Distortion Analysis of Weakly Nonlinear Filters Using Volterra Series

by

James A. Cherry

A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements
for the degree of
Master of Engineering

Ottawa-Carleton Institute for Electrical Engineering,
Department of Electronics,
Carleton University,
Ottawa, Ontario, Canada

December 6, 1994

The undersigned recommend to the Faculty of Graduate
Studies and Research acceptance of the thesis

Distortion Analysis of Weakly Nonlinear Filters Using Volterra Series

submitted by James A. Cherry
in partial fulfilment of the requirements for
the degree of Master of Engineering.

Thesis Supervisor

Chairman,
Department of Electronics

Ottawa-Carleton Institute for Electrical Engineering,
Department of Electronics
Faculty of Engineering
Carleton University
December 6, 1994

Acknowledgements

Thanks to Arthur Castonguay for his suggestions on how best to edit and typeset this thesis.

Thanks to Luc Lussier for laying out test boards for me on short notice, and to Norm Filiol and John Long for discussions on how to use a network analyzer.

Thanks to the Natural Sciences and Engineering Research Council and to the Department of Electronics at Carleton for their financial assistance. They made the completion of this work possible.

Most of all, thanks to my supervisor, Martin Snelgrove, for motivation when I needed it most, for talking to me no matter how little time he had, and for all his thoughtful comments and advice.

Abstract

An analysis of weakly nonlinear band pass filters using Volterra series is presented. The Volterra transfer functions for a typical G_m-C biquad are derived analytically and used to quantify and unify the distortion that arises with multiple input sinusoids, specifically gain compression, desensitization, and intermodulation distortion. A feedback structure that can reduce distortion is analyzed algebraically, and practical examples of the structure are simulated and their distortion terms extracted using a novel technique. The latter is applied to an actual G_m-C biquad in the laboratory and measured performance agrees with that predicted by the Volterra analysis.

Contents

List of Figures

List of Tables

List of Symbols

dB in this thesis: Unless otherwise stated, a quantity x in dB is defined as $20\log_{10} x$.

Here is an explanation of some of the notation used in this thesis.

$\sum\limits_{(v;l,n)}$	Sum over all partitions v of n into l parts
$\sum_N'$	Sum of non-identical products of a partition
$\sum\limits_{n!}(x_{(1)},\cdots,x_{(n)})$	Sum over all permutations of subscripts i of x_i
$\dot{x}$	First time derivative of x
$\ddot{x}$	Second time derivative of x
$\|[x]\|$	$\exp(jxt)$

Here is a list of symbols and their definitions.

α	Duffing equation nonlinearity parameter
a	Duffing equation output magnitude
a_1, a_2, a_3	Coefficients of example nonlinear system
$a + jb$	$M_1(f_a)$, Volterra linear transfer function at f_a
A_0	Gain at filter center frequency f_0

A_{0A}	Gain at center frequency of desired tone filter in 3filt
A_{0B}, A_{0C}	Gain at center frequency of interferer filters in 3filt
A_1	Magnitude of measured Volterra M_1 term
A_3	Magnitude of measured Volterra M_3 term
$A(s)$	Linear transfer function of desired tone filter in 3filt
B_1	Magnitude in dB of measured Volterra M_1 term
B_3	Magnitude in dB of measured Volterra M_3 term
$B(s)$	Linear transfer function of one intererer filter in 3filt
$c + jd$	$M_3(f_a, f_a, -f_a)$, Volterra third-order transfer function for compression
C_1	Capacitor at output of TA 1 in general biquad filter
C_2	Capacitor at output of TA 2 in general biquad filter
$C(s)$	Linear transfer function of other interferer filter in 3filt
$CS\%$	Percentage channel separation, defined in (4.45)
ϵ	Duffing equation small parameter, §4.4.3
ϵ	Filter nonlinearity coefficient, everywhere but §4.4.3
ϵ_i	Coefficient of third-order term of TA i in general biquad filter
ϵ_1	Coefficient of third-order term of TA 1 in general biquad filter
ϵ_2	Coefficient of third-order term of TA 2 in general biquad filter

$e + jf$	$M_3(f_a, f_b, -f_b)$, Volterra third-order transfer function for desensitization
$E(t)$	Duffing equation forcing term
$\phi(t)$	Impulse response of filter in feedback path of 3filt
$\Phi(f)$	Linear transfer function of filter in feedback path of 3filt
$\triangle f$	Channel separation in Hz
$\triangle f_{min}$	Minimum channel separation in Hz
f_0	Filter center frequency in Hz
f_{0A}	Center frequency in Hz of desired tone filter in 3filt
f_{0B}, f_{0C}	Center frequency in Hz of interferer filters in 3filt
f_1, f_2	Input signal frequencies, §2.1.2
$f_1(v)$	Current-voltage characteristic of TA 1 in general biquad filter
$f_2(v)$	Current-voltage characteristic of TA 2 in general biquad filter
f	Frequency in Hz
f_a	Desired tone frequency in Hz
f_b, f_c	Interferer frequencies in Hz
$f_i(v)$	Current-voltage characteristic of TA i in general biquad filter
γ	Duffing equation output phase
g_{mi}	Transconductance on TA i in general biquad filter
g_{m1}	Transconductance on TA 1 in general biquad filter

g_{m2}	Transconductance on TA 2 in general biquad filter
g_n	nth Volterra kernel of an entire system
$g + jh$	$M_3(f_b, f_b, -f_c)$, Volterra third-order transfer function for intermodulation
G_{m12}	Transconductance of one f_0-tuning TA in [Shov93] filter
G_{m21}	Transconductance of other f_0-tuning TA in [Shov93] filter
G_{m22}	Transconductance of Q-tuning TA in [Shov93] filter
G_{mi}	Transconductance of A_0-tuning TA in [Shov93] filter
G_n	nth Volterra transfer function of an entire system
G_n	nth Volterra transfer function of 3filt
$h(t)$	Impulse response of a system
H_{An}	nth Volterra transfer function from input to good guy filter output in 3filt
H_{Bn}, H_{Cn}	nth Volterra transfer function from input to interferer filter outputs in 3filt
i	Current
IP_3	Input third-order intercept point
j	$\sqrt{-1}$
k	Duffing equation input amplitude parameter, §4.4.3
k	Feedback coefficient in 3filt
K_n	nth Volterra transfer function from input to error signal in 3filt

μ	Duffing equation damping parameter
M_n	nth Volterra transfer function in general biquad filter
M_{An}	nth Volterra transfer function in desired tone filter in 3filt
M_{Bn}, M_{Cn}	nth Volterra transfer function in interferer filters in 3filt
N	Number of partitions of n into l parts
$NL\%$	Percentage nonlinearity, defined in (4.44)
Q	Filter quality factor
Q_n	Transistor in schematic
R_1	Damping resistor in general biquad filter
σ	Duffing equation detuning parameter
s	Complex frequency
t	Time
T_0	Constant time shift
T_c	Compression term $M_3(f_a, f_a, -f_a)$
T_d, T_{d1}, T_{d2}	Desensitization terms $M_3(f_a, f_b, -f_b)$ and $M_3(f_a, f_c, -f_c)$
T_i, T_{i1}, T_{i2}	Intermodulation terms $M_3(f_b, f_b, -f_c)$ and $M_3(-f_a, f_b, f_c)$
θ_1	Phase angle of measured Volterra M_1 term
θ_3	Phase angle of measured Volterra M_3 term
u	Duffing equation output

v	Voltage
v_i, v_1, v_2	Time-domain signals at nodes of general biquad filter
v_1, v_2, v_3	Partitions of n into l parts
V_i, V_1, V_2	Frequency-domain signals at nodes of general biquad filter
V_1, V_2	Input signal amplitudes, §2.1.2
V_1	Output amplitude of first harmonic, Chapter 6
V_3	Output amplitude of third harmonic, Chapter 6
V_5	Output amplitude of fifth harmonic, Chapter 6
V_a	Amplitude of desired tone signal
$V_{a1,2,3,4}$	Amplitudes of desired tone signal for Volterra coefficient extraction
V_b, V_c	Amplitude of interferer signals
$V_{b1,2,3}$	Amplitudes of interferer signal for Volterra coefficient extraction
V_{c1}	Input voltage which causes 1dB compression at output
V_{in}	Duffing equation input voltage
$V_{out1,2,3,4}$	Amplitudes of output signals for Volterra coefficient extraction
$V_{Rout,1,2}$	Real part of output signals for Volterra coefficient extraction
$V_{Iout,1,2}$	Imaginary part of output signals for Volterra coefficient extraction
V_x	Input signal amplitude, Chapter 6
ω	Frequency in $\frac{\text{rad}}{\text{s}}$

Ω	Duffing equation input frequency in $\frac{\text{rad}}{\text{s}}$
ω_0	Filter center frequency in $\frac{\text{rad}}{\text{s}}$
ω_{0A}	Center frequency in $\frac{\text{rad}}{\text{s}}$ of desired tone filter in 3filt
ω_{0B}, ω_{0C}	Center frequency in $\frac{\text{rad}}{\text{s}}$ of interferer filters in 3filt
ω_a	Desired tone frequency in $\frac{\text{rad}}{\text{s}}$
$w(t)$	Time-domain error signal in 3filt
$W(s)$	Linear frequency-domain error signal in 3filt
$x(t)$	Time-domain input to filter or system
$X(f)$	Linear frequency-domain input to filter or system
$y(t)$	Time-domain output of filter or system
$y_A(t)$	Time-domain output of desired tone filter in 3filt
$y_B(t), y_C(t)$	Time-domain output of interferer filters in 3filt
y_n	nth Volterra kernel of system output
$Y(f)$	Linear frequency-domain output of filter or system
$Y_A(s)$	Linear frequency-domain output of desired tone filter in 3filt
$Y_B(s), Y_C(s)$	Linear frequency-domain output of interferer filters in 3filt
$z(t)$	Time-domain fed-back signal in 3filt

Chapter 1

Introduction

1.1 Linear and Nonlinear Systems

The concept of linear and nonlinear systems will certainly be familiar to the engineer. However, most will be more familiar with linear systems because these systems are easier to understand. They are amenable to straightforward algebraic manipulation, and hence can be solved and grasped more easily using any number of well-tried and well-understood techniques, such as matrix algebra and Laplace transforms.

Unfortunately, or perhaps fortunately, nothing in nature is perfectly linear under all conditions. Rather than throwing in the towel the engineer will frequently disregard nonlinearity if it can be gotten away with and "linearize" a problem so that it may be more easily understood and solved. Often, this approach succeeds: small-signal circuit analysis is a linearization technique that has been used with great success for years.

Linearization, however, is not the panacea of analysis. Some phenomena arise precisely *because* of nonlinearity and linear analysis offers little or no insight in such cases. Two such phenomena are injection locking [Vdp34] and chaos [IEEE87], and while linearization belies their very existence, both are demonstrable beyond doubt in the laboratory.

In the world of radio receivers there exist many phenomena that arise from nonlinearity, some desired, some not. Among the undesired are gain compression and intermodulation. The former occurs as input signals become too large for an amplifier or filter, which results in a loss of gain; the latter occurs when multiple signals interact to produce intermodulation products. It is these effects that this thesis considers.

In radio receivers, compression is usually associated with amplifiers, although it occurs in filters too. And both amplifiers and filters with slight nonlinearities can cause intermodulation distortion. These two effects are usually treated as different things, but they arise from the same problem: the unintentional yet unavoidably-present nonlinearities in circuit components. It transpires that these unintentional nonlinearities are quite "weak" under many conditions.

A well-known tool for characterizing weakly-nonlinear systems is the so-called Volterra series approach. This thesis applies Volterra series to the analysis of weakly-nonlinear band pass filters in radio receivers. It shows that compression and intermodulation, which are both generally classifiable as "distortion" effects, can be treated with the same formulae. It also demonstrates that Volterra series are ideally suited to the analysis of distortion in radio filtering circuits.

1.2 Contributions

This thesis contributes the following to the study of weakly nonlinear filters:

1. It gives explicit equations for Volterra transfer functions involving products of integrals such as $[\int y]^2[\int y^3]$. To the author's knowledge, such expressions have not been presented explicitly in the literature before.

2. It derives an explicit formula for the Volterra transfer functions of cascaded nonlinear systems. Such a formula is implied in [Bed71], equation (56) and partly derived in the time domain in [Rugh81] chapter 1 equation (46), but the

author has not seen it stated as compactly or generally in the frequency domain as it is here.

3. It analyzes an architecture that can greatly reduce the adjacent-channel interference that arises from nonlinearity in filters. Such a structure has been proposed before but has not been analyzed in this manner.

4. It describes a new technique for extracting Volterra transfer function terms from numerical simulations. Past efforts in this area typically cannot deal with gain compression and desensitization terms since they are only concerned with finding the *magnitudes* of the higher-order transfer functions, and this has been overcome here.

5. It applies the extraction method to the output of a SPICE simulation of a real circuit. More importantly, it demonstrates and automates the extraction of distortion values on an actual circuit in the laboratory using a network analyzer. This is the first time the author has seen a method for measuring the Volterra kernels for overlapping output tones.

6. It formalizes gain compression and expansion for a filter and unifies them, and uses the results to predict the 1dB-compression point for a real circuit at various frequencies.

7. It demonstrates that Volterra series are well-suited for calculating distortion in RF filtering problems. Moreover, it gives good evidence that if a filter is a second-order biquadratic BPF, the distortion results are independent of the exact implementation of the filter. All that matters is the transfer function itself.

1.3 Organization

Chapter 2 contrasts linear and nonlinear systems, examines the output characteristics of both types of systems, illustrates the typical front-end of a radio receiver, and discusses nonlinearity in circuit components and the problems introduced by it, such as gain compression and intermodulation. It culminates by proving the infeasibility of a circuit from the literature for AMPS cellular phones.

Chapter 3 introduces nonlinear systems with memory and Volterra series, explains how to derive the Volterra series representation from system equations, lists the formulae for the spectrum of the output when the input is sinusoidal, and discusses why Volterra series are good and circumstances under which they are valid.

Chapter 4 analyzes a simple biquadratic band pass filter, deriving its Volterra transfer functions explicitly and comparing their predictions of system performance to those of SPICE and a Runge-Kutta numerical differential equation solver for one- and two-tone inputs and characterizing general trends for three-tone inputs. It also explores the effect of strong nonlinearity by investigating the forced Duffing equation.

Chapter 5 proposes a feedback structure that can reduce the distortion produced by a weakly nonlinear filter. The Volterra transfer functions for the new structure are determined algebraically, and a method for numerically extracting distortion components from simulations is proposed and demonstrated both with the Runge-Kutta program and with SPICE on a realistic filter. It concludes with a discussion of some general properties of the feedback structure.

Chapter 6 applies the numerical extraction technique to an integrated circuit in the laboratory. It uses a network analyzer, a signal source, and a BASIC program to automate the extraction of linear gain, compression, and desensitization values. Gain compression and expansion for a filter are formalized and measured, and the general trends mentioned in Chapter 4 are confirmed.

Chapter 7 draws conclusions about this work and makes recommendations for future work.

Chapter 2

Problem Background

2.1 System Classification

Following is an exceedingly brief discussion of linear and nonlinear systems. It is certainly not intended to be all-encompassing or rigorous; many, many books have been written on both subjects, [Lath74] on linear systems, [Chua69] on nonlinear systems, to name just two. The intention is to provide an overview the key concepts that will be important in this thesis.

2.1.1 Linear, Time-Invariant Versus Nonlinear

By definition, a system is *linear* if and only if the principle of superposition holds. Formally, suppose a system takes an input $x(t)$ and produces an output $y(t)$ via some function f,

$$y(t) = f[x(t)] \tag{2.1}$$

where t is an independent variable, usually time. Suppose further that two inputs $x_1(t)$ and $x_2(t)$ produce outputs $y_1(t)$ and $y_2(t)$, respectively:

$$\begin{aligned} y_1(t) &= f[x_1(t)] & (2.2) \\ y_2(t) &= f[x_2(t)] & (2.3) \end{aligned}$$

Then, if the system is linear, an input $ax_1(t)+bx_2(t)$ will produce an output $ay_1(t)+by_2(t)$, where a and b are scalars. That is,

$$f[ax_1(t)+bx_2(t)] = ay_1(t)+by_2(t) \tag{2.4}$$

for a linear system. A system is *nonlinear* if and only if it is not linear.

A system is *time-invariant* if a time shift in the input is reproduced at the output. Suppose

$$y(t) = f[x(t)] \tag{2.5}$$

Let T_0 be a constant. Then, for all values of T_0,

$$f[x(t-T_0)] = y(t-T_0) \tag{2.6}$$

in a time-invariant system. A system is *time-varying* if it is not time-invariant.

Many real systems are both linear and time-invariant, and they are often denoted "LTI systems" for short.

2.1.2 Consequences

An important feature of LTI systems is this: *the output spectrum can only have tones at the same frequencies as the input spectrum.* A formal proof of this assertion will not be given, but it is not difficult to see why it is true.

First, it can be shown [Pap80] that an LTI system satisfies the familiar convolution formula

$$y(t) = \int_0^t x(\tau)h(t-\tau)d\tau = \int_0^t x(t-\tau)h(\tau)d\tau \tag{2.7}$$

where $y(t)$ is the output, $x(t)$ the input, and $h(t)$ the impulse response. Next, if $x(t)$ is an exponential e^{at}, we can see in (2.7)

$$\begin{aligned} y(t) &= \int_0^t e^{a(t-\tau)}h(\tau)d\tau \\ &= e^{at}\int_0^t e^{-a\tau}h(\tau)d\tau \end{aligned}$$

That is, the exponent at is not changed by convolution because the integral evaluates to a constant. Lastly, any sinusoidal signal is a sum of exponentials e^{jwt} and convolution doesn't alter the exponents jwt, i.e., the frequency. We may conclude that *in an LTI system, no new tones can appear at the output.*

The same cannot be said for a nonlinear system. Consider a nonlinear (though time-invariant) system with a square and a cubic nonlinearity. Let its defining equation be

$$y(t) = a_1 x(t) + a_2[x(t)]^2 + a_3[x(t)]^3 \tag{2.8}$$

If the input is a sum of two sinusoids at frequencies f_1 and f_2,

$$x(t) = V_1 \sin(2\pi f_1 t) + V_2 \sin(2\pi f_2 t) \tag{2.9}$$

its spectrum will be as shown in Figure 2.1, top [Wein80]. The spectrum in the Figure is for positive frequencies only; the spectrum for negative frequencies is the mirror image of the one depicted. By substituting (2.9) in (2.8) and making use of trigonometric identities such as

$$\begin{aligned}
\sin^2 x &= \frac{1}{2} - \frac{1}{2}\cos 2x \\
\sin^3 x &= \frac{3}{4}\sin x - \frac{1}{4}\sin 3x \\
\sin x \sin y &= \frac{1}{2}\cos(x-y) - \frac{1}{2}\cos(x+y) \\
\sin x \cos y &= \frac{1}{2}\sin(x+y) + \frac{1}{2}\sin(x-y) \\
\cos x \cos y &= \frac{1}{2}\cos(x+y) + \frac{1}{2}\cos(x-y)
\end{aligned} \tag{2.10}$$

we will arrive at the output spectrum shown in Figure 2.1, bottom. The two input tones have become thirteen tones at the output. The numbers above each line give the order of the tone, or, the number of terms that must be multiplied to give a tone at that frequency. It can be seen that when there are m input tones $f_1, \ldots, f_m$, a nonlinearity of order n produces a tone at all possible sums of n of $+f_1, -f_1, \ldots, +f_m, -f_m$.

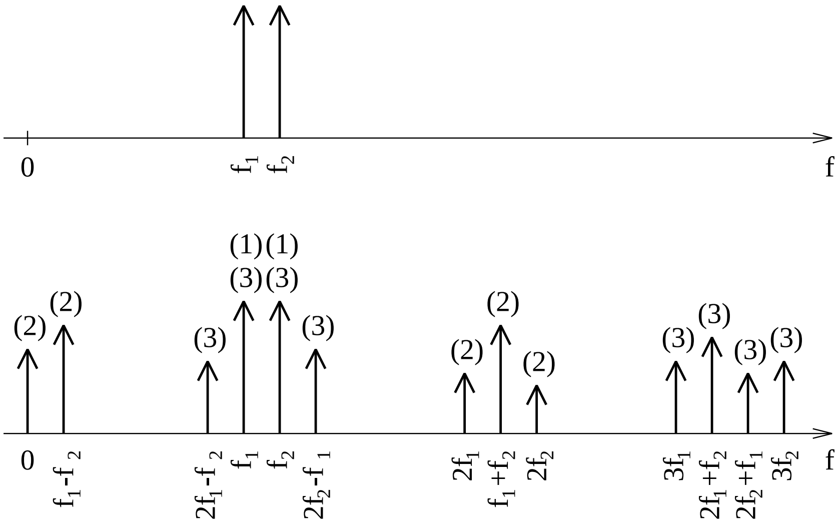


Figure 2.1: Input (top) and output (bottom) tones for a nonlinear system.

