2);
```

```
dpm=abs(pm1-pm2);
dwc=abs(wc1-wc2)/2/pi;

disp(' ')
disp([blanks(5) '  The change in phase margin and cross over
frequency'])
disp([blanks(5) '  with changing load current is shown below.
(degrees & hz)'])
dpm
dwc

disp([blanks(5) '  1:  Simulate regulator using these values
with comparators.'])
disp([blanks(5) '  2:  Simulate regulator using these values
without comparators.'])
disp([blanks(5) '  3:  Quit.'])
ch6=input([blanks(5) 'Enter your choice: -->  ']);
if eat==1,
     if ch6==1,
          vamp2
     end
     if ch6==2,
          vamp1
     end
        end
end

if eat==2,
     if ch6==1,
          tamp2
     end
     if ch6==2,
          tamp1
     end
     end
end
```

Appendix B

Motorola MMSF5NO3HD Power FET Model

```
***** MCSPICE SIMULATORS ****************************************
*************************  INSTANTIATION   **********************
.subckt PowerFET 10 20 30
*
* 10 = Drain 20 = Gate 30 = Source
*
*****************************************************************
*
*------------------------ EXTERNAL PARASITICS --------------------
* PACKAGE INDUCTANCE
*
LDRAIN  10  11  7.5e-09
LGATE   20  21  4.5e-09
LSOURCE 30  31  4.5e-09
*
* RESISTANCES
*
RDRAIN1    4  11 0.02556 TC1=0.01064 TC2=-6.14682e-06
RDRAIN2    4   5 0.001 TC1=0.01064 TC2=-6.14682e-06
RSOURCE   31   6 0.01518  TC1=-0.009967 TC2=2.36438e-05
RDBODY     8  30 0.03772 TC1=0.001953 TC2=-6.62384e-06
*
RGATE      21   2 5
*
*---------------------------------------------------------------
```

```
*
*--------------- CAPACITANCES AND BODY DIODE --------------------
*
DBODY       8  11 DBODY
DGD         3  11 DGD
CGDMAX      2   3 2.3e-09
RGDMAX      2   3 1e+08
CGS         2   6 1.182e-09
*
*----------------------------------------------------------------
*
*----------------------- CORE MOSFET ----------------------------
*
M1       5  2  6  6  MAIN
*
*----------------------------------------------------------------
*
*.MODEL RDRAIN R  &
*TC1=0.01064 &
*TC2=-6.14682e-06
*
*.MODEL RSOURCE R  &
*TC1=-0.009967 &
*TC2=2.36438e-05
*
*.MODEL RDBODY R  &
*TC1=0.001953 &
*TC2=-6.62384e-06
*
*
.MODEL MAIN NMOS  &
LEVEL=3 &
VTO=2.359 &
KP=22.07 &
GAMMA=1.5 &
PHI=0.6 &
LAMBDA=0.001 &
RD=0 &
RS=0 &
CBD=0 &
CBS=0 &
IS=1e-14 &
PB=0.8 &
CGSO=0 &
CGDO=0 &
```

```
CGBO=0 &
RSH=0 &
CJ=0 &
MJ=0.5 &
CJSW=0 &
MJSW=0.33 &
JS=1e-14 &
TOX=1e-07 &
NSUB=1e+15 &
NSS=0 &
NFS=2e+11 &
TPG=1 &
XJ=0 &
LD=0 &
UO=600 &
UCRIT=0 &
UEXP=0 &
UTRA=0 &
VMAX=0 &
NEFF=1 &
KF=0 &
AF=1 &
FC=0.5 &
DELTA=0 &
THETA=0 &
ETA=0 &
KAPPA=0.2
*
*-----------------------------------------------------------------
*
.MODEL DGD D  &
IS=1e-15 &
RS=0 &
N=1000 &
TT=0 &
CJO=8.633e-10 &
VJ=0.1 &
M=0.487 &
EG=1.11 &
XTI=3 &
KF=0 &
AF=1 &
FC=0.5 &
BV=10000 &
IBV=0.001
```