The reader will see that in the output spectrum, at the two input frequencies f_1 and f_2, there are contributors of order one *and* order three. We will be returning to this fact shortly.

2.2 Radio Receivers

2.2.1 Architecture and Functional Blocks

In 1918, E. H. Armstrong first perfected the superheterodyne radio receiver, and since then almost every radio receiver has been built the same way. A typical front-end for such a radio receiver is shown in Figure 2.2. The main functional blocks are as follows.

Band pass filter (BPF) This filter usually provides broad frequency selectivity. In many systems (such as those for receiving commercial FM radio broadcasts) it is tunable; its center frequency is set to the that of the desired radio-frequency (RF)

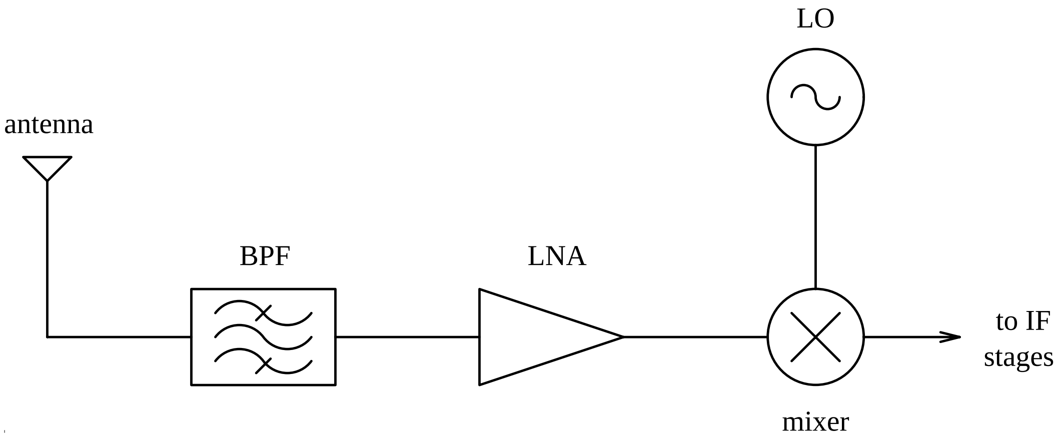


Figure 2.2: Typical front-end for superheterodyne radio receiver.

signal. Selection of the specific radio channel, which requires narrower filtering, is often difficult to accomplish at this stage because of the high frequency and the tunable nature of the filter, and it is usually done at the intermediate frequency (IF).

Low-noise amplifier (LNA) The weakest radio signals are not much stronger than thermal noise, and the LNA must both amplify weak signals *and* not degrade their signal-to-noise ratio (SNR) excessively. Excessive degradation leads to poor reception.

Local oscillator (LO) This tunable oscillator is set to the frequency given by the sum of the frequencies of the desired RF signal and the IF.

Mixer The mixer "heterodynes" (that is, multiplies) the LO signal and the desired RF signal, the effect of which is to make two copies of the signal: one at the IF and one at a much higher frequency. The mixer output is usually passed through a narrow BPF to attenuate adjacent channels and keep only the desired signal; the high-frequency copy is filtered out simultaneously.

2.2.2 Interference

The purpose of a radio receiver is to pick up a particular signal while ignoring all other signals. The radio spectrum is filled with signals that have frequencies as low as 9KHz for marine communications to frequencies as high as 300GHz for some satellites. A large amount of power also appears at 60Hz, the electrical power frequency in North America. The radio receiver must therefore select the desired signal out of a host of "interfering" signals.

It is possible that there will be large signals at a frequency close to the desired signal frequency. For example, commercial FM radio [Cook68] has channels allocated between 87.9MHz and 107.9MHz in 200kHz-steps; the desired signal could be at 105.9MHz while there might be a station only 200kHz away at 106.1MHz. Since the band pass filter is only broadly selective it will not usually be capable of attenuating such adjacent-channel interferers much.

To combat the problem of adjacent-channel interferers, frequency allocation is usually performed. The CRTC in Canada and the FCC in the United States assign and regulate frequencies throughout the spectrum. In FM radio, for example, transmitters within the same geographic region must be separated by at least 800kHz, which effectively prevents the most serious adjacent-channel interference that would arise were two stations to be adjacent in frequency, 200kHz apart, and in close physical proximity. Allocation is not completely effective because transmitters are not ideal; they transmit small signals at harmonics of the transmission frequency as well as broad-band noise.

Signals become less problematic as we move further away from the desired signal frequency because the band pass filter attenuation becomes stronger. Only large signals can cause significant interference. Signals near the mixer's so-called "image frequency" can be problematic.

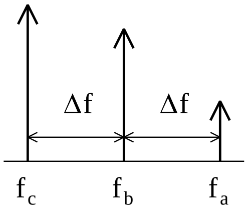


Figure 2.3: Example input spectrum.

2.2.3 Distortion

The discussion so far has been only concerned with linear response of the circuits used to build the receiver. When the circuits are nonlinear, interferers result in "distortion". We have already seen an example of distortion in Figure 2.1: at the input frequencies f_1 and f_2, the output spectrum has contributors of orders one *and* three. The order one terms are the desired *linear* output terms while the order three terms which coincide with the linear terms are distortion terms. These are invariably bad: let us first define the nomenclature connected with distortion and then demonstrate numerically why it is bad.

Suppose that a circuit in a radio receiver has an input-output relation like that in (2.8) with a square and cubic nonlinearity. Furthermore, suppose we are trying to receive a signal at a frequency f_a of amplitude V_a (the "desired signal") and that there are two interfering signals at $f_b = f_a - \triangle f$ and $f_c = f_a - 2\triangle f$ of amplitudes V_b and V_c (the "interferers"). The input spectrum will then be as shown in Figure 2.3.

The input equation will be

$$x(t) = V_a \sin(2\pi f_a t) + V_b \sin(2\pi f_b t) + V_c \sin(2\pi f_c t) \tag{2.11}$$

Substituting (2.11) in (2.8) and simplifying using (2.10) we obtain for the output

$$\begin{aligned}
y(t) &= a_1 V_a \sin(2\pi f_a t) && \text{order 1, due to } f_a \\
&+ \tfrac{3a_3 V_a^3}{4} \sin(2\pi f_a t) && \text{order 3, due to } f_a + f_a - f_a \\
&+ \tfrac{3a_3 V_a V_b^2}{2} \sin(2\pi f_a t) && \text{order 3, due to } f_a + f_b - f_b \\
&+ \tfrac{3a_3 V_a V_c^2}{2} \sin(2\pi f_a t) && \text{order 3, due to } f_a + f_c - f_c \\
&+ \tfrac{3a_3 V_b^2 V_c}{4} \sin(2\pi f_a t) && \text{order 3, due to } f_b + f_b - f_c \\
&+ \cdots && \text{terms at other frequencies}
\end{aligned} \tag{2.12}$$

The five terms in (2.12) comprise a complete list of the tones that appear at the desired signal frequency f_a. They can be classified as follows.

Linear gain The first term corresponds to the linear gain, a_1, that the tone at f_a experiences. This term is usually the one the circuit was designed to produce. The other four terms are all due to the undesired yet still present third-order nonlinearity.

Gain compression/Gain expansion The second term tells how the gain varies as a function of input amplitude. In a linear receiver, the gain remains constant for any input amplitude, but for a nonlinear receiver, the gain changes as the input becomes larger. The gain ultimately becomes smaller, and this is termed *gain compression*, but in some cases, the gain increases slightly first, and this is *gain expansion.* Gain compression can be observed even if f_a is the only tone in $x(t)$.

Desensitization The third and fourth terms result from the adjacent signals f_b and f_c interacting with the desired signal f_a. Consider an experiment where the amplitude V_a is held constant and the amplitude V_b is slowly increased. In a linear system, the gain at f_a will be unaffected by V_b, but in this nonlinear system, the gain at f_a will eventually change. It almost invariably decreases, and this lowering in gain is called *desensitization.*

Intermodulation In general, *intermodulation* refers to the interaction of tones through a multiplicative effect. In a broad sense, all the tones in Figure 2.1 that are not at f_1 and f_2 are a result of intermodulation and some might refer to all the terms of order three in (2.12) as intermodulation terms. In this thesis, the phrase "intermodulation term" will be applied to any term that is not a compression or a desensitization term. Thus, only the fifth term of (2.12) would be denoted an intermodulation term.

2.2.4 Quantification of Interference Effects

The severity of the distortion introduced by the third-order terms depends both on the strength of the third-order nonlinearity *and* the magnitudes of the adjacent signals. In the North American cellular phone ("AMPS") environment, adjacent channels are 30kHz apart, and a signal two channels away can be as much as 60dB or 80dB stronger than the desired signal [Fish79, Rap94]. Even tiny nonlinearities in the base station BPF or LNA can swamp the desired signal as we shall now demonstrate.

We will use a circuit from the literature, [Mey94]. The circuit is an LNA and a mixer integrated on silicon and designed for operation at around 900MHz. The LNA is designed to be linear and has a linear gain of about 16dB at 900MHz. Of course, nonlinearity is unavoidably present in the LNA and a quantification of it can be found from the quoted *input third-order intercept* value of $\mathrm{IP}_3 = -10\mathrm{dBm}$ into $R_s = 50\Omega$.

To measure IP_3, a graph such as the one in Figure 2.4 is drawn. A single tone of frequency f_a and amplitude V_a is applied to the input and the amplitude of the output tone is plotted as a function of V_a (the $\times$ on the graph). Next, two equal-amplitude tones are applied at f_a and $f_a + \epsilon$ and the amplitude of the output tone at $f_a + 2\epsilon$ is plotted (the $\circ$ on the graph). Here, $\epsilon \approx 0$ so that the measurement approximately characterizes the second term of (2.12), the compression term. As V_a increases, both lines deviate from linear so they are linearly extrapolated. The x-coordinate of their intersection point is the value of IP_3.

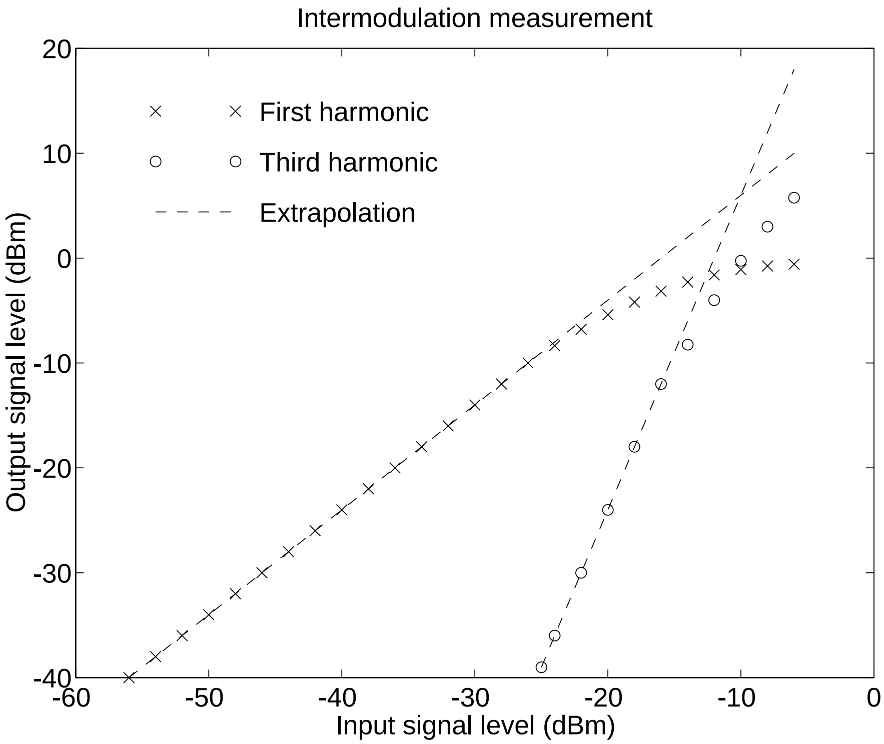


Figure 2.4: Example graph for IP$_3$ measurement.

Let us assume the LNA input-output relation is given by (2.8), repeated here for convenience:

$$y(t) = a_1 x(t) + a_2[x(t)]^2 + a_3[x(t)]^3 \tag{2.13}$$

In an actual circuit, the (a_1, a_2, a_3) values will depend on frequency but we will ignore that for the moment. We can calculate a_1 and a_3 as follows. First, for an input of amplitude V_a, the linear output amplitude is $a_1 V_a$ from (2.12). The linear gain is given as 16dB, so

$$\begin{aligned} \frac{\text{Output amplitude}}{\text{Input amplitude}} &= 16\text{dB} \\ \frac{a_1 V_a}{V_a} &= 10^{\frac{16}{20}} \\ a_1 &= 6.31 \end{aligned}$$

Second, at IP$_3$, i.e., at $V_a = -10\text{dBm} = 100\text{mV}$, the compression term and linear

Table 2.1: Distortion components in [Mey94] filter.

Component	$V_b = 1$mV $V_c = 1$mV,	$V_b = 1$mV $V_c = 10$mV,	$V_b = 10$mV $V_c = 10$mV,
Linear gain	63.1μV	63.1μV	63.1μV
Gain compression	0.63pV	0.63pV	0.63pV
Desensitization from f_b	12.6nV	12.6nV	1.26μV
Desensitization from f_c	12.6nV	12.6nV	1.26μV
Intermodulation	0.63μV	6.31μV	631μV

term have equal amplitudes. So, from (2.12),

$$\begin{aligned} a_1 V_a &= \frac{3a_3 V_a^3}{4} \text{ for } V_a = 100\text{mV} \\ a_3 &= \frac{4a_1}{3V_a^2} \\ &= \frac{4 \times 6.31}{3(0.1)^2} \\ &= 841 \end{aligned}$$

So, let us use these a_1 and a_3 in (2.12) to calculate the amplitudes of *all* distortion terms. Assuming a small amplitude of $V_a = 10\mu$V for the desired signal, Table 2.1 shows the output amplitudes for the five terms in (2.12) for various interferer amplitudes.

Even in the case where the adjacent signals are only 40dB and 60dB stronger than the desired signal, already the intermodulation term is 10% of the linear term, as can be seen in the second column of the Table. When both signals are 60dB stronger (the third column), the linear term is dominated by the intermodulation term. Under worst-case conditions, then, this circuit would be unsuitable for cellular telephone applications in North America. (The problem is the low IP$_3$ value of -10dBm; better circuits can improve this value to almost $+20$dBm, in which case a_3 and the intermodulation component both drop by three orders of magnitude and the worst-case performance becomes acceptable.)

We now see why distortion is bad: it can lead to the desired signal being swamped by the interferers and hence rendering it unrecoverable — completely defeating the purpose of a radio receiver. This example has been investigating nonlinearity in the LNA, but nonlinearity in the BPF can be just as detrimental.

2.2.5 Why Volterra Series?

The previous example may seem more complicated than necessary. After all, any radio engineer worth his or her salt knows how to read an IP_3 graph and can calculate directly what limits signals should not surpass without resorting to the analysis presented here.

It transpires that Volterra series are an attractive way of generalizing distortion calculations. After their introduction in the next chapter, a case is made for their use.

Chapter 3

Volterra Series

3.1 Introduction to Volterra Series

3.1.1 Background

The Spanish mathematician Vito Volterra first introduced the notion of what is now known as a Volterra series in his "Theory of Functionals" [Vol59]. The first major application of Volterra's work to nonlinear circuit analysis was done by the mathematician Norbert Wiener at M.I.T., who used them in a general way to analyze a number of problems including the spectrum of an FM system with a Gaussian noise input [Wien58]. Since then, Volterra series have found a great deal of use in calculating small, but nevertheless troublesome, distortion terms in transistor amplifiers and other systems [Nar70, Buss74].

In the next section, a brief introduction to the concept of Volterra series and the associated terminology is given. Discussion of some of the conditions under which Volterra series may be applied will be postponed until §3.4.

3.1.2 Volterra Series Representation

A linear, causal system with memory can be described by the convolution representation [Rugh81]

$$y(t) = \int_{-\infty}^{\infty} h(\sigma)x(t-\sigma)d\sigma \tag{3.1}$$

where $x(t)$ is the input, $y(t)$ the output, and $h(t)$ the impulse response of the system. A nonlinear system without memory can be described with a Taylor series

$$y(t) = \sum_{n=1}^{\infty} a_n[x(t)]^n \tag{3.2}$$

where, again, $x(t)$ is the input and $y(t)$ the output. The a_n are the Taylor series coefficients.

A Volterra series combines the above two representations to describe a nonlinear system with memory

$$\begin{aligned}
y(t) &= \sum_{n=1}^{\infty} \frac{1}{n!} \int_{-\infty}^{\infty} du_1 \cdots \int_{-\infty}^{\infty} du_n g_n(u_1, \ldots, u_n) \prod_{r=1}^{n} x(t-u_r) && (3.3)\\
&= \frac{1}{1!} \int_{-\infty}^{\infty} du_1 g_1(u_1)x(t-u_1) && (3.4)\\
&+ \frac{1}{2!} \int_{-\infty}^{\infty} du_1 \int_{-\infty}^{\infty} du_2 g_2(u_1,u_2)x(t-u_1)x(t-u_2) && (3.5)\\
&+ \frac{1}{3!} \int_{-\infty}^{\infty} du_1 \int_{-\infty}^{\infty} du_2 \int_{-\infty}^{\infty} du_3 g_3(u_1,u_2,u_3)x(t-u_1)x(t-u_2)x(t-u_3) && (3.6)\\
&+ \cdots
\end{aligned}$$

$x(t)$ is the input, $y(t)$ is the output, and the $g_n(u_1, \ldots, u_n)$ are called the *Volterra kernels* of the system or simply the *kernels* [Bed71]. The u_i are time variables and are labeled u_i instead of t_i to distinguish them better from t. For $n = 1$, $g_1(u_1)$ will be recognized as the familiar impulse response ($h(t)$ in equation (3.1)); thus, g_n for $n > 1$ are rather like "higher-order impulse responses." These serve to characterize the various orders of nonlinearity [Bed71]. The first few terms of (3.3) have been explicitly written out; (3.4) is the familiar convolution integral (3.1), and (3.5) and (3.6) may be thought of as two-fold and three-fold convolution. (3.3) is an infinite sum of n-fold convolution integrals.

The leading $\frac{1}{n!}$ is omitted by almost all authors except Bedrosian and Rice [Bed71] (see, for example, [Rugh81] chap. 1 eq. (36), [Buss74] eq. (2.3), [Chua82] eq. (1.1)). It is included in this thesis because it simplifies many calculations. The underlying assumption is that the kernels g_n are *symmetric*, which means that $g_n(u_1, \ldots, u_n)$ must have the same value regardless of the permutation of $u_1, \ldots, u_n$. If a system has an unsymmetric kernel γ_n, Wiener showed [Wien58] that it may be symmetrized by permuting the subscripts on the t_i in all possible ways and then taking g_n to be $\frac{1}{n!}$ times the sum of all such γ_n. The rest of this work assumes the kernels are symmetric. (In point of fact, if Chua had used symmetric kernels in [Chua82], his equations (2.16) and (4.12) would have become much shorter.)

Just as (3.3) is analogous to n-fold convolution, there exist n-fold analogies to Laplace and Fourier transforms. These are defined by

$$G_n(f_1, \ldots, f_n) = \int_{-\infty}^{\infty} du_1 \cdots \int_{-\infty}^{\infty} du_n g_n(u_1, \ldots, u_n) e^{-s_1 u_1} \cdots e^{-s_n u_n} \tag{3.7}$$

and

$$G_n(f_1, \ldots, f_n) = \int_{-\infty}^{\infty} du_1 \cdots \int_{-\infty}^{\infty} du_n g_n(u_1, \ldots, u_n) e^{-j\omega_1 u_1} \cdots e^{-j\omega_n u_n} \tag{3.8}$$

where $\omega_i = 2\pi f_i$, $s_i = j\omega_i$. It is clear that $G_1(s_1)$ is the familiar linear transfer function, and thus G_n in general is referred to as the *nth-order transfer function.* These frequency-domain representations G_n are useful because often they are much simpler to calculate than the sometimes prohibitively-complex time-domain representations g_n, and because radio problems are often approached in the frequency domain. From the definitions (3.7) and (3.8), it follows that if $g_n(u_1, \ldots, u_n)$ is a symmetric function of the u_i, then $G_n(f_1, \ldots, f_n)$ is a symmetric function of the f_i. The traditional frequency-domain input-output representation now becomes [Bed71]

$$\begin{aligned} Y(f) &= \frac{1}{1!} G_1(f) X(f) \\ &+ \frac{1}{2!} \int_{-\infty}^{\infty} df_1 G_2(f_1, f - f_1) X(f_1) X(f - f_1) \end{aligned}$$

$$\begin{aligned} &+ \frac{1}{3!}\int_{-\infty}^{\infty} df_1 \int_{-\infty}^{\infty} df_2 G_3(f_1, f_2, f - f_1 - f_2) X(f_1) X(f_2) X(f - f_1 - f_2) \\ &+ \cdots \end{aligned} \tag{3.9}$$

The term "kernel" will sometimes be used to describe either g_n or G_n. Although this is technically incorrect, the author feels it will be clear what is meant.

3.2 Determination of the Kernels

When an explicit equation relating the input $x(t)$ to the system output $y(t)$ is known, the techniques below may be used to determine the Volterra kernels G_n or g_n. Practical measurement techniques (i.e., those that can be used in the laboratory to characterize g_n for an actual circuit) are not considered here and will be discussed in Chapter 6.

3.2.1 The Harmonic Input Method

This method is for determining the kernels G_n in the frequency domain. When the input is

$$x(t) = \exp(j\omega_1 t) + \ldots + \exp(j\omega_n t) \tag{3.10}$$

where $\omega_i = 2\pi f_i, i = 1, \ldots, n$, and the ω_i are incommensurable, then

$$G_n(f_1, \ldots, f_n) = \{\text{coefficient of } \exp[j(\omega_1 + \ldots + \omega_n)t] \text{ in } (3.3)\} \tag{3.11}$$

A formal proof is provided in Appendix A.

Thus, if we assume

$$\begin{aligned} x(t) &= \exp(j\omega_1 t) \\ y(t) &= \sum_{k=1}^{\infty} c_k \exp(jk\omega_1 t) \end{aligned}$$

then $G_1(f_1)$ is equal to c_1. Similarly, if we assume

$$\begin{aligned} x(t) &= \exp(j\omega_1 t) + \exp(j\omega_2 t) \\ y(t) &= \sum_{k=0}^{\infty}\sum_{l=0}^{\infty} c_{kl}\exp(j(k\omega_1 + l\omega_2)t) \end{aligned}$$

then $G_2(f_1, f_2) = c_{11}$. Also, we have that $c_{00} = 0$, $c_{10} = G_1(f_1)$, and $c_{01} = G_1(f_2)$. Similar relations hold true when $x(t)$ is composed of more than two terms.

A nice example of the harmonic input method is to apply it to a system where

$$y(t) = x(t) + \epsilon[\dot{x}(t)]^2\ddot{x}(t) \tag{3.12}$$

This equation arises in some forms of the quasi-static approximation to filtered FM [Bed71]. Let us find the Volterra kernels G_n for this system. First, let $x(t) = \exp(j\omega_1 t)$ and substitute this into (3.12).

$$\begin{aligned} y(t) &= \exp(j\omega_1 t) + \epsilon\left[(j\omega_1)^2\exp(2j\omega_1 t)\right](j\omega_1)^2\exp(j\omega_1 t) \\ &= \exp(j\omega_1 t) + \epsilon\omega_1^4\exp(3j\omega_1 t) \qquad (3.13) \end{aligned}$$

Therefore from (3.11)

$$\begin{aligned} G_1(f_1) &= \{\text{coeff of } \exp(j\omega_1 t) \text{ in } (3.13)\} \\ &= 1. \end{aligned}$$

If $x(t) = \exp(j\omega_1 t) + \exp(j\omega_2 t)$ in (3.12) now,

$$\begin{aligned} y(t) &= \exp(j\omega_1 t) + \exp(j\omega_2 t) + \epsilon\left[j\omega_1\exp(j\omega_1 t) + j\omega_2\exp(j\omega_2 t)\right]^2 \\ &\quad \times\left[(j\omega_1)^2\exp(j\omega_1 t) + (j\omega_2)^2\exp(j\omega_2 t)\right] \\ &= |[\omega_1]| + |[\omega_2]| + \epsilon\left\{\omega_1^2|[2\omega_1]| + \omega_1\omega_2|[\omega_1+\omega_2]| + \omega_2^2|[2\omega_2]|\right\} \\ &\quad \times\left\{\omega_1^2|[\omega_1]| + \omega_2^2|[\omega_2]|\right\} \\ &= |[\omega_1]| + |[\omega_2]| + \epsilon\left\{\omega_1^4|[3\omega_1]| + \omega_1^3\omega_2|[2\omega_1+\omega_2]| + \omega_1^2\omega_2^2|[\omega_1+2\omega_2]|\right. \\ &\quad \left. + \omega_1^2\omega_2^2|[2\omega_1+\omega_2]| + \omega_1\omega_2^3|[\omega_1+2\omega_2]| + \omega_2^4|[3\omega_2]|\right\} \qquad (3.14) \end{aligned}$$

where $|[x]|$ is an abbreviation for $\exp(jxt)$. There are no $|[\omega_1+\omega_2]|$ terms in (3.14), which means

$$G_2(f_1, f_2) = 0.$$

Lastly, letting $x = |[\omega_1]| + |[\omega_2]| + |[\omega_3]|$ and substituting into (3.12) gives

$$\begin{aligned} y(t) &= |[\omega_1]| + |[\omega_2]| + |[\omega_3]| + \epsilon \left\{ j\omega_1|[\omega_1]| + j\omega_2|[\omega_2]| + j\omega_3|[\omega_3]| \right\}^2 \\ &\quad \times \left\{ -\omega_1^2|[\omega_1]| - \omega_2^2|[\omega_2]| - \omega_3^2|[\omega_3]| \right\} \qquad (3.15) \end{aligned}$$

We can see there will be three $|[\omega_1+\omega_2+\omega_3]|$ terms in (3.15), which makes

$$\begin{aligned} G_3(f_1, f_2, f_3) &= -2\epsilon\omega_1\omega_2(-\omega_3^2) - 2\epsilon\omega_1\omega_3(-\omega_2^2) - 2\epsilon\omega_2\omega_3(-\omega_1^2) \\ &= 2\epsilon\omega_1\omega_2\omega_3(\omega_1+\omega_2+\omega_3) \end{aligned}$$

For $n > 3$, $G_n(f_1, \ldots, f_n)$ will turn out to be zero.

It is clear that the complexity of the harmonic input method increases rapidly as n increases, but symbolic algebra programs such as Maple or Macsyma can assist greatly in such calculations [Chua82].

3.2.2 The Direct Expansion Method

This method is for determining the kernels g_n in the time domain. Here, the system equations are manipulated until they are brought into the form of a Volterra series (3.3), and the g_n are simply "read off" the representation. The circuit in Figure 3.1 [Rugh81], the multiplicative connection of three linear subsystems, is amenable to analysis using the direct expansion method.

For each subsystem, the defining equation can be written

$$y_i(t) = \int_{-\infty}^{\infty} du_i h_i(u_i) x(t-u_i), \ \ i = 1, 2, 3$$

and so the overall transfer function is

$$y(t) \ = \ y_1(t)y_2(t)y_3(t)$$

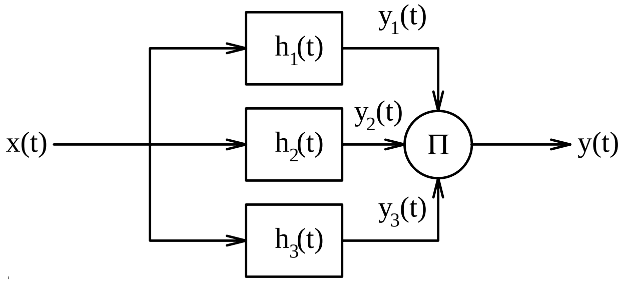


Figure 3.1: Example system for direct expansion method.

$$\begin{aligned} &= \int_{-\infty}^{\infty} du_1 h_1(u_1) x(t-u_1) \int_{-\infty}^{\infty} du_2 h_2(u_2) x(t-u_2) \int_{-\infty}^{\infty} du_3 h_3(u_3) x(t-u_3) \\ &= \int_{-\infty}^{\infty} du_1 \int_{-\infty}^{\infty} du_2 \int_{-\infty}^{\infty} du_3 h_1(u_1) h_2(u_2) h_3(u_3) \prod_{r=1}^{3} x(t-u_r) \qquad (3.16) \end{aligned}$$

Comparing (3.16) to (3.3), we see that $g_n(u_1, \ldots, u_n) = 0$ for all n except $n = 3$. At $n = 3$, we must multiply (3.16) by $\frac{1}{3!}$ in front and multiply by 3! inside the integrals:

$$\begin{aligned} y(t) &= \frac{1}{3!} \int_{-\infty}^{\infty} du_1 \int_{-\infty}^{\infty} du_2 \int_{-\infty}^{\infty} du_3 3! h_1(u_1) h_2(u_2) h_3(u_3) \prod_{r=1}^{3} x(t-u_r) \\ &= \frac{1}{3!} \int_{-\infty}^{\infty} du_1 \int_{-\infty}^{\infty} du_2 \int_{-\infty}^{\infty} du_3 \gamma_3(u_1, u_2, u_3) \prod_{r=1}^{3} x(t-u_r) \end{aligned}$$

where

$$\gamma_3(u_1, u_2, u_3) = 3! h_1(u_1) h_2(u_2) h_3(u_3)$$

This kernel γ_3 is, in general, *unsymmetric* since, for example, $h_1(u_1)h_2(u_2)$ will not equal $h_1(u_2)h_2(u_1)$, and thus $\gamma_3(u_1, u_2, u_3) \neq \gamma_3(u_2, u_1, u_3)$. To find a *symmetric* kernel g_3, we must symmetrize γ_3 by permuting its arguments in 3! ways, adding the results, and dividing by 3!. Thus, for the system of Figure 3.1,

$$\begin{aligned} g_3(u_1, u_2, u_3) &= \frac{1}{3!} \sum_{3!} \gamma_3(u_{(1)}, u_{(2)}, u_{(3)}) \\ &= \frac{1}{3!} \sum_{3!} 3! h_1(u_{(1)}) h_2(u_{(2)}) h_3(u_{(3)}) \\ &= \sum_{3!} h_1(u_{(1)}) h_2(u_{(2)}) h_3(u_{(3)}) \end{aligned}$$

Experience shows that the harmonic input method is generally easier when n is small; the direct expansion method seems to work best when the general formulae for high n are needed.

3.2.3 Powers of Transfer Functions

Often in the system equations it transpires that we have some power of the output variable $[y(t)]^l$. Bedrosian and Rice [Bed71] derive the n-fold Fourier transform of g_n in the Volterra series for $[y(t)]^l$ (l a positive integer) as

$$\begin{aligned} G_n^{(l)}(f_1, \ldots, f_n) &= l! \sum_{(v;l,n)} {\sum}'_N G_{v_1}(f_1, \ldots, f_{v_1}) \\ &\quad \times G_{v_2}(f_{v_1+1}, \ldots, f_{v_1+v_2}) \times \cdots \\ &\quad \times G_{v_l}(f_\mu, \ldots, f_n) \end{aligned} \tag{3.17}$$

Unfortunately the proof for this is rather involved so the interested reader is referred to [Bed71]. In much of the literature on Volterra series, the notation for G_n for general n varies widely and is confusing. In the author's opinion, Bedrosian and Rice's notation in [Bed71], reproduced in (3.17) and used throughout this thesis, is the clearest and quickest to write, particularly the ${\sum}'_N$ notation. An explanation of their notation follows.

The $(v; l, n)$ under the first summation sign denotes all sets v_i of l natural numbers (positive integers) such that

$$v_1 + \cdots + v_l = n, \ 1 \leq v_1 \leq v_2 \leq \cdots \leq v_l \tag{3.18}$$

Another way to say this is, $(v; l, n)$ represents *all partitions of n into l parts.* Each member of the partition accounts for one member of the sum $\sum_{(v;l,n)}$.

The second sum ${\sum}'_N$ extends over the so-called *non-identical products* that arise via permuting the subscripts of the f_i. Two products are *identical* if either (a) the f_i arguments of G_j in one product are the same in the other product, except for a permutation (like $G_2(f_1, f_2)$ and $G_2(f_2, f_1)$), or (b) the ordering of the terms in the products is the only thing that is changed (like $G_1(f_1)G_1(f_2)$ and $G_1(f_2)G_1(f_1)$). A simple combinatorial argument gives the number of non-identical products N as

$$N = \frac{n!}{v_1! \cdots v_n! r_1! \cdots r_k!} \tag{3.19}$$

where r_1 is the number of equal v_i in the first run of inequalities in

$$v_1 \leq v_2 \leq \cdots \leq v_l \tag{3.20}$$

from (3.18), r_2 is the number in the second run, and so on. $r_j = 1$ if a v_i is not equal to any others. The μ inside the G_{v_l} term in (3.17) is defined as

$$\mu = v_1 + \cdots + v_{l-1} + 1 = n - v_l + 1. \tag{3.21}$$

This all becomes much clearer with an example. To calculate $G_2^{(2)}(f_1, f_2)$, we see that $n = 2$ and $l = 2$. The only partition of n into l elements is $v_1 = v_2 = 1$. Thus, $r_1 = 2$, and

$$N = \frac{n!}{v_1! v_2! r_1!} = \frac{2!}{1!1!2!} = 1$$

(3.17) then becomes

$$\begin{aligned} G_2^{(2)}(f_1, f_2) &= 2! \sum\nolimits_1' G_1(f_1)G_1(f_2) \\ &= 2G_1(f_1)G_1(f_2) \end{aligned}$$

To find $G_3^{(2)}(f_1, f_2, f_3)$, where $n = 3$ and $l = 2$, again, there is only one partition of n into l parts: $v_1 = 1$, $v_2 = 2$ ($v_1 \leq v_2$ from (3.18) must hold). So $r_1 = 1$, $r_2 = 1$, and

$$N = \frac{n!}{v_1! v_2! r_1! r_2!} = \frac{3!}{1!2!1!1!} = 3$$

(3.17) is thus

$$\begin{aligned} G_3^{(2)}(f_1, f_2, f_3) &= 2! \sum\nolimits_3' G_1(f_1)G_2(f_2, f_3) \\ &= 2\left[G_1(f_1)G_2(f_2, f_3) + G_1(f_2)G_2(f_1, f_3) + G_1(f_3)G_2(f_1, f_2)\right] \\ &= 2\left\{[1][2,3] + [2][1,3] + [3][1,2]\right\} \end{aligned}$$

The last line is an abbreviated notation for the middle line. Finally, to calculate $G_4^{(2)}(f_1, f_2, f_3, f_4)$, $n = 4$ and $l = 2$, and now there are two partitions: $v_{11} = 1$, $v_{12} = 3$ and $v_{21} = 2$, $v_{22} = 2$. For each partition,

$$\begin{aligned} N_1 &= \frac{n!}{v_{11}!v_{12}!r_1!r_2!} = \frac{4!}{1!3!1!1!} = 4 \\ N_2 &= \frac{n!}{v_{21}!v_{22}!r_1!} = \frac{4!}{2!2!2!} = 3 \end{aligned}$$

giving an overall formula

$$\begin{aligned} G_4^{(2)}(f_1, f_2, f_3, f_4) &= 2! \textstyle\sum'_{N_1} G_1(f_1)G_3(f_2, f_3, f_4) + 2! \textstyle\sum'_{N_2} G_2(f_1, f_2)G_2(f_3, f_4) \\ &= 2\{[1][2,3,4] + [2][1,3,4] + [3][1,2,4] + [4][1,2,3] \\ &\quad + [1,2][3,4] + [1,3][2,4] + [1,4][2,3]\} \end{aligned}$$

An example of a circuit requiring use of equation (3.17) will be presented in the next chapter.

3.3 Output Spectrum from Volterra Kernels

For several different types of input signals, the spectrum of the output signal can be expressed in terms of the nth order transfer functions. The most important types of inputs in this thesis are sinusoids and sums of sinusoids.

If the input is a single sinusoid $x(t) = V_a \cos(\omega_a t)$, $\omega_a = 2\pi f_a$, then the spectrum of the output can be expressed as [Bed71]

$$y(t) = \sum_{n=1}^{\infty} \left(\frac{V_a}{2}\right)^n \sum_{k=0}^{n} \frac{\exp[j(2k-n)\omega_a t]}{k!(n-k)!} G_{k,n-k}(f_a) \qquad (3.22)$$

where $G_{k,n-k}(f_a)$ is shorthand for $G_n(f_1, \ldots, f_n)$ with the first k of the f_i equal to $+f_a$ and the remaining $n-k$ equal to $-f_a$. (Technically, because the G_n are symmetric, *any* k of the f_i can be set to $+f_a$ and the remaining $n-k$ to $-f_a$. It is least confusing to set the *first* k to $+f_a$.) Expanding this for a few different values of $(2k-n)$ leads to

$$\begin{aligned}
y(t) \;=\; & [\frac{V_a^2}{2}G_2(f_a,-f_a)+\cdots] \\
& +e^{j\omega_a t}[\frac{V_a}{2}G_1(f_a)+\frac{V_a^3}{16}G_3(f_a,f_a,-f_a)+\cdots] \\
& +e^{j2\omega_a t}[\frac{V_a^2}{8}G_2(f_a,f_a)+\cdots] \\
& +e^{j3\omega_a t}[\frac{V_a^3}{48}G_3(f_a,f_a,f_a)+\cdots]+\cdots \\
& +e^{-j\omega_a t}[\frac{V_a}{2}G_1(-f_a)+\frac{V_a^3}{16}G_3(-f_a,-f_a,f_a)+\cdots] \\
& +e^{-j2\omega_a t}[\frac{V_a^2}{8}G_2(-f_a,-f_a)+\cdots] \\
& +e^{-j3\omega_a t}[\frac{V_a^3}{48}G_3(-f_a,-f_a,-f_a)+\cdots]+\cdots \qquad (3.23)
\end{aligned}$$

Thus, at the output, the contributors to the fundamental $e^{j\omega_a t}$ and the odd harmonics are the odd G_n only. Even harmonics are made up of sums of even G_n only. In the general term $e^{jN\omega_a t}$, the f_i arguments to $G_n(f_1,\ldots,f_n)$ always sum to Nf_a. If the input has a phase angle ϕ so that $x(t)=V_a\cos(\omega_a t+\phi)$, then all the $\omega_a t$ in (3.23) must be replaced with $\omega_a t+\phi$.

For a two-input sinusoid $x(t)=V_a\cos(\omega_a t)+V_b\cos(\omega_b t)$, the term at frequency $N\omega_a+M\omega_b$, $N,M\geq 0$ is given by

$$e^{j(N\omega_a+M\omega_b)t}\sum_{l=0}^{\infty}\sum_{k=0}^{\infty}\frac{(V_a/2)^{2l+N}(V_b/2)^{2k+M}}{(N+l)!l!(M+k)!k!}G_{N+l,l;M+k,k}(f_a,f_b) \qquad (3.24)$$

where $G_{N+l,l;M+k,k}(f_a,f_b)$ is $G_n(f_1,\ldots,f_n)$ with

$$\begin{aligned}
N+2l+M+2k &= n, \\
\text{first } N+l \text{ of } f_i &= +f_a, \\
\text{next } l \text{ of } f_i &= -f_a, \qquad (3.25) \\
\text{next } M+k \text{ of } f_i &= +f_b, \\
\text{last } k \text{ of } f_i &= -f_b.
\end{aligned}$$

For $N < 0$, the signs of f_a in (3.25) are reversed; for $M < 0$, the signs of f_b are reversed.