```
*
*---------------------------------------------------------------
*
.MODEL DBODY D  &
IS=1.668e-12 &
RS=0 &
N=1.018 &
TT=5e-09 &
CJO=1.2e-09 &
VJ=0.5302 &
M=0.3689 &
EG=1.11 &
XTI=4 &
KF=0 &
AF=1 &
FC=0.5 &
BV=45.91 &
IBV=0.00025
.ENDS
```

Bibliography

[1] Jade Alberkrack. Theory and applications of the MC34063 and μA78S40 switching regulator control circuits. Application Note AN920, Motorola Semiconductor Product Sector, Tempe, AZ, 1989.

[2] Vrej Barkhordarian. Power MOSFET basics. *Power Conversion and Intelligent Motion*, pages 28–39, June 1996.

[3] Richard M. Bass, Bonnie S. Heck, and Raheel A. Khan. Average modelling of current-mode controlled converters: Instability prediction. *International Journal of Electronics*, 77(5):613–628, 1994.

[4] Ashok Bindra. Pentium power ICs roll. *Electronic Engineering Times*, page 82, April 8, 1996.

[5] Charles-Edouard Cordonnier. Spice model for TMOS power MOSFETs. Application Note AN1043, Motorola Semiconductor Product Sector, 1989.

[6] Lloyd H. Dixon. Average current mode control of switching power supplies. Application Note U-140, Unitrode Integrated Circuits, Merrimack, NH, 1995-1996.

[7] Lloyd H. Dixon. Control loop cookbook. *Unitrode Switching Power Supply Design Seminar*, pages 5.1–5.26, June 1996.

[8] Lloyd H. Dixon. Switching power supply topology review. *Unitrode Switching Power Supply Design Seminar*, June 1996.

[9] Frank Goodenough. Power-supply rails plummet and proliferate. *Electronic Design*, pages 51–55, July 24, 1995.

[10] Frank Goodenough. High-output switcher IC converts 5 V to less than 3 V. *Electronic Design*, pages 55–60, June 24, 1996.

[11] Frank Goodenough. Advanced microprocessors demand amperes of current at < 2 V. *Electronic Design*, 45(2):31–42, January 1997.

[12] Frank Goodenough. VRMs: Technological first aid, but for how long? *Electronic Design*, 45(2):44–50, January 1997.

[13] Steve Goodfellow and Don Weiss. Designing power systems around processor specifications. *Electronic Design*, 45(2):53–60, January 1997.

[14] S. Hsu, A. Brown, L. Rensink, and R. D. Middlebrook. Modelling and analysis of switching DC-to-DC converters in constant-frequency current-programmed mode. In *IEEE Power Electronics Specialists Conference Proceedings*, pages 284–301, June 1979.

[15] Pentium® Pro processor power distribution guidelines. Application Note AP-523, Intel Corporation, November 1995.

[16] 5-bit programmable synchronous switching regulator controller for Pentium® Pro processor. Data sheet, Linear Technology Corporation, February 1997.

[17] Robert Mammano. Switching power supply topology - voltage mode vs. current mode. *Electronic Design*, June 27, 1994.

[18] Robert Mammano. Fueling the megaprocessors - empowering dynamic energy management. *Unitrode Switching Power Supply Design Seminar*, pages 1.11 – 1.22, June 1996.

[19] Craig Matsumoto. Linfinity leans toward systems. *Electronic Engineering Times*, pages 27–28, April 1996.

[20] Step-down controllers with synchronous rectifier for CPU power. Data Sheet MAX796/MAX797/MAX799, Maxim Integrated Products, Sunnyvale, CA, November 1994.

[21] John Maxwell. Capacitor applications in switching power supplies. In *15th Capacitor and Resistor Technology Symposium*, 1995.

[22] R. D. Middlebrook. Input filter considerations in design and application of switching regulators. In *IEEE Industry Applications Society Annual Meeting*, pages 366–382, 1976.

[23] R. D. Middlebrook. Design techniques for preventing input filter oscillations in switched-mode converters. In *Proceedings of the Powercon 5 Conference*, pages A3.1–A3.16, 1978.

[24] R. D. Middlebrook. Topics in multiple-loop regulators and current-mode programming. In *IEEE Power Electronics Specialists Conference Proceedings*, pages 716–732, Toulouse, France, June 1985.

[25] R. D. Middlebrook. Small-signal modeling of pulse-width modulated switched-mode power converters. *Proceedings of the IEEE*, 76(4):343–353, April 1988.

[26] Dennis O'Neill. Powering 3.3 V digital systems. Design Note 33, Linear Technology Corporation, April 1990.

[27] W. M. Polivka, P. R. K. Chetty, and R. D. Middlebrook. State-space average modelling of converters with parasitics and storage-time modulation. In *IEEE Power Electronics Specialists Conference*, pages 119–143, 1980.

[28] Construction and characteristics of oscon. Application Note 6, Sanyo Corporation, 1996.

[29] Bahram Shahian and Michael Hassul. *Control System Design Using Matlab.* Prentice Hall, Englewood Cliffs, NJ, 1993.

[30] Thomas A. Somerville and Robert L. Vyne. Improved curvature-corrected bandgap current and voltage reference. Patent Disclosure SC10110C, Motorola Semiconductor Products Sector, Tempe, AZ, 1996.

[31] F. D. Tan and R. D. Middlebrook. A unified model for current-programmed converters. *IEEE Transactions on Power Electronics*, 10(4):397–407, July 1995.

[32] Yannis Tsividis and Guido Masetti. Problems in precision modeling of the MOS transistor for analog applications. *IEEE Transactions on Computer-Aided Design*, CAD-3(1):72–79, January 1984.

[33] Yannis Tsividis and Ken Suyama. MOSFET modeling for analog circuit CAD: Problems and prospects. *IEEE Journal of Solid-State Circuits*, 29(3):210–216, March 1994.

[34] C. Varga. Power supply trends: How low can we go? *Power Conversion and Intelligent Motion*, pages 28–37, May 1996.

[35] H. D. Venable. The K factor: A new mathematical tool for stablitiy analysis and synthesis. In *Proceedings of Powercon 10*, 1993.

[36] G. W. Wester and R. D. Middlebrook. Low frequency characterization of switched DC-DC converters. *Third IEEE Power Processing and Electronics Specialists Conference*, AES-9(3):376–385, May 1973.