Lastly, for a three-tone excitation $x(t) = V_a \cos(\omega_a t) + V_b \cos(\omega_b t) + V_c \cos(\omega_c t)$, the output term at $N\omega_a + M\omega_b + L\omega_c$, $N, M, L \geq 0$ is

$$e^{j(N\omega_a+M\omega_b+L\omega_c)t}\sum_{l=0}^{\infty}\sum_{k=0}^{\infty}\sum_{i=0}^{\infty}\frac{(V_a/2)^{2l+N}(V_b/2)^{2k+M}(V_c/2)^{2i+L}}{(N+l)!l!(M+k)!k!(L+i)!i!}G_{N+l,l;M+k,k;L+i,i}(f_a, f_b, f_c) \tag{3.26}$$

where $G_{N+l,l;M+k,k;L+i,i}(f_a, f_b, f_c)$ is $G_n(f_1, \ldots, f_n)$ with

$$\begin{aligned} N + 2l + M + 2k + L + 2i &= n, \\ \text{first } N + l \text{ of } f_i &= +f_a, \\ \text{next } l \text{ of } f_i &= -f_a, \\ \text{next } M + k \text{ of } f_i &= +f_b, \\ \text{next } k \text{ of } f_i &= -f_b, \\ \text{next } L + i \text{ of } f_i &= +f_c, \\ \text{last } i \text{ of } f_i &= -f_c. \end{aligned}$$

For $N < 0, M < 0, L < 0$, the signs of f_a, f_b, f_c, respectively, are reversed.

The output spectrum for several other types of input has been derived in [Bed71], namely, Gaussian noise, sine wave plus Gaussian noise, and random pulse train.

3.4 Applicability of Volterra Series

Now that Volterra series have been introduced, we may answer the question first posed in §2.2.5.

3.4.1 When Volterra Series Are Good

Volterra series give a fairly simple, algebraically tractable method of calculating small distortion terms in weakly nonlinear systems.

For example, consider again the first harmonic of the expansion of $y(t)$ in (3.23), shown here to three terms:

$$e^{j\omega_a t}[\frac{V_a}{2}G_1(f_a) + \frac{V_a^3}{16}G_3(f_a, f_a, -f_a) + \frac{V_a^5}{384}G_5(f_a, f_a, f_a, -f_a, -f_a) + \cdots] \qquad (3.27)$$

In a weakly nonlinear system, the first term $\frac{V_a}{2}G_1(f_a)$ will dominate for small inputs. This term is the familiar linear transfer function, and it corresponds to the linear gain of the device. As the input gets larger, the second term $\frac{V_a^3}{16}G_3(f_a, f_a, -f_a)$ starts to contribute more, and it represents the gain compression or gain expansion of the system. Thus, Volterra series give us a *direct* method of calculating, for example, the 1-dB compression point of a system. To derive the third-order intercept point IP_3, we need simply equate the first two terms in (3.27) and solve for V_a.

Furthermore, the desensitization and intermodulation terms in §2.2.3 are now directly expressible using equation (3.26): for three tones (f_a, f_b, f_c) with amplitudes (V_a, V_b, V_c), the desensitization terms can be found to be $\frac{V_a V_b^2}{8}G_3(f_a, f_b, -f_b)$ and $\frac{V_a V_c^2}{8}G_3(f_a, f_c, -f_c)$, and the intermodulation term is $\frac{V_b^2 V_c}{16}G_3(f_b, f_b, -f_c)$.

What about deriving the kernels for an actual system? It transpires that for many real-life systems, an algebraic expression for the kernels *can* be derived. Narayanan [Nar70], for example, developed a nonlinear model of the bipolar transistor and quite successfully applied Volterra series to it to calculate distortion in amplifiers.

Even when algebraic expressions are too complex or unknown, Volterra series can sometimes still be used. Volterra kernels can be extracted from numerical simulations and/or measured data [Boyd83]. This will be further demonstrated in Chapter 5 and Chapter 6.

3.4.2 When Volterra Series Are Bad

In problems using a Volterra series approach, the results are expressed as sums of infinite numbers of terms, like equation (3.27). These sums will either converge or diverge. This has several implications.

1. If the sum converges, it will do so to a single value. This means that in systems with multiple possible output values (like systems with hysteresis), the best we can hope for is convergence to one of the possible values.

2. If the sum converges, then the single value it converges to will be the steady-state value. Any simulations of system behavior *must* be carried out to the point where transients have died out, or else comparing results with Volterra series calculations will be of little value.

3. If the sum converges, we hope it will do so fairly rapidly. Otherwise, the time to compute the sum increases exponentially. In (3.27), we expect the linear transfer function term, G_1, to dominate. The G_3 term should be very much smaller than the G_1 term, and the G_5 term must be very much smaller than the G_3 term. G_5 takes longer to compute than G_3; if G_5 is significant relative to G_3, we must compute G_7 to see if the sum converges, which takes longer still. And so on.

4. If the sum diverges, obviously Volterra series do not give us much quantitative information.

These points all revolve around the notion of the nonlinearity's strength. If it is "weak enough", then our infinite sums will converge and converge rapidly; G_3 *will* be much smaller than G_1, and G_5 will be so small that it is negligible. If the nonlinearity is "too strong", the sums will either converge but require a long time to compute, or they will diverge. Thus Volterra series are impractical in strongly nonlinear problems.

3.4.3 Volterra Series in This Thesis

How do we know how strong a nonlinearity we are dealing with? The path to answering this question is paved with heavy mathematics. The reader is invited to peruse [Boyd85], [Sand83a], [Sand83b], [Sand83c] for a sampling.

This thesis skims over the mathematical foundations of Volterra series in favor of applying them to practical problems. Such an approach may seem haphazard, but the author feels it is a reasonable method of proceeding for the following reasons:

1. Chapter 4 of this thesis is devoted to finding the conditions that make the nonlinearity strong. G_1, G_3, and G_5 are all calculated and their sum checked for convergence. Thereafter, every attempt is made to stay within the weakly nonlinear region of operation.

2. Volterra series are not used alone: calculations are often supplemented with numerical simulations and practical measurements.

3. Despite lack of adherence to strict mathematical theory, many, many authors have applied Volterra series to real systems with more than moderate success. This is a good indication that the method is quite robust, and the author feels no qualms in following a tried and true path trodden safely by many before.

Chapter 4

Analysis of a Filter using Volterra Series

4.1 Filter Circuit

The filtering circuit that will be investigated here is shown in Figure 4.1. This particular structure was chosen because something quite similar might be used in a modern integrated circuit [Sch90]. The active devices are *transconductance amplifiers*, or TAs, that turn the voltage difference between their two inputs into a current output. That is, $i = f(v_+ - v_-)$ for each device, where f is some function, as yet unspecified. The TAs are assumed to have infinite input and output impedances.

Using Kirchoff's current law, we may write the time-domain nodal equations at v_1 and v_2:

$$\begin{aligned} f_i(v_i) + f_1(v_2) &= C_1\frac{dv_1}{dt} + \frac{v_1}{R_1} \\ f_2(-v_1) &= C_2\frac{dv_2}{dt} \end{aligned}$$

Upon rearrangement, these become

$$\begin{aligned} \tfrac{dv_1}{dt} &= \tfrac{1}{C_1}[-\tfrac{v_1}{R_1} + f_i(v_i) + f_1(v_2)] \\ \tfrac{dv_2}{dt} &= \tfrac{1}{C_2}f_2(-v_1) \end{aligned} \qquad (4.1)$$

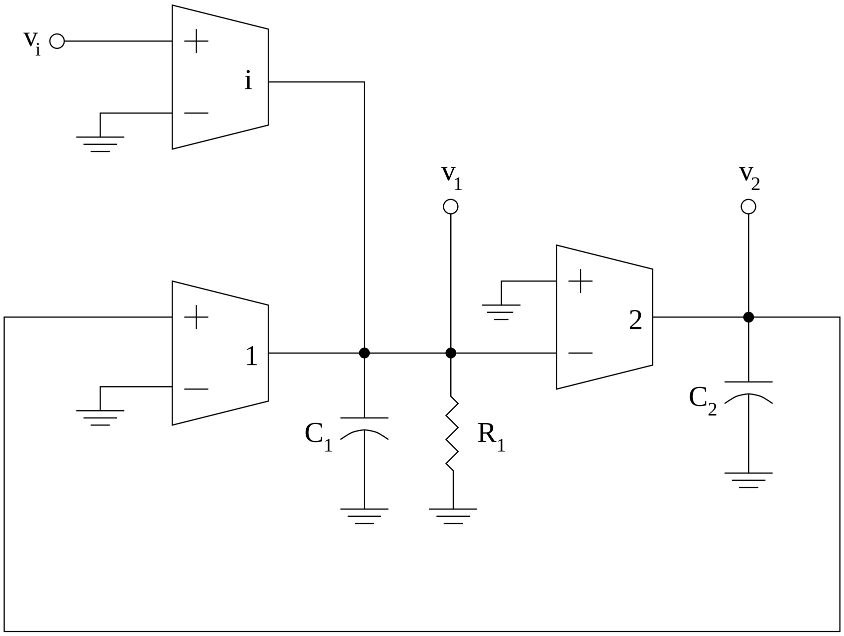


Figure 4.1: Filter circuit.

4.1.1 Linear Circuit Equations

If we assume the TAs are linear, then $f(v_+ - v_-) = g_m(v_+ - v_-)$ where g_m is a constant called the *transconductance*. Substituting this in (4.1) and taking Laplace transforms,

$$\begin{aligned} sV_1 &= -\frac{1}{R_1C_1}V_1 + \frac{g_{mi}}{C_1}V_i + \frac{g_{m1}}{C_1}V_2 \\ sV_2 &= -\frac{g_{m2}}{C_2}V_1 \end{aligned}$$

We can solve these for $\frac{V_1}{V_i}$ and $\frac{V_2}{V_i}$ to arrive at the system transfer functions

$$\frac{V_1}{V_i} = \frac{\frac{g_{mi}}{C_1}s}{s^2 + \frac{1}{R_1C_1}s + \frac{g_{m1}g_{m2}}{C_1C_2}} \tag{4.2}$$

$$\frac{V_2}{V_i} = \frac{\frac{-g_{mi}g_{m2}}{C_1C_2}}{s^2 + \frac{1}{R_1C_1}s + \frac{g_{m1}g_{m2}}{C_1C_2}} \tag{4.3}$$

v_1 and v_2 will be recognized as *band pass* and *low pass* outputs, respectively. The corner frequency of the low pass filter and the center frequency of the band pass filter

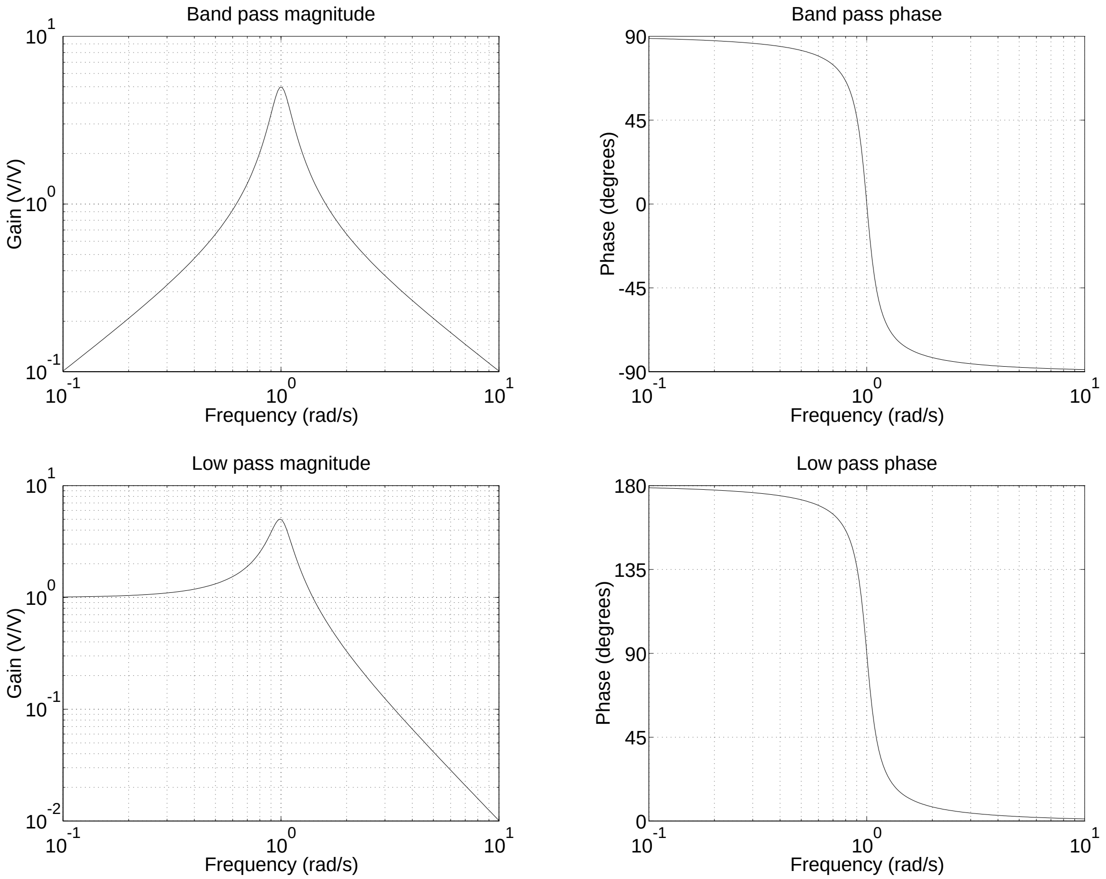


Figure 4.2: Band pass (top) and low pass (bottom) magnitude and phase graphs.

are both given by $\omega_0^2 = \frac{g_{m1}g_{m2}}{C_1C_2}$, and the filter quality factor is $Q = R_1C_1\omega_0$. Graphs of the magnitude and phase of these transfer functions are shown in Figure 4.2, with $\omega_0 = 1$, $Q = 5$, and the numerators set to one. In radio reception circuits we are more interested in the band pass filter, and so we will concern ourselves with the voltage v_1.

4.1.2 Nonlinear Circuit Equations

At this point, we may well ask ourselves what sort of nonlinearity we should assume in this circuit. Obviously, once the circuit is manufactured in real life, the nature of its nonlinearities will depend on a great many factors.

In this thesis, the passive components of the filter will be assumed ideal, and the TAs will be assumed to contain a cubic nonlinearity. This will make the current-voltage equation

$$i = g_m(v_+ - v_-) + \epsilon(v_+ - v_-)^3 \tag{4.4}$$

This choice is for two reasons. First, it does not complicate the equations to the point where they are algebraically intractable. Second, it is a "realistic" choice in terms of circuit design in that were the TAs to be fabricated as part of an integrated circuit, they would most likely contain a bipolar or MOS differential pair at their input. The i-v characteristic of a differential pair might have odd symmetry, and so will be made up of odd Taylor series powers only. For example, in a bipolar differential pair the i-v characteristic will be a hyperbolic tangent function, $i = \tanh(kv)$, k some constant; the Taylor series expansion for this can be expressed as

$$i = kv - \frac{(kv)^3}{3} + \frac{(kv)^5}{15} - O((kv)^7) \tag{4.5}$$

which does indeed contain only odd Taylor series powers. Moreover, if kv is small enough, the terms after $(kv)^3$ can be neglected. Hence, (4.4) becomes a good approximation to (4.5).

Is it reasonable not to include a dc offset or a square term in (4.4)? It will simplify the algebra, but how realistic is this omission? It can be shown that polynomial nonlinearities with even powers lead to harmonics at even powers of the fundamental. That is, a dc offset or a square term will contribute to the output harmonics at dc, $2\omega_0$, $4\omega_0$, etc. Odd powers lead to harmonics at odd multiples of the fundamental, i.e., ω_0, $3\omega_0$, etc. Since we are concerned only with input tones close to ω_0, the center frequency of the band pass filter, and we are interested in output tones close to ω_0, polynomial nonlinearities with even powers are far less important than those with odd powers because the even powers contribute only to harmonics far away from ω_0. As long as the input has no dc offset — an assumption we will also make — we can then safely omit even power terms in (4.4).

Substituting (4.4) for each TA in (4.1) gives

$$\begin{aligned} C_1\frac{dv_1}{dt} &= -\frac{v_1}{R_1} + g_{mi}v_i + \epsilon_i v_i^3 + g_{m1}v_2 + \epsilon_1 v_2^3 && (4.6)\\ C_2\frac{dv_2}{dt} &= -g_{m2}v_1 - \epsilon_2 v_1^3 && (4.7)\end{aligned}$$

Solving (4.7) for v_2 yields

$$v_2 = -\frac{g_{m2}}{C_2}\int v_1 dt - \frac{\epsilon_2}{C_2}\int v_1^3 dt \qquad (4.8)$$

Substituting (4.8) in (4.6) gives

$$\begin{aligned} C_1\frac{dv_1}{dt} &= -\frac{v_1}{R_1} + g_{mi}v_i + \epsilon_i v_i^3 - \frac{g_{m1}g_{m2}}{C_2}\int v_1 dt - \frac{\epsilon_1 g_{m2}}{C_2}[\int v_1 dt]^3 \\ &\quad - \epsilon_1\left[-\frac{g_{m2}}{C_2}\int v_1 dt - \frac{\epsilon_2}{C_2}\int v_1^3 dt\right]^3 && (4.9)\end{aligned}$$

Letting $v_i = x$ (the input) and $v_1 = y$ (the output) in (4.9), expanding the cubed term, and collecting terms in a way that will be easy to handle gives an overall equation of

$$\begin{aligned} C_1\frac{dy}{dt} + \frac{y}{R_1} + \frac{g_{m1}g_{m2}}{C_2}\int y dt &= g_{mi}x + \epsilon_i x^3 \\ &- \frac{\epsilon_2 g_{m1}}{C_2}[\int y^3 dt] \\ &- \frac{\epsilon_1 g_{m2}^3}{C_2^3}[\int y dt]^3 \\ &- \frac{3\epsilon_1\epsilon_2 g_{m2}^2}{C_2^3}[\int y dt]^2[\int y^3 dt] \\ &- \frac{3\epsilon_1\epsilon_2^2 g_{m2}}{C_2^3}[\int y dt][\int y^3 dt]^2 \\ &- \frac{\epsilon_1\epsilon_2^3}{C_2^3}[\int y^3 dt]^3 && (4.10)\end{aligned}$$

The linear terms in y have been written on the LHS of (4.10), and the terms with x and the nonlinear terms in y have been written on the RHS.

4.2 Volterra Transfer Functions

Equation (4.10) can now be used to write the Volterra transfer functions, which we will label M_n, for the filter with nonlinear TAs. Since (4.10) is not explicitly solved for $y(t)$, we must assume that $y(t)$ can be expanded into a series where the coefficient of $\exp[j(\omega_1 + \cdots + \omega_n)t]$ is $M_n(f_1, \ldots, f_n)$. Then, for the terms on the RHS such as $[\int y dt]^3$, we must use the results of §3.2.3 to express the coefficient of $\exp[j(\omega_1 + \cdots + \omega_n)t]$ in terms of $M_n^{(3)}$. Finally, all the results may be substituted and solved for $M_n(f_1, \ldots, f_n)$. The Volterra transfer functions for each term in (4.10) are calculated in the subsections below.

4.2.1 $C_1 \frac{dy}{dt} + \frac{y}{R_1} + \frac{g_{m1} g_{m2}}{C_2} \int y dt$

Taking each term one at a time, starting with the middle one, if the nth coefficient of $y(t)$ is $M_n(f_1, \ldots, f_n)$, then the nth coefficient of $\dfrac{y(t)}{R_1}$ is clearly $\dfrac{1}{R_1} M_n(f_1, \ldots, f_n)$.

For the first term, we know that $y(t)$ will be composed of terms of the form

$$\exp[j(\omega_1 + \cdots + \omega_n)t] M_n(f_1, \ldots, f_n) \tag{4.11}$$

where the coefficient of $\exp[j(\omega_1 + \cdots + \omega_n)t]$ is $M_n(f_1, \ldots, f_n)$. Thus, the coefficient of $C_1 \frac{dy}{dt}$ can be found by differentiating (4.11) and multiplying by C_1, which means the coefficient of $\exp[j(\omega_1 + \cdots + \omega_n)t]$ will be

$$C_1 [j(\omega_1 + \cdots + \omega_n)] M_n(f_1, \ldots, f_n)$$

By a similar argument, the coefficient of $\exp[j(\omega_1 + \cdots + \omega_n)t]$ from $\frac{g_{m1} g_{m2}}{C_2} \int y dt$ will be

$$\frac{g_{m1} g_{m2}}{C_2} \frac{1}{[j(\omega_1 + \cdots + \omega_n)]} M_n(f_1, \ldots, f_n)$$

Combining these results, the LHS of (4.10) will become

$$\left[C_1 \sum_{i=1}^{n} j\omega_i + \frac{1}{R_1} + \frac{g_{m1} g_{m2}}{C_2 \sum_{i=1}^{n} j\omega_i} \right] M_n(f_1, \ldots, f_n) \tag{4.12}$$

for general n. Both sides of (4.10) can be divided by the term inside square brackets in (4.12) yielding a closed-form expression for $M_n(f_1, \ldots, f_n)$ for the overall filter. First, however, the Volterra transfer functions for the terms on the RHS of (4.10) must be found.

4.2.2 $g_{mi}x + \epsilon_i x^3$

Here, $x(t)$ is set to the sum of n exponentials and the coefficient of $\exp[j(\omega_1 + \cdots + \omega_n)t]$ extracted. It is not difficult to see that for $n = 1$, the coefficient will be g_{mi}; for $n = 2$, the coefficient will be zero; for $n = 3$, the coefficient will be $6\epsilon_i$; and for $n > 3$, the coefficient will again be zero.

4.2.3 $\frac{g_{m1}\epsilon_2}{C_2} \int y^3 dt$

As stated at the beginning of the section, if we assume that $y(t)$ has a Volterra transfer function $M_n(f_1, \ldots, f_n)$, then we can use the results from §3.2.3 to say that the coefficient of y^3 will be $M_n^{(3)}(f_1, \ldots, f_n)$.

The integral sign might appear problematic at first, but as in §4.2.1 with $\int y dt$, all an integration does is multiply a coefficient by $[\sum_{i=1}^n j\omega_i]^{-1}$. The coefficient for this term will be

$$\frac{g_{m1}\epsilon_2}{C_2 \sum_{i=1}^n j\omega_i} M_n^{(3)}(f_1, \ldots, f_n)$$

4.2.4 $\frac{\epsilon_1 g_{m2}^3}{C_2^3}[\int y dt]^3$

This term is rather like the one in §4.2.3, and we expect the coefficient will involve $M_n^{(3)}$, except here the integration is done before the cube is taken. Although it is not obvious at first, this means that when $M_n^{(3)}$ is written (following §3.2.3) as $M_{v_1} \times M_{v_2} \times M_{v_3}$ where $v_1 + v_2 + v_3 = n$, the whole product is multiplied by

$$\left(\sum_{i=1}^{v_1} j\omega_i\right)^{-1} \left(\sum_{i=v_1+1}^{v_1+v_2} j\omega_i\right)^{-1} \left(\sum_{i=\mu}^{n} j\omega_i\right)^{-1}$$

where $\mu = v_1 + v_2 + 1 = n - v_3 + 1$ as defined in §3.2.3. Furthermore, these must be included inside the $\sum_N'$ summand.

Thus, the coefficient of this term can be written

$$\frac{\epsilon_1 g_{m2}^3}{C_2^3} 3! \sum_{(v;3,n)} \sum_N' \frac{M_{v_1}(f_1, \ldots, f_{v_1}) M_{v_2}(f_{v_1+1}, \ldots, f_{v_1+v_2}) M_{v_3}(f_\mu, \ldots, f_n)}{\sum_{i=1}^{v_1} j\omega_i \sum_{i=v_1+1}^{v_1+v_2} j\omega_i \sum_{i=\mu}^n j\omega_i}$$

or, in an abbreviated (yet hopefully clear) notation,

$$\frac{\epsilon_1 g_{m2}^3}{C_2^3} 3! \sum_{(v;3,n)} {\sum}'_N \frac{M_{v_1} M_{v_2} M_{v_3}}{\sum_{v_1} j\omega_i \sum_{v_2} j\omega_i \sum_{v_3} j\omega_i}$$

To clarify this further, the coefficients are written out in full for a few small n. For $n = 1$ and $n = 2$, the coefficient is zero, since there is no way for $v_1 + v_2 + v_3 < 3$ to be satisfied when $v_i \geq 1$, $i = 1, 2, 3$, must also be satisfied. For $n = 3$, the only partition is $v_1 = v_2 = v_3 = 1$, making $N = \frac{3!}{1!1!1!3!} = 1$ and the coefficient

$$\frac{\epsilon_1 g_{m2}^3}{C_2^3} 3! \frac{M_1(f_1) M_1(f_2) M_1(f_3)}{(j\omega_1)(j\omega_2)(j\omega_3)}$$

For $n = 4$, the only partition is $(v_1, v_2, v_3) = (1, 1, 2)$, making $N = \frac{4!}{1!1!2!2!} = 6$ and the coefficient

$$\begin{aligned}
& \frac{\epsilon_1 g_{m2}^3}{C_2^3} 3! \quad {\sum}'_6 \frac{M_1(f_1) M_1(f_2) M_2(f_3,f_4)}{(j\omega_1)(j\omega_2)(j\omega_3+j\omega_4)} \\
= & \frac{\epsilon_1 g_{m2}^3}{C_2^3} 6 \, [\quad \frac{M_1(f_1) M_1(f_2) M_2(f_3,f_4)}{(j\omega_1)(j\omega_2)(j\omega_3+j\omega_4)} + \frac{M_1(f_1) M_1(f_3) M_2(f_2,f_4)}{(j\omega_1)(j\omega_3)(j\omega_2+j\omega_4)} \\
& \qquad + \frac{M_1(f_1) M_1(f_4) M_2(f_2,f_3)}{(j\omega_1)(j\omega_4)(j\omega_2+j\omega_3)} + \frac{M_1(f_2) M_1(f_3) M_2(f_1,f_4)}{(j\omega_2)(j\omega_3)(j\omega_1+j\omega_4)} \\
& \qquad + \frac{M_1(f_2) M_1(f_4) M_2(f_1,f_3)}{(j\omega_2)(j\omega_4)(j\omega_1+j\omega_3)} + \frac{M_1(f_3) M_1(f_4) M_2(f_1,f_2)}{(j\omega_3)(j\omega_4)(j\omega_1+j\omega_2)} \,]
\end{aligned}$$

4.2.5 $\frac{3g_{m2}^2 \epsilon_1 \epsilon_2}{C_2^3} [\int y dt]^2 [\int y^3 dt]$

The Volterra coefficients start to become more complicated now. Essentially, we have a product of three terms:

$$[\int y dt][\int y dt][\int y^3 dt]$$

This suggests a form involving M_{v_1}, M_{v_2}, and $M_{v_3}^{(3)}$ where $v_1 + v_2 + v_3 = n$. After careful examination, the coefficient will be seen to be

$$\frac{3g_{m2}^2 \epsilon_1 \epsilon_2}{C_2^3} \sum_{(v;3,n)} \left({\sum}'_{N_{12}} \frac{2! M_{v_1} M_{v_2}}{\sum_{v_1} j\omega_i \sum_{v_2} j\omega_i} \right) \left(\frac{M_{v_3}^{(3)}}{\sum_{v_3} j\omega_i} \right)$$

The 2! inside the first brackets is to allow v_1 and v_2 to be exchanged, and

$$N_{12} = \frac{n!}{v_1! v_2! r!} \text{ where } r = \begin{cases} 1, & v_1 \neq v_2 \\ 2, & v_1 = v_2 \end{cases}$$

Here it is implied that $v_3 \geq 3$ since if $v_3 = 1$ or $v_3 = 2$, then $M_{v_3}^{(3)} = 0$ from §3.2.3.

4.2.6 $\frac{3g_{m2}\epsilon_1\epsilon_2^2}{C_2^3}[\int y dt][\int y^3 dt]^2$

By similar reasoning to §4.2.5, the coefficient for this term is

$$\frac{3g_{m2}\epsilon_1\epsilon_2^2}{C_2^3}\sum_{(v;3,n)}\left({\sum}'_{N_1}\frac{M_{v_1}}{\sum_{v_1} j\omega_i}\right)\left({\sum}'_{N_{23}}\frac{2!M_{v_2}^{(3)}M_{v_3}^{(3)}}{\sum_{v_2} j\omega_i\sum_{v_3} j\omega_i}\right)$$

where

$$\begin{aligned} N_1 &= \frac{n!}{v_1!(n-v_1)!} \\ N_2 &= \frac{(n-v_1)!}{v_2!v_3!r!} \text{ where } r = \begin{cases} 1, & v_2 \neq v_3 \\ 2, & v_2 = v_3 \end{cases} \end{aligned}$$

As in the previous section, here the implied conditions are $v_2 \geq 3$ and $v_3 \geq 3$. Otherwise, the coefficient is zero.

4.2.7 $\frac{\epsilon_1\epsilon_2^3}{C_2^3}[\int y^3 dt]^3$

The coefficient for this last term can be written (no doubt with great relief) as

$$\frac{\epsilon_1\epsilon_2^3}{C_2^3}\sum_{(v;3,n)}{\sum}'_N\frac{3!M_{v_1}^{(3)}M_{v_2}^{(3)}M_{v_3}^{(3)}}{\sum_{v_1} j\omega_i\sum_{v_2} j\omega_i\sum_{v_3} j\omega_i}$$

As in the previous two sections, we require $v_1 \geq 3$, $v_2 \geq 3$, $v_3 \geq 3$. Since $n = v_1 + v_2 + v_3$, this coefficient is non-zero only for $n \geq 9$.

4.2.8 Including All Terms

Table 4.1 summarizes these results. A few comments are in order.

1. It is a fair amount of work to calculate the terms in the Table. It would be useful to be able to teach a symbolic algebra program to find the terms for a general equation for us. Such an effort is beyond the scope of this thesis but would be useful for future work.

Table 4.1: Volterra coefficients.

Term	Coefficient for general n
$C_1\frac{dy}{dt}+\frac{1}{R_1}y+\frac{g_{m1}g_{m2}}{C_2}\int y\,dt$	$(C_1\sum j\omega_i+\frac{1}{R_1}+\frac{g_{m1}g_{m2}}{C_2\sum j\omega_i})M_n$
$g_{mi}x+\epsilon_i x^3$	$\begin{cases} g_{mi}, & n=1 \\ 6\epsilon_i, & n=3 \\ 0, & \text{otherwise} \end{cases}$
$\frac{\epsilon_2 g_{m1}}{C_2}[\int y^3 dt]$	$\frac{\epsilon_2 g_{m1}}{C_2\sum j\omega_i}M_n^{(3)}$
$\frac{\epsilon_1 g_{m2}^3}{C_2^3}[\int y\,dt]^3$	$\frac{\epsilon_1 g_{m2}^3}{C_2^3}3!\sum_{(v;3,n)}{\sum}'_N\frac{M_{v_1}M_{v_2}M_{v_3}}{\sum_{v_1}j\omega_i\sum_{v_2}j\omega_i\sum_{v_3}j\omega_i}$
$\frac{3g_{m2}^2\epsilon_1\epsilon_2}{C_2^3}[\int y\,dt]^2[\int y^3 dt]$	$\frac{3g_{m2}^2\epsilon_1\epsilon_2}{C_2^3}\sum_{(v;3,n)}\left(\sum{}'_{N_{12}}\frac{2!M_{v_1}M_{v_2}}{\sum_{v_1}j\omega_i\sum_{v_2}j\omega_i}\right)\left(\frac{M_{v_3}^{(3)}}{\sum_{v_3}j\omega_i}\right)$
$\frac{3g_{m2}\epsilon_1\epsilon_2^2}{C_2^3}[\int y\,dt][\int y^3 dt]^2$	$\frac{3g_{m2}\epsilon_1\epsilon_2^2}{C_2^3}\sum_{(v;3,n)}\left({\sum}'_{N_1}\frac{M_{v_1}}{\sum_{v_1}j\omega_i}\right)\left({\sum}'_{N_{23}}\frac{2!M_{v_2}^{(3)}M_{v_3}^{(3)}}{\sum_{v_2}j\omega_i\sum_{v_3}j\omega_i}\right)$
$\frac{\epsilon_1\epsilon_2^3}{C_2^3}[\int y^3 dt]^3$	$\frac{\epsilon_1\epsilon_2^3}{C_2^3}\sum_{(v;3,n)}{\sum}'_N\frac{3!M_{v_1}^{(3)}M_{v_2}^{(3)}M_{v_3}^{(3)}}{\sum_{v_1}j\omega_i\sum_{v_2}j\omega_i\sum_{v_3}j\omega_i}$

2. Let us write out the first few terms of the series explicitly. For $n = 1$, only the first two rows of Table 4.1 are not zero:

$$\begin{aligned} [C_1 j\omega_1 + \frac{1}{R_1} + \frac{g_{m1}g_{m2}}{C_2 j\omega_1}]M_1(f_1) &= g_{mi} \\ M_1(f_1) &= \frac{g_{mi}}{C_1 j\omega_1 + \frac{1}{R_1} + \frac{g_{m1}g_{m2}}{C_2 j\omega_1}} \end{aligned} \tag{4.13}$$

Multiplying top and bottom by $\dfrac{j\omega_1}{C_1}$ and letting $j\omega_1 = s$ reveals

$$M_1(f_1) = \frac{\frac{g_{mi}}{C_1}s}{s^2 + \frac{1}{R_1C_1}s + \frac{g_{m1}g_{m2}}{C_1C_2}} \tag{4.14}$$

(4.14) is the same as the linear band pass transfer function in (4.2). This is the expected result: if the system is linear, the Volterra approach should yield the same transfer function as the usual frequency-domain analysis.

For $n = 2$, the only non-zero term is the one on the LHS; everything on the RHS is zero.

$$\begin{aligned} [C_1(j\omega_1 + j\omega_2) + \frac{1}{R_1} + \frac{g_{m1}g_{m2}}{C_2(j\omega_1 + j\omega_2)}]M_2(f_1, f_2) &= 0 \\ M_2(f_1, f_2) &= 0 \end{aligned} \tag{4.15}$$

This result is interesting, for it means that for *any* even n, M_n for this filter is zero. The result follows from looking at the last five rows of the Table: all of them involve partitioning n into three natural numbers v_1, v_2, v_3. When n is even, at least one of v_1, v_2, v_3 must be even. We can use mathematical induction to show that for all even n, $M_n = 0$. Here is a proof for small even n.

n=2 $M_2 = 0$ from (4.15).

n=4 The only partition is $(v_1, v_2, v_3) = (1, 1, 2)$. So all terms will be made up of sums of the product $M_1M_1M_2$. But $M_2 = 0$. Thus $M_4 = 0$.

n=6 Partitions are $(v_1, v_2, v_3) = (1, 1, 4)$, $(1, 2, 3)$, and $(2, 2, 2)$. All of these contain M_2 or M_4, which are both zero. Thus $M_6 = 0$.

And so on. For n even, each M_n depends on lower-order M_m where m is even and $m < n$. By induction, then, $M_n = 0$ for all even n.

Lastly, for $n = 3$, the non-zero terms are the ones in the first four rows of Table 4.1. The only partition of $n = 3$ is $(v_1, v_2, v_3) = (1, 1, 1)$, giving

$$\begin{aligned}(C_1{\textstyle\sum_3} j\omega_i + \frac{1}{R_1} + \frac{g_{m1}g_{m2}}{C_2\sum_3 j\omega_i}) &\times \\ M_3(f_1, f_2, f_3) &= 6\epsilon_i \\ &- \frac{\epsilon_2 g_{m1}}{C_2\sum_3 j\omega_i} M_3^{(3)}(f_1, f_2, f_3) \\ &- \frac{\epsilon_1 g_{m2}^3}{C_2^3} {\sum}'_N \frac{3! M_1(f_1)M_1(f_2)M_1(f_3)}{(j\omega_1)(j\omega_2)(j\omega_3)} \qquad (4.16)\end{aligned}$$

Expanding the second term on the RHS and recognizing that $N = 1$ in the third term, (4.16) becomes

$$M_3(f_1, f_2, f_3) = \frac{6\epsilon_i - 6M_1(f_1)M_1(f_2)M_1(f_3)\left[\frac{\epsilon_2 g_{m1}}{C_2\sum_3 j\omega_i} + \frac{\epsilon_1 g_{m2}^3}{C_2^3\prod_3 j\omega_i}\right]}{C_1\sum_3 j\omega_i + \frac{1}{R_1} + \frac{g_{m1}g_{m2}}{C_2\sum_3 j\omega_i}} \qquad (4.17)$$

M_3 thus depends on lower-order M_n, in this case, M_1. The same pattern holds true for higher n: M_5 depends on M_3 and M_1, M_7 depends on M_5, M_3, and M_1, and so on. M_3 also depends on the i-v characteristic coefficients g_m and ϵ because it is derived from the interconnection of the three nonlinear TAs — that is, we are deriving the Volterra series for a circuit from the kernels of its subcircuits.

3. The last three rows in Table 4.1 are non-zero for $n \geq 5$, $n \geq 7$, and $n \geq 9$, respectively. As discussed in §3.4.2, the effort of computing M_5, M_7, etc. increases exponentially. We will be concentrating on M_1 and M_3 for much of the thesis except in §4.4 where M_5 will also be computed to check the nonlinearity

strength. The extra effort involved in finding M_7 and higher is not justified since by the time they become numerically significant the nonlinearity is probably too strong for Volterra series to be useful anyway.

4. Given that the transfer functions higher than M_5 will not be evaluated, it may seem that writing out their formulae is pointless. The author disagrees:

   (a) To the author's knowledge, explicit Volterra transfer functions for terms such as $[\int y dt]^2[\int y^3 dt]$ have not been presented before in the literature.

   (b) Once the correct approach is found, calculating these terms is not difficult. It seems wrong not to find them simply because "it's too hard". They *can* be found explicitly, so why not do it?

4.3 Numerical Transient Analysis

The Volterra transfer functions can now be used to predict the behavior of the weakly nonlinear circuit when the input is composed of one or more sinusoids. We wish to investigate two things:

1. The improvement in accuracy that Volterra series afford us over simple linear analysis.

2. The magnitudes of the harmonics generated due to the nonlinearity, in particular, the distortion terms mentioned in §2.2.3.

§4.4 addresses the first problem while the rest of the thesis addresses the second. It will not do to simply use the Volterra transfer functions to calculate results; we would like to check our calculations with a numerical simulator.

Frequently in this thesis we will know explicit time-domain equations for circuits, e.g., equations (4.6) and (4.7). To solve them, we can implement our own numerical integration program in a high-level language. A program that implements the

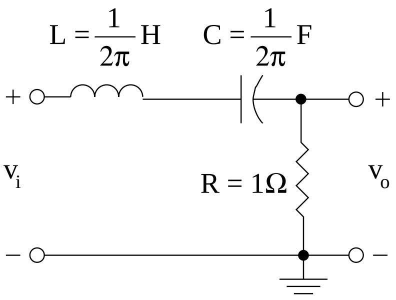


Figure 4.3: Simple linear band pass filter circuit.

fourth-order Runge-Kutta method [Pre92] for solving coupled differential equations was written in C. At other times we will want to simulate a circuit with complex devices such as transistors. SPICE is the industry standard for this task. We will be using its transient analysis feature for distortion simulations in Chapter 5.

But how accurate are Runge-Kutta and SPICE? Anyone who has used SPICE will know how dangerous it is to presuppose it provides accurate numerical results. The accuracy of SPICE is a topic upon which an entire thesis could be written, but let us confine ourselves to some simple observations about both SPICE and Runge-Kutta numerical simulation which will be relevant in this thesis. For the remainder of this section, we will be simulating the simple *linear* band pass circuit shown in Figure 4.3. Choosing a linear circuit is logical because we know exactly how it should behave, and we can use it to examine the accuracy of SPICE and Runge-Kutta transient analysis. It will transpire that our observations will apply equally well to nonlinear circuits.

The component values in the linear filter have been set so that the center frequency is $f_0 = 1$Hz. In the frequency domain, the transfer function for this filter is

$$\frac{V_o}{V_i} = \frac{\frac{R}{L}s}{s^2 + \frac{R}{L}s + \frac{1}{LC}} \tag{4.18}$$

Because the adaptive time step algorithm in the "H92b" version of SPICE was found to give clearly erroneous results under certain conditions, the "H9007" option was included in the SPICE input file. This is an older fixed time step algorithm that

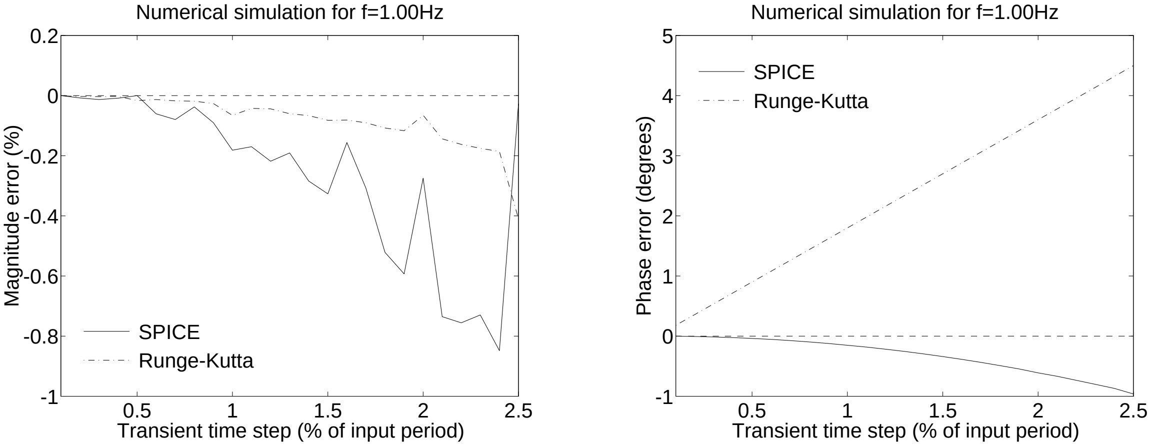


Figure 4.4: Results of SPICE and Runge-Kutta transient analysis for $f_{in} = 1$Hz.

gives much better results with simulation control variables left at their default values. Although [Pre92] recommends adaptive time stepping when using the Runge-Kutta algorithm to increase speed, the simulations we are doing are not long enough to warrant it.

4.3.1 The Importance of Time Step

Let us investigate what happens when the input is a 1V-sinusoid at the center frequency, 1Hz. From (4.18), the output should be a 1V-sinusoid in phase with the input.

Graphs of the magnitude and phase of the output from the numerical transient analysis are plotted in Figure 4.4. The x-axis shows the transient analysis time step normalized to $1/f_{in} = T_{in}$, the period of the input voltage, and the y-axis shows for the magnitude and phase graphs, respectively, the percent magnitude error and the number of degrees phase error.

It is apparent that as the time step gets smaller, the magnitude and phase from the transient simulation get closer to the expected values. This behavior is not surprising: intuitively, a numerical integration algorithm should perform better with a

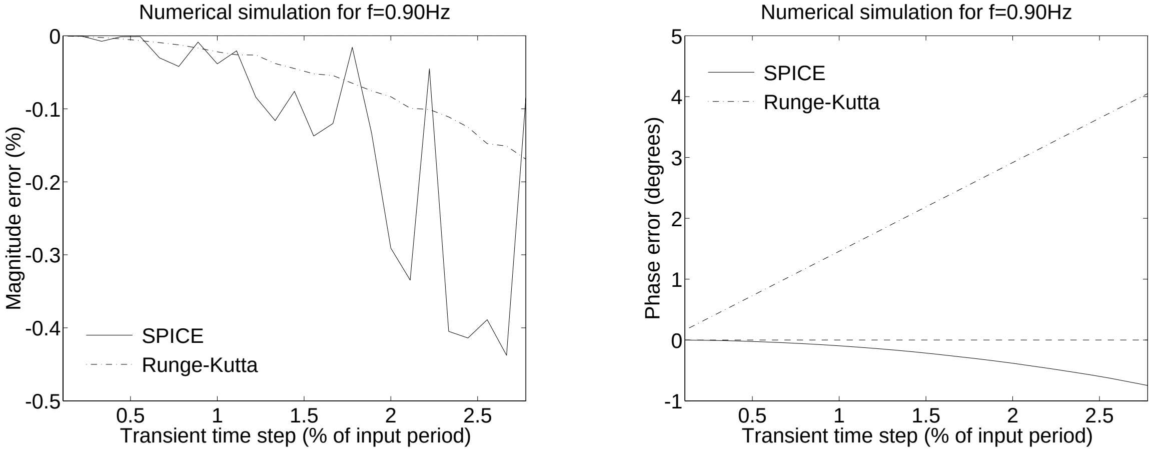


Figure 4.5: Results of SPICE and Runge-Kutta transient analysis for $f_{in} = 0.9$Hz.

smaller time step provided there are no discontinuities in the system equations and no roundoff, and this circuit is linear and the equations continuous. Of course, the trade-off is that a more accurate solution requires longer simulation time.[1]

Graphs similar to Figure 4.4 are plotted in Figure 4.5 and Figure 4.6 for $f_{in} = 0.9$Hz and 1.1Hz, respectively, to illustrate that the trend of increasing accuracy for smaller time steps holds at input frequencies other than band center. It is interesting that the Runge-Kutta phase error is a linear function of time step. It can be calculated that as a function of input frequency f and time step T the phase error is about $1800fT$ degrees.

4.3.2 The Importance of Long Simulations

Several places in this thesis will require a frequency spectrum measurement, which can be obtained from a fast Fourier transform (FFT) of a numerical time-domain transient analysis. This subsection and the next illustrate two important considerations for FFT calculations. The results quoted here are for SPICE but they hold equally true

[1] The large magnitude spike in SPICE at 2.5% (i.e., a time step of 25ms) probably occurs because 25ms is an even divisor of one second, the period of the input.

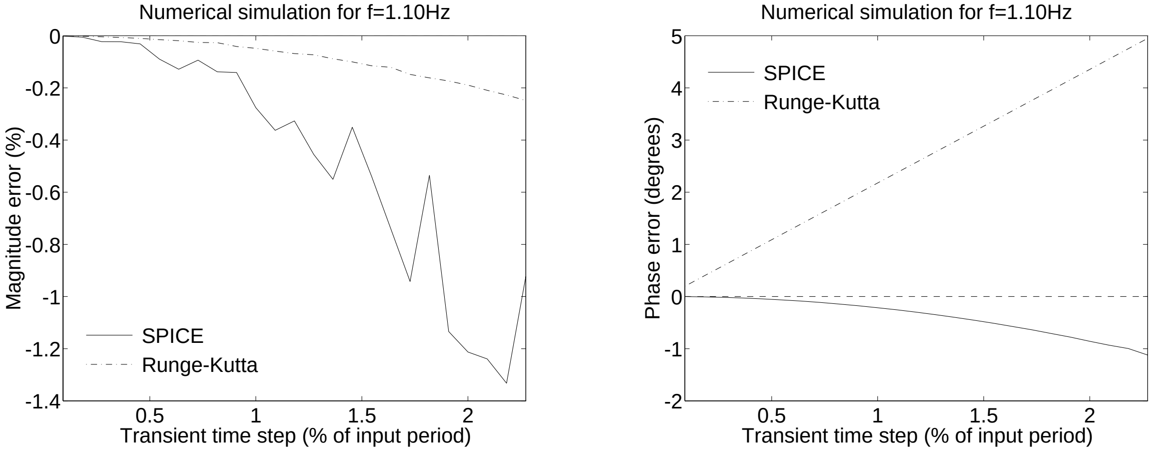


Figure 4.6: Results of SPICE and Runge-Kutta transient analysis for $f_{in} = 1.1\text{Hz}$.

for Runge-Kutta.

We know that in Figure 4.3 when the input is a single sinusoid, the output should be a single sinusoid at the same frequency after the transients have died out. The left graph in Figure 4.7 shows the first ten cycles of the output when the input is a 1Hz, 1V sine wave and the time step is 10ms. The right graph shows the 10,000-point FFT of 100 cycles of the same simulation. Line A (the top line) is the FFT of the *first* 100 cycles of the output waveform, line B (the middle line) is the FFT of the 100 cycles after the first *five* cycles, and line C (the bottom line) is the FFT of the 100 cycles after the first *ten* cycles.

Experience with FFTs tells us that a change in a signal's amplitude raises the "noise floor" of its FFT graph. This is evident in line A: we are including the very first cycle of the output which has a smaller amplitude than the rest of cycles. As a result the FFT noise floor around the 0dB-tone at 1Hz is between -50dB and -70dB. Even small changes in amplitude can have a deleterious effect; to the naked eye, it appears that the transients in the output have died out fully after five cycles — an observation refuted by the FFT graph, which shows that the noise floor of line C is about 50dB below that indicated by line B. This demonstrates that the phrase "the

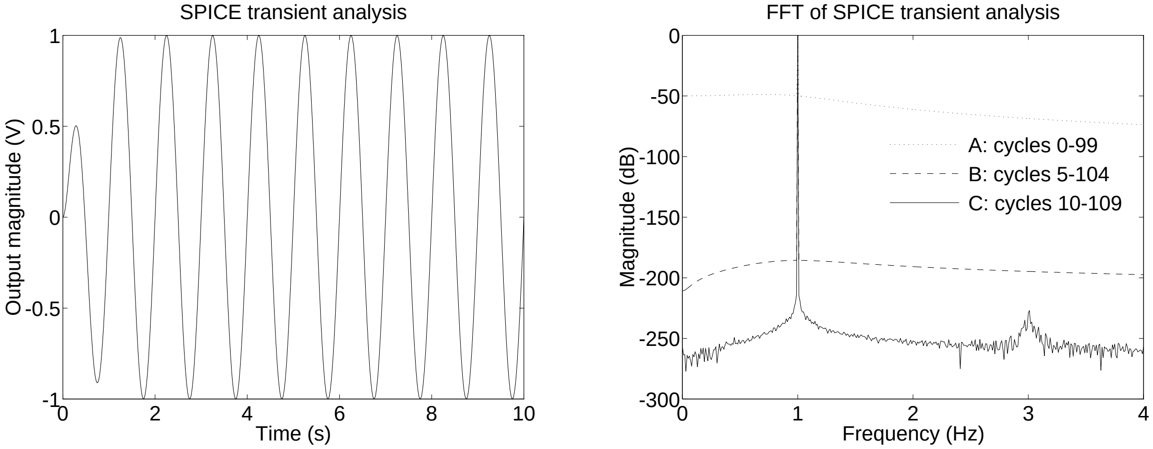


Figure 4.7: Output waveform for $f_{in} = 1\text{Hz}$.

transients have died out" is context-dependent. In the time domain, transients may appear to have died out; in the frequency domain, the FFT may indicate otherwise.

It might come as a surprise to some that SPICE can achieve a noise floor more than ten orders of magnitude below the desired tone, for this means that SPICE can be accurate to more than ten decimal places, albeit for a linear circuit. We will be hoping this accuracy holds for complex nonlinear circuits in the numerical Volterra series extraction coming up in Chapter 5: each order of magnitude lowering of the noise floor will be crucial for results that are not misleading. We shall see in §5.5.2 that the price to be paid in high-Q filter circuits is mammoth simulation times to ensure transients have fully settled.

4.3.3 The Importance of Decimal Places

The FFT noise floor is not tied to the purity of the output signal alone; it depends directly on the number of decimal places of accuracy.

Figure 4.8 shows the FFT of the output in three different simulations, all with a 1V-input sinusoid. Line A has $f_{in} = (1 + 10^{-6})\text{Hz}$, line B has $f_{in} = (1 + 10^{-9})\text{Hz}$, and line C has $f_{in} = 1\text{Hz}$ exactly. These graphs illustrate that if the input frequency is

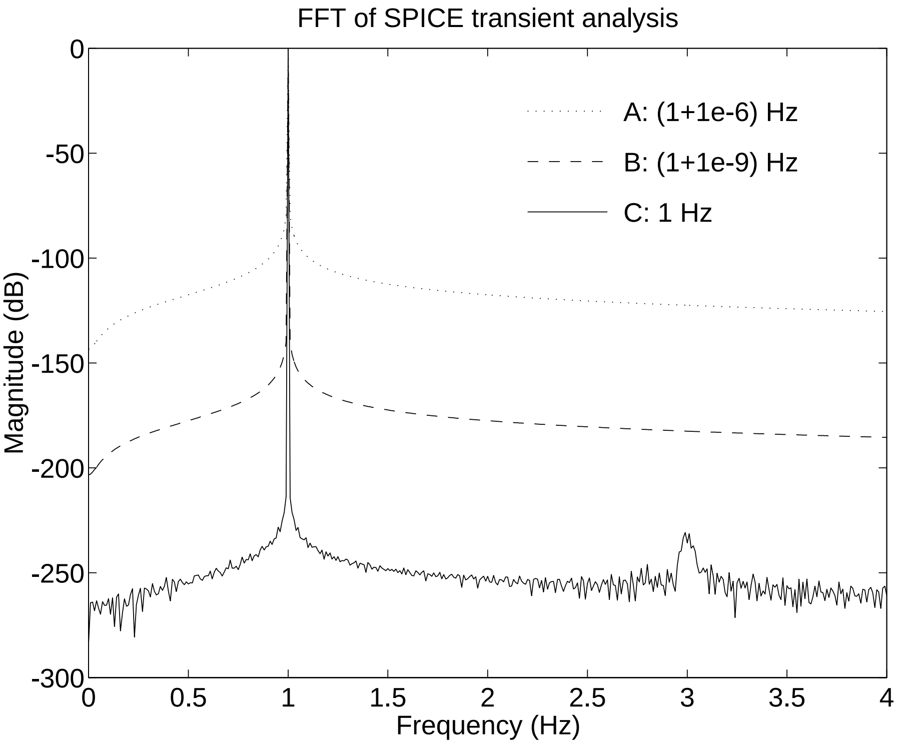


Figure 4.8: Noise floor with slight input frequency errors.

not an exact multiple of the time step, even by as small an amount as one part per billion, the FFT noise floor can be degraded by several orders of magnitude. This is similar to what was observed in §4.3.2: transients that have not *fully* died out are akin to small errors in the last few decimal places of the output.

4.3.4 What We Have Learned

The preceding three sections have offered the following two insights into the numerical simulation of circuits:

- We must choose a small enough time step that our results are acceptably accurate. We will probably have to live with small inaccuracies in magnitudes and phases if we want simulation times that are reasonable.

- We must ensure that all transients really have died out, and we must specify components and frequencies to as many decimal places as possible. These will minimize the FFT noise floor.

4.4 Single Tone Tests

We are now ready to compare the transient analysis to the Volterra transfer functions for the nonlinear filter. To begin, let us assign numerical values to its components. We will make $g_{mi} = g_{m1} = g_{m2} = 2\pi$ and $C_1 = C_2 = 1\text{F}$. This will make $\omega_0 = \sqrt{\frac{g_{m1}g_{m2}}{C_1 C_2}} = 2\pi\frac{\text{rad}}{\text{s}}$, or $f_0 = 1\text{Hz}$, and $Q = \omega_0 R_1 = 2\pi R_1$. The linear gain at the center frequency f_0 will be $A_0 = g_{mi}R_1 = 2\pi R_1$. We shall see in §4.6 that choosing such a center frequency is a sort of *normalization*.

Let us set $R_1 = 5\Omega$; this will give a filter Q of about 30. Let us also make $\epsilon_2 = 0.05$ and $\epsilon_i = \epsilon_1 = 0$. This is making only one TA nonlinear and is perhaps not very "realistic" in that each TA would likely have a similar nonlinearity in a fabricated circuit. For now, our choice will put *some* nonlinearity in the circuit and will give us a starting point. A circuit diagram with the components and their values is shown in Figure 4.9.

4.4.1 Small Amplitude and Frequency at Band Center

For a start, let us calculate the amplitude and phase of the output signal for an input at the center frequency, $f_a = 1\text{Hz}$, with a small amplitude — say $V_a = 1\text{mV}$. We shall use equation (3.22) from §3.3, which says that for an input of $V_a \cos 2\pi f_a t$ the output component at f_a will be

$$\begin{aligned} &e^{j\omega_a t}\left[\frac{V_a}{2}M_1(f_a) + \frac{V_a^3}{16}M_3(f_a,f_a,-f_a) + \frac{V_a^5}{384}M_5(f_a,f_a,f_a,-f_a,-f_a) + \cdots\right] \\ &+e^{-j\omega_a t}\left[\frac{V_a}{2}M_1(-f_a) + \frac{V_a^3}{16}M_3(-f_a,-f_a,f_a) + \frac{V_a^5}{384}M_5(-f_a,-f_a,-f_a,f_a,f_a) + \cdots\right] \end{aligned} \tag{4.19}$$

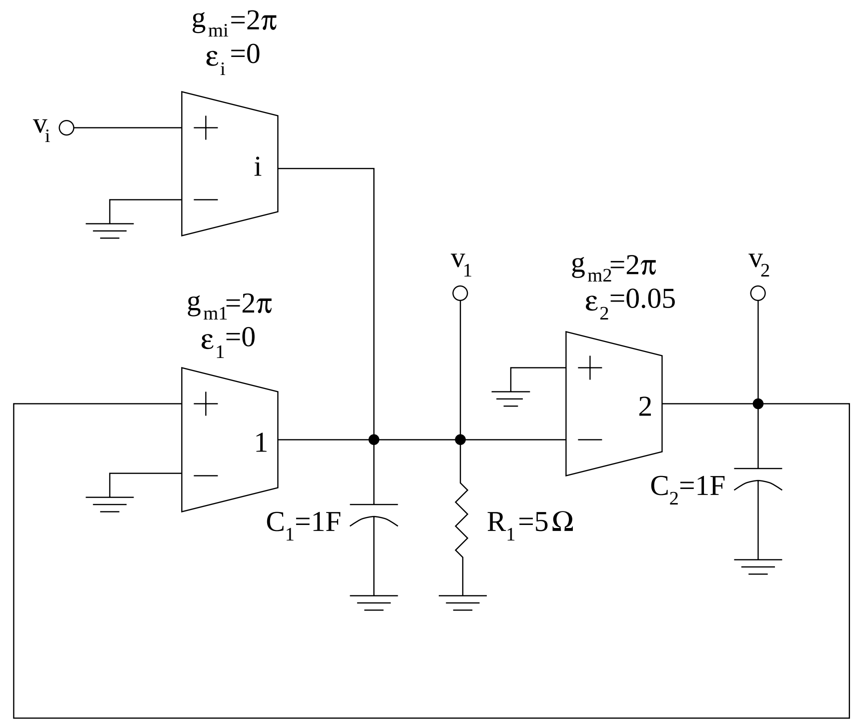


Figure 4.9: Filter circuit with values.

where $\omega_a = 2\pi f_a$. From equation (3.8) it follows that if the coefficient of $e^{j\omega_a t}$ is $a+jb$, then the coefficient of $e^{-j\omega_a t}$ will be $a-jb$, its complex conjugate. This means that (4.19) can be rewritten and simplified as follows:

$$\begin{aligned}
& (a+jb)e^{j\omega_a t} + (a-jb)e^{-j\omega_a t} \\
= \; & 2a\cos\omega_a t - 2b\sin\omega_a t \\
= \; & 2\sqrt{a^2+b^2}\cos(\omega_a t + \arctan\frac{b}{a}) \qquad (4.20)
\end{aligned}$$

Using $V_a = 10^{-3}$V and $f_a = 1$Hz in (4.19), the individual terms can be calculated as

$$\begin{aligned}
\tfrac{V_a}{2}M_1(f_a) &= 1.5708\times10^{-2} &&+ \; j0 \\
\tfrac{V_a^3}{16}M_3(f_a,f_a,-f_a) &= 0 &&+ \; j2.9068\times10^{-6} \\
\tfrac{V_a^5}{384}M_5(f_a,f_a,f_a,-f_a,-f_a) &= -5.3793\times10^{-10} &&+ \; j7.1334\times10^{-13}
\end{aligned}$$

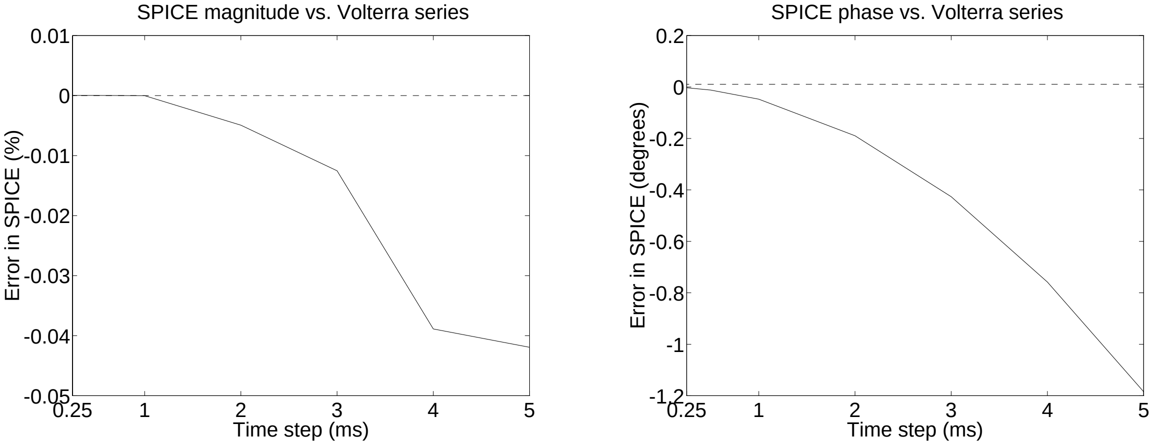


Figure 4.10: Difference between SPICE and Volterra series.

Adding the first, then the first two, then all three terms and applying (4.20) predicts the output signal amplitude and phase as

$$\begin{array}{rll} M_1 : & 31.4159265\angle 0^o & \text{mV} \\ M_1 + M_3 : & 31.4159271\angle 0.0106^o & \text{mV} \\ M_1 + M_3 + M_5 : & 31.4159260\angle 0.0106^o & \text{mV} \end{array} \qquad (4.21)$$

The magnitudes in equation (4.21) agree to seven digits. M_1 (which is just the linear transfer function) shows no phase shift at the center frequency, of course, but it can be seen that the nonlinearity produces a tiny phase shift. At least, that is what the Volterra transfer functions predict — does a numerical simulation of the circuit show the same behavior?

Using SPICE, the caveats from §4.3 now come into play. By trial and error, it is found that 300 seconds is long enough for the transients to fully die. Figure 4.10 shows the difference between the SPICE transient analysis and $M_1 + M_3$ as a function of time step. As expected, a smaller time step results in a more accurate numerical simulation; a time step of 0.25ms gives a magnitude error of $2.3 \times 10^{-5}\%$ and a phase error of -0.0029^o. That is, SPICE predicts a slight positive phase shift of 0.0077^o, compared with 0.0106^o for $M_1 + M_3$. It can be inferred from the graphs that if we

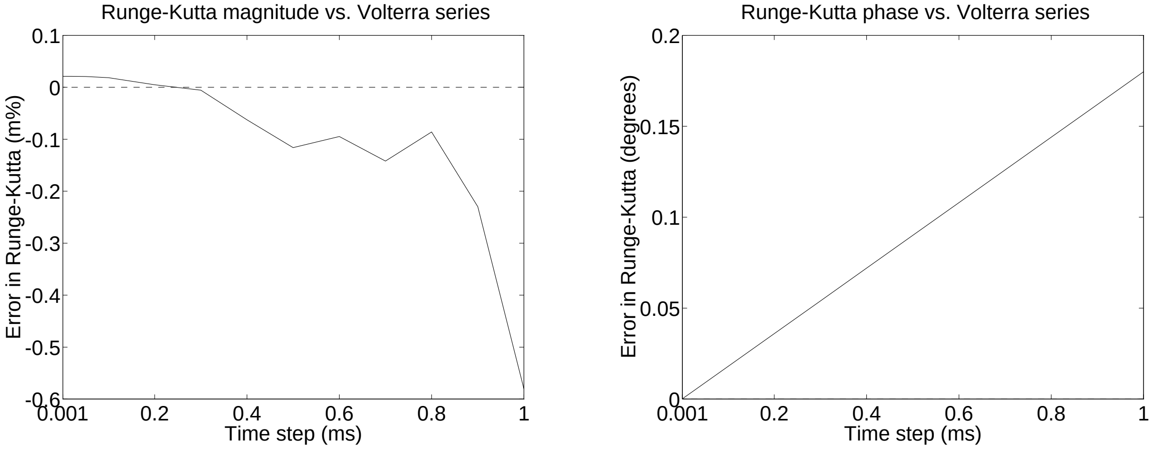


Figure 4.11: Difference between Runge-Kutta and Volterra Series.

could make the time step in the SPICE simulation infinitely small, the results would agree very closely to the Volterra transfer functions' prediction.

It would be nice to actually make the time step even smaller and increase our confidence, but this becomes unwieldy in SPICE for two reasons:

1. Simulation time increases rapidly. At a time step of 0.25ms, a simulation takes over twenty minutes on a fast computer. The simulation is $300 \times 4000 = 1,200,000$ time steps total, which in twenty minutes means 1000 steps per minute. This is not very fast, presumably because SPICE, being such a complex program, has a good deal of overhead which becomes particularly noticeable in simple circuits.

2. It uses more memory for finer time divisions. This memory is not really needed for our purposes: all we wish to do is measure the output peak and phase shift, yet in SPICE we have no choice but to save the values at every time step.

We can overcome both difficulties with the Runge-Kutta program: it is very fast when compiled and it can be made to extract only the information we want — the amplitude of the output signal and the phase difference between it and the input.

The difference between the calculated $M_1 + M_3$ output and the simulated one as a function of time step is shown in Figure 4.11. Let us compare the results of a 300-second SPICE simulation to those of a 300-second Runge-Kutta simulation.

For a 1ms time step, SPICE gives a magnitude error of $-1.6 \times 10^{-5}\%$ and a phase error of -0.047^o, whereas the Runge-Kutta algorithm gives errors of $-5.8 \times 10^{-4}\%$ and 0.18^o, respectively. SPICE, however, takes over five minutes to execute, while Runge-Kutta takes under ten seconds. If we reduce the time step in Runge-Kutta to 0.1ms, the simulation takes about eighty seconds, and the errors are now $-1.8 \times 10^{-5}\%$ and 0.018^o, respectively — better than a SPICE simulation that takes almost four times as long. Not only that, but the clumsy post-processing that must be done in SPICE to extract the magnitude and phase information has been avoided in the Runge-Kutta program: the program extracts only the necessary information.

Particularly interesting is the fact that the Runge-Kutta phase error varies almost linearly with time step (the graph on the right of Figure 4.11) for this nonlinear circuit just as it did for the linear circuit in §4.3. The trend continues to very small time steps, even 10μs and 1μs. The phase error at this latter time step is a mere 0.00018^o. If we perform a linear regression of the last few points on the phase graph, we find that the phase at a time step of zero would be 0.0106027^o, compared with 0.0106029^o from $M_1 + M_3$. For small time steps, then, the magnitude of the Volterra series calculation agrees with the simulation to about seven digits, and the phase to about six digits. Therefore, it seems reasonable to conclude that the Volterra series approach models the real circuit extremely closely.

The case we have examined is for a small input at the center frequency. How well do Volterra series approximate reality at other amplitudes and frequencies?

4.4.2 Various Amplitudes and Frequencies

For the moment, let us keep the frequency fixed at band center, 1Hz, and examine what happens as the amplitude is increased. We shall continue to use the Runge-

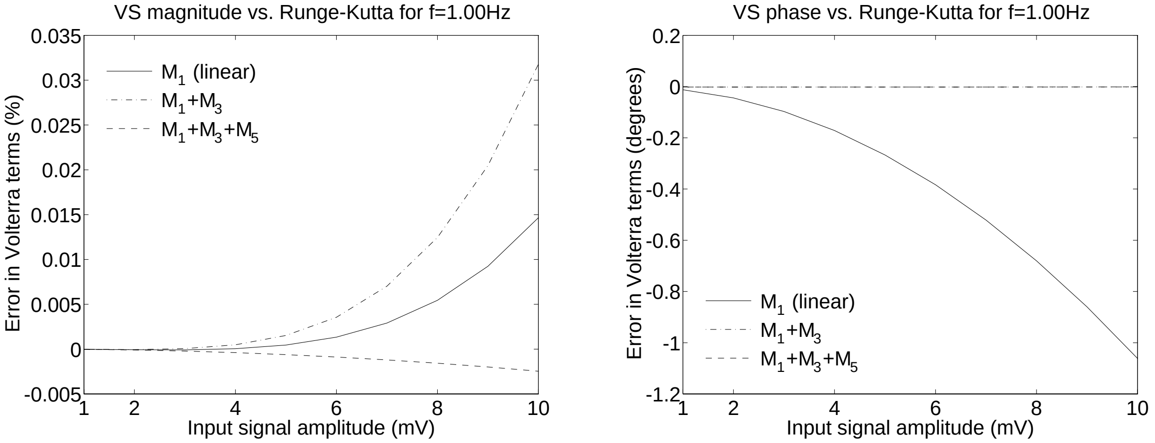


Figure 4.12: Volterra series accuracy as a function of input amplitude.

Kutta program.

Figure 4.12 shows the difference between the predicted and simulated output magnitude and phase with the input amplitude varying from 1mV to 10mV. The error in the magnitude calculation is small in all three cases, but interestingly, the linear transfer function M_1 alone is *more* accurate than $M_1 + M_3$. Including M_5 seems to correct this deficiency somewhat. However, the Volterra transfer functions do a much better job of predicting the *phase* than the linear transfer function alone: with an input of 10mV, the linear transfer function is off by a whole degree while the nonlinear transfer functions correct the error to less than 0.002^o.

The results of sweeping the input signal amplitude from 10mV to 100mV and comparing the calculation to the simulation are as shown in Figure 4.13. The nonlinear transfer functions provide some amplitude and phase correction over the linear transfer function for input amplitudes up to about 20mV, but after that the simulation predicts quite different values from the transfer functions. Moreover, $M_1 + M_3$ and $M_1 + M_3 + M_5$ disagree wildly with each other for an input of 100mV. It would appear then that for inputs of this size, the sum is diverging (recall the discussion in §3.4.2), which means Volterra series are no longer giving meaningful predictions about the system. We will attempt to model the system with a large input of 200mV

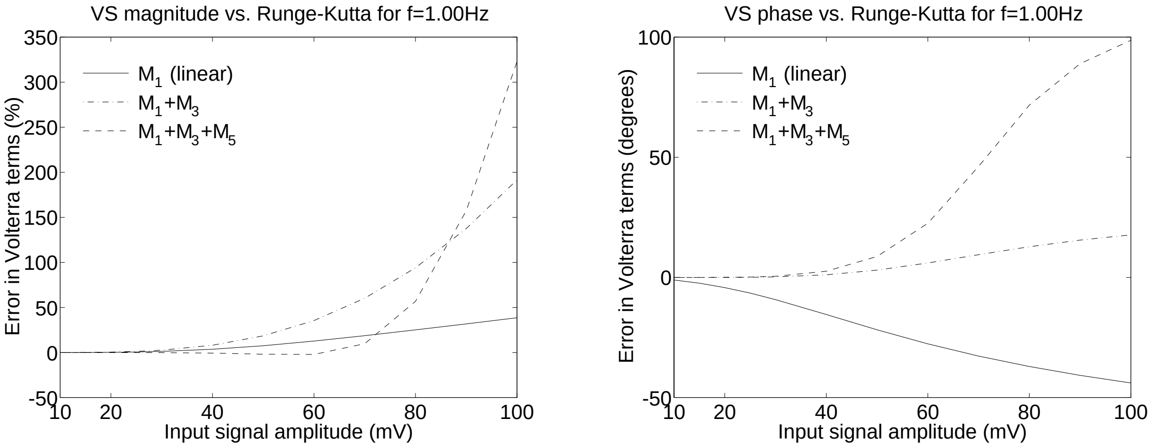


Figure 4.13: Volterra series versus input amplitude for larger inputs.

more accurately in the next section.

For completeness, Figure 4.14 shows the error in the Volterra transfer function calculations at three other frequencies: 0.90Hz, 0.95Hz, and 1.05Hz. It is evident that $M_1 + M_3$ provides a significantly more accurate prediction of the output amplitude and phase than does M_1 alone. As well, the effect of the nonlinearity is weaker at frequencies farther away from band center in that the error at a particular input amplitude gets smaller. For example, at 100mV, the error in the linear transfer function is 5% at $f = 0.95$Hz but only 0.8% at $f = 0.90$Hz. (At these frequencies, the Volterra series appears to be converging: the correction M_3 adds to M_1 is large, but the correction M_5 adds is quite small.)

Let us briefly examine some simulations with a fixed amplitude. In Figure 4.15, the input amplitude has been set to 5mV and the error in the Volterra terms is plotted as a function of frequency. Clearly, the linear transfer function error *is* smaller the further away from the center frequency we go as was implied in Figure 4.14. As well, both $M_1 + M_3$ and $M_1 + M_3 + M_5$ agree closely with the simulation — more so than does M_1 alone.

To get an overall picture of the Volterra transfer functions' accuracy, it is instructive to draw a three-dimensional graph with amplitude and frequency being the

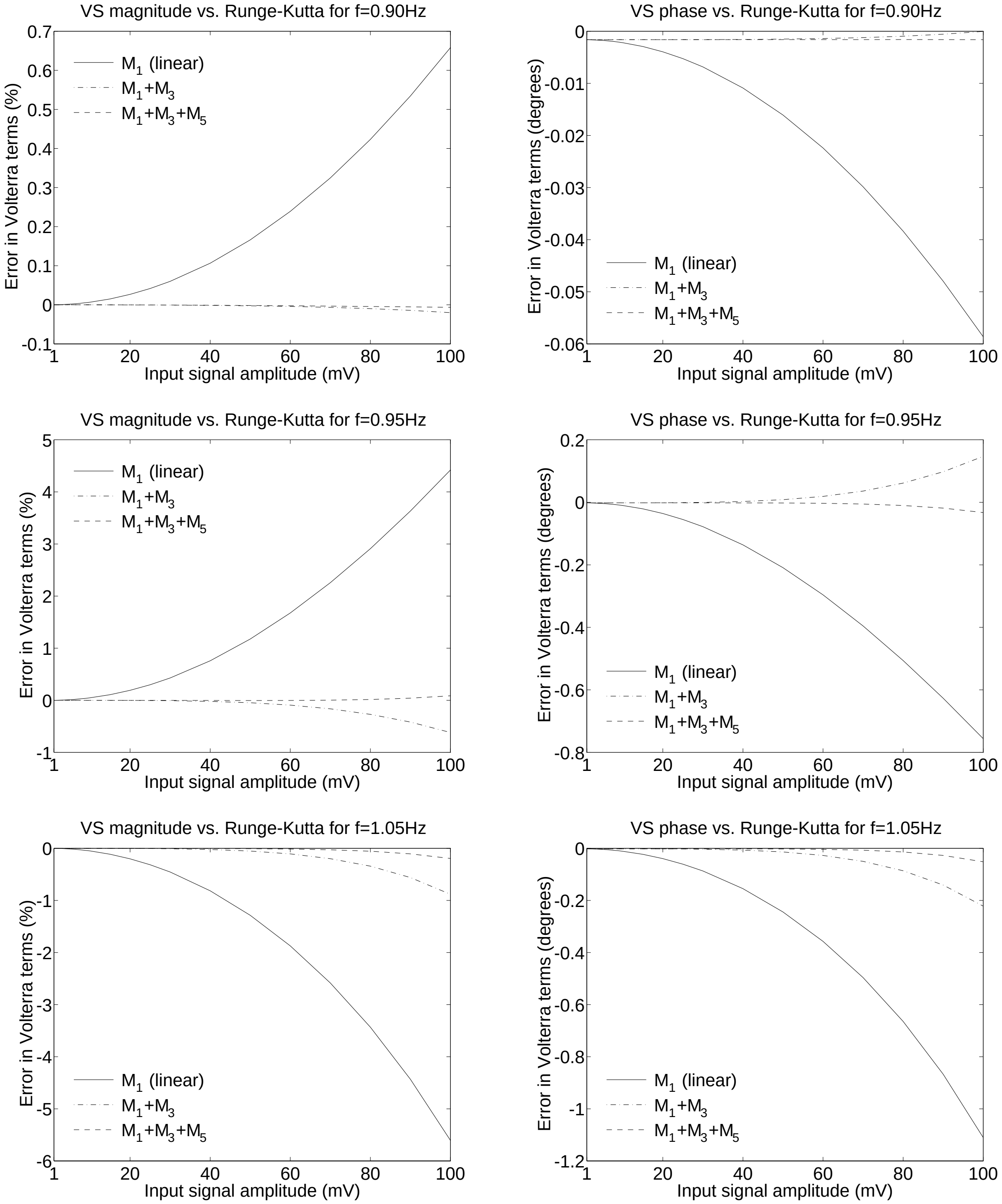


Figure 4.14: Volterra series versus input amplitude at various frequencies.

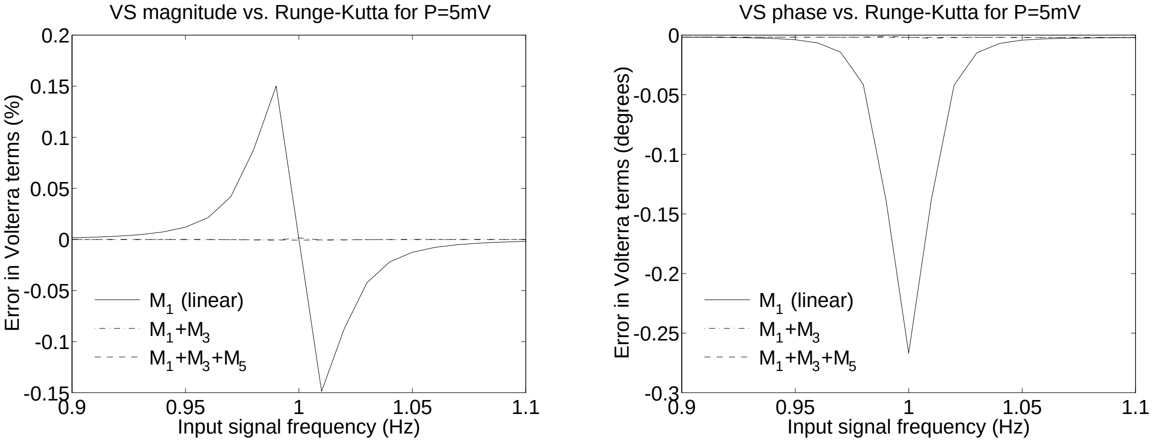


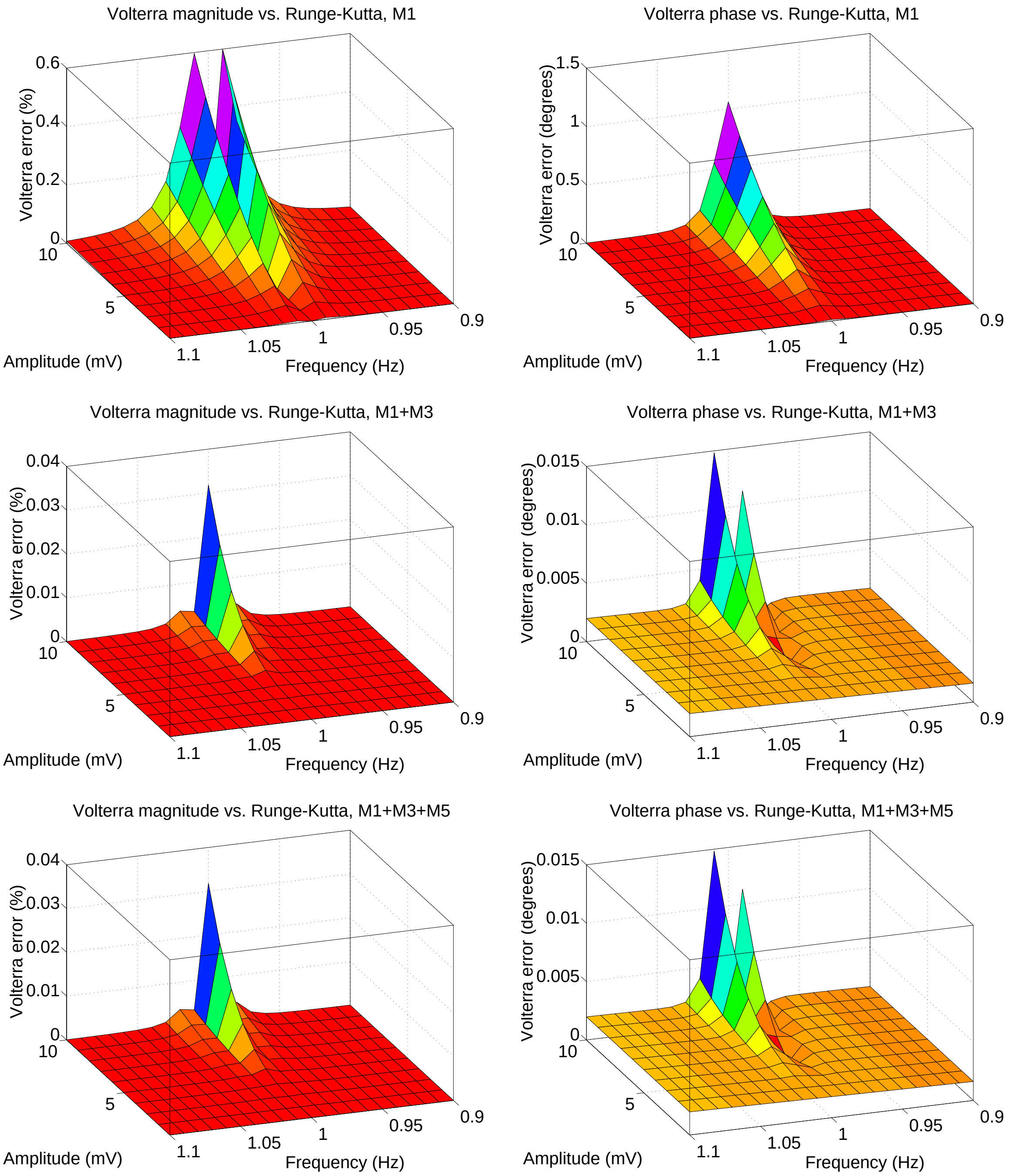
Figure 4.15: Volterra series versus frequency for a 5mV input.

independent variables. Figure 4.16 illustrates such plots for input amplitudes between 1mV and 10mV, and Figure 4.17 is for 10mV to 100mV. In the first set of graphs, $M_1 + M_3$ provides a significant correction over M_1 around the center frequency where M_1 is most inaccurate. $M_1 + M_3 + M_5$ is almost identical with $M_1 + M_3$. For the second set of graphs (the ones with a larger input amplitude range), the nonlinear transfer functions do not provide accurate correction at band center for the nonlinearity; all six surfaces exhibit large errors.

Because of the simplicity of the assumed nonlinearity and because of our knowledge of the system equations, we *can* explain why the Volterra transfer functions do not do a good job of modeling the simulated circuit behavior for large inputs.

4.4.3 Strong Nonlinearity in the Filter

Figure 4.18 shows the simulated magnitude and phase at the output as a function of both input amplitude and frequency. For small inputs, the band pass magnitude and phase characteristics both look smooth, but for larger inputs, discontinuities are apparent on both surfaces. We can show that these are due to the nonlinearity becoming "too strong" as the input becomes larger.


Volterra magnitude vs. Runge-Kutta, M1

Volterra phase vs. Runge-Kutta, M1

Volterra magnitude vs. Runge-Kutta, M1+M3

Volterra phase vs. Runge-Kutta, M1+M3

Volterra magnitude vs. Runge-Kutta, M1+M3+M5

Volterra phase vs. Runge-Kutta, M1+M3+M5

Figure 4.16: Volterra accuracy surface plots for small inputs.

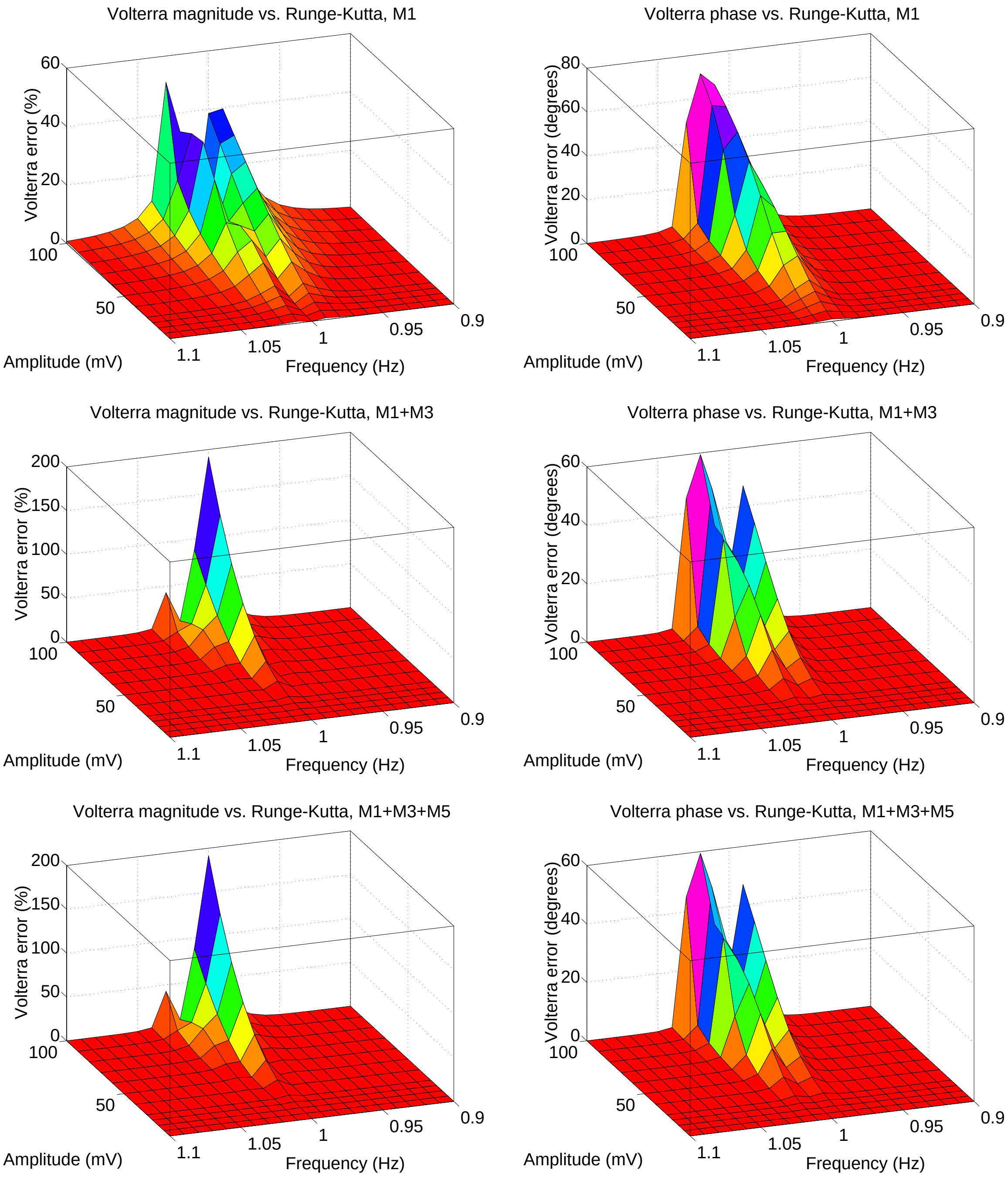
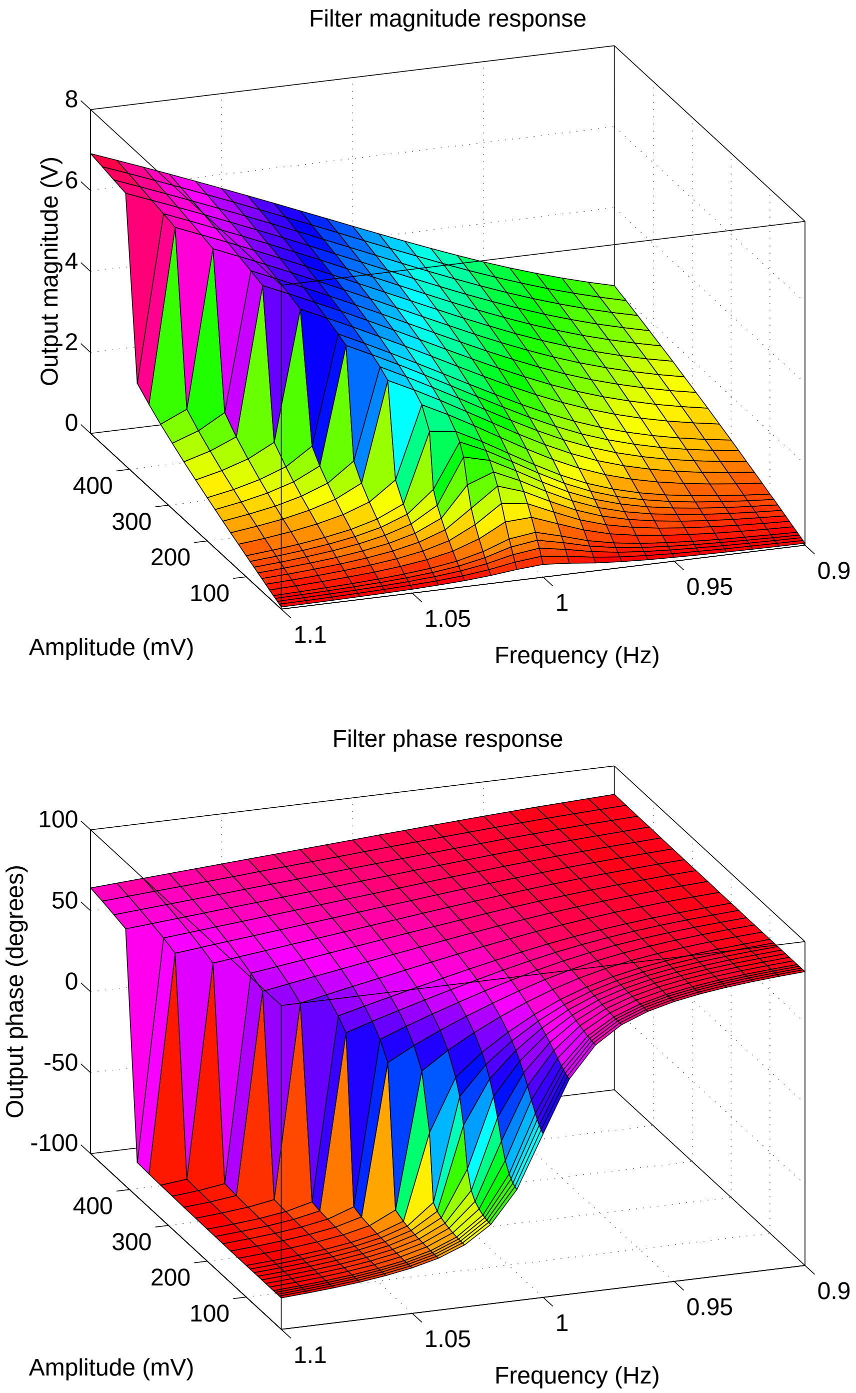


Figure 4.17: Volterra accuracy surface plots for large inputs.


Filter magnitude response

Filter phase response

Figure 4.18: Filter magnitude and phase response surfaces.

Algebraic analysis

Let us return to the time domain representation of the filter, equations (4.6) and (4.7), repeated here for convenience:

$$\begin{aligned} C_1\frac{dv_1}{dt} &= -\frac{v_1}{R_1} + g_{mi}v_i + \epsilon_i v_i^3 + g_{m1}v_2 + \epsilon_1 v_2^3 \\ C_2\frac{dv_2}{dt} &= -g_{m2}v_1 - \epsilon_2 v_1^3 \end{aligned}$$

We have been investigating the case where $\epsilon_i = \epsilon_1 = 0$. Substituting these gives

$$C_1\frac{dv_1}{dt} = -\frac{v_1}{R_1} + g_{mi}v_i + g_{m1}v_2 \tag{4.22}$$

$$C_2\frac{dv_2}{dt} = -g_{m2}v_1 - \epsilon_2 v_1^3 \tag{4.23}$$

Solving equation (4.22) for v_2 then differentiating it with respect to t gives

$$v_2 = \frac{C_1}{g_{m1}}\frac{dv_1}{dt} + \frac{1}{R_1 g_{m1}}v_1 - \frac{g_{mi}}{g_{m1}}v_i \tag{4.24}$$

$$\frac{dv_2}{dt} = \frac{C_1}{g_{m1}}\frac{d^2v_1}{dt^2} + \frac{1}{R_1 g_{m1}}\frac{dv_1}{dt} - \frac{g_{mi}}{g_{m1}}\frac{dv_i}{dt} \tag{4.25}$$

Substituting (4.25) into (4.23) and putting things in a nice form gives

$$\ddot{v}_1 + \frac{g_{m1}g_{m2}}{C_1C_2}v_1 = -\frac{1}{R_1C_1}\dot{v}_1 - \frac{\epsilon_2 g_{m1}}{C_1C_2}v_1^3 + \frac{g_{mi}}{C_1}\dot{v}_i \tag{4.26}$$

In the literature we find that equation (4.26) can be identified with the *forced Duffing equation* [Nayf79]

$$\ddot{u} + \omega_0^2 u = -2\epsilon\mu\dot{u} - \epsilon\alpha u^3 + E(t) \tag{4.27}$$

This is simply a non-homogeneous second-order differential equation in u with the addition of a u^3 term, denoted a *nonlinear restoring force.* In (4.27), ω_0 is the resonant frequency, μ is the damping coefficient, ϵ is a small parameter[2], α is the strength of

[2] The author apologizes for the notational conflict between ϵ_2, the TA nonlinearity, and ϵ, the small Duffing equation parameter. The conflict only exists in this section.

the nonlinearity, and $E(t)$ is the input. We will assume that the input is sinusoidal and close to the resonant frequency:

$$\begin{aligned} E(t) &= \epsilon k \cos(\Omega t) \\ &= \epsilon k \cos((\omega_0 + \epsilon\sigma)t) \end{aligned} \tag{4.28}$$

where ϵ is the small parameter and σ is the *detuning.* Comparing our filter equation (4.26) with the Duffing equation (4.27) and the input (4.28), we can identify the variables as follows:

$$\begin{aligned} u &= v_1 \\ \omega_0^2 &= \frac{g_{m1}g_{m2}}{C_1C_2} \text{ as expected} \\ \mu &= \frac{1}{2\epsilon R_1C_1} \\ \alpha &= \epsilon_2\frac{g_{m1}}{\epsilon C_1C_2} \\ k &= \frac{g_{mi}\Omega}{\epsilon C_1}V_{in} \text{ when } v_{in} = V_{in}\cos(\Omega t) \end{aligned} \tag{4.29}$$

Equation (4.27) does not have an exact closed-form solution for u, but we can solve it approximately for frequencies close to ω_0. Solving it and then making the substitutions listed in (4.29) will yield the solution to our filter equation (4.26). To start with, u is assumed to be of the form

$$\begin{aligned} u &= a\cos(\Omega t - \gamma) + O(\epsilon) \\ &\approx a\cos(\Omega t - \gamma) \\ &= a\cos((\omega_0 + \epsilon\sigma)t - \gamma) \end{aligned} \tag{4.30}$$

That is, the output is assumed to be predominantly a sinusoid at the same frequency as the input with amplitude a and phase γ with respect to the input. The $O(\epsilon)$ means that we are assuming any other components in u (for example, the component at 3Ω which will arise from the u^3 term) are much smaller than the main component at frequency Ω.

To solve (4.27), we will proceed as Nayfeh and Mook do in [Nayf79]. The derivation is not identical because we are differentiating the input — we have $\dot{v}_i(t)$ instead of v_i — but it is similar, and the reader is referred to [Nayf79] for more detail. Using the method of multiple scales [Nayf79], we can derive the pair of equations:

$$\begin{aligned} -\mu a &= \tfrac{1}{2}\frac{k}{\omega_0}\cos\gamma && (4.31) \\ a\sigma - \tfrac{3}{8}\frac{\alpha}{\omega_0}a^3 &= -\tfrac{1}{2}\frac{k}{\omega_0}\sin\gamma && (4.32) \end{aligned}$$

To solve (4.31) and (4.32) for the amplitude a we square and add them, yielding

$$\left[\mu^2 + \left(\sigma - \tfrac{3}{8}\frac{\alpha}{\omega_0}a^2\right)^2\right]a^2 = \frac{k^2}{4\omega_0^2} \qquad (4.33)$$

which is an implicit equation for the output amplitude a as a function of the detuning σ (i.e., the frequency of the input) and the input amplitude k. To solve for the phase γ we divide (4.31) by (4.32), yielding

$$\gamma = \arctan\frac{\sigma a - \frac{3}{8}\frac{\alpha}{\omega_0}a^3}{\mu a} \qquad (4.34)$$

How do we use (4.33) and (4.34) to draw magnitude and phase graphs for v_1? For the magnitude, we could solve (4.33) for a (amplitude) in terms of σ (frequency), but this involves solving a cubic in a^2. A simpler approach is to solve it for σ in terms of a. This yields

$$\sigma = \tfrac{3}{8}\frac{\alpha}{\omega_0}a^2 \pm \sqrt{\frac{k^2}{4\omega_0^2 a^2} - \mu^2} \qquad (4.35)$$

At each particular a value a_p, there will an upper and a lower solution to (4.35), σ_u and σ_l. The two (x, y) points on the v_1 magnitude response graph will be

$$(\omega_0 + \epsilon\sigma_l, a_p) \text{ and } (\omega_0 + \epsilon\sigma_u, a_p) \qquad (4.36)$$

The range of a_p values over which we evaluate (4.35) is from $a_{min} = 0$ up to the a that makes the quantity under the square root sign zero:

$$\begin{aligned} \frac{k^2}{4\omega_0^2 a^2} - \mu^2 &= 0 \\ a_{max} &= \frac{k}{2\omega_0\mu} && (4.37) \end{aligned}$$

For the phase graph, all that is required is substitution of (σ_l, a_p) and (σ_u, a_p) into (4.34) to obtain two points (σ_u, γ_u) and (σ_l, γ_l). The corresponding two points on the v_1 phase response graph will be

$$(\omega_0 + \epsilon\sigma_l, \gamma_l) \text{ and } (\omega_0 + \epsilon\sigma_u, \gamma_u) \tag{4.38}$$

Aside: the Duffing Equation Parameters

For those who have never seen the Duffing equation before, it is most informative to show the effect on the response of varying the parameters α, μ, and k. This section is somewhat tutorial in nature and is perhaps inappropriate in a thesis, but the author feels it is short enough and of significant enough interest that it is worthy of inclusion.

Figure 4.19 shows the magnitude and phase response of the filter with a 200mV input and with three different values of ϵ_2: 0, 0.05, and -0.05. The solid line in both graphs corresponds to $\epsilon_2 = 0$; since this corresponds to a filter with no nonlinearity, it should behave exactly as a linear band pass filter and the graphs confirm this. Choosing a positive value for ϵ_2 makes the magnitude graph bend to the right and distorts the central portion of the phase graph to the right; choosing a negative value for ϵ_2 distorts both graphs to the left instead. The larger $|\epsilon_2|$, the greater the bending-over effect. For $\epsilon_2 \neq 0$, it can be seen that at some frequencies there are *three* possible output magnitudes and phases. It can be shown that the middle one is unstable but that the outer two are stable [Nayf79]; which of the steady states the filter chooses depends upon the initial conditions. This will be discussed in greater detail shortly.

Figure 4.20 shows what happens when we fix $\epsilon_2 = 0.05$ and vary the damping coefficient μ. The filter Q is inversely proportional to damping, and we can see that the lower the damping, i.e., the higher the Q, the greater the distortion in the output magnitude and phase. If Q is low enough the output has only one stable solution at every frequency, as can be seen for the $Q = 2\pi R_1 = 6.28$ case.

Figure 4.21 shows the effect of varying the input amplitude. For small inputs, the output has only one stable operating point, but as the input gets larger, the

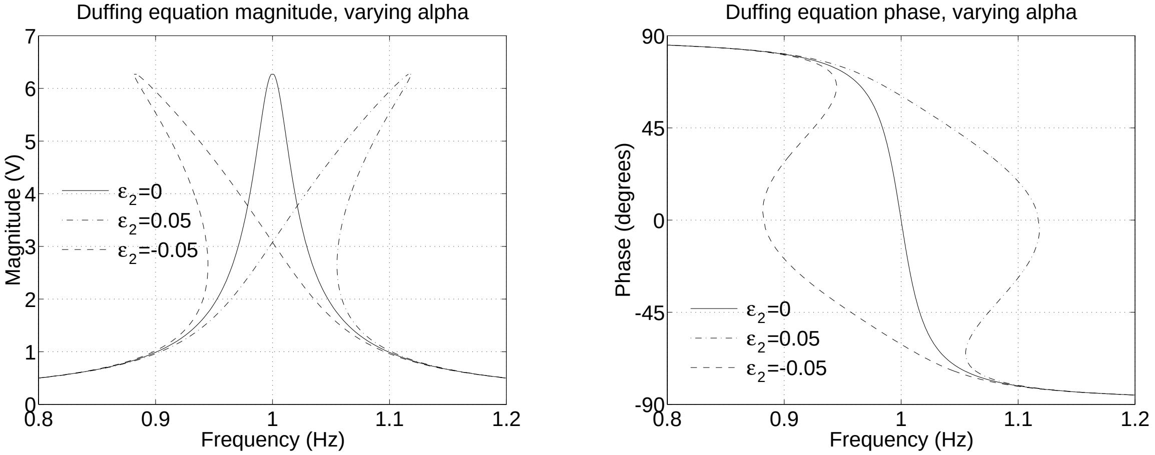


Figure 4.19: Effect of varying sign of nonlinearity in Duffing equation.

distortion of the characteristics becomes severe enough to give rise to two possible operating points.

To close this aside, we will comment briefly on a type of hysteresis which seems confounding to explain when observed in the laboratory. In Figure 4.22, the magnitude curve for the filter with $V_{in} = 200$mV has been plotted. If we were to connect this circuit in the laboratory and increase the frequency slowly starting from point A, then on a spectrum analyzer the gain would follow the dash-dot line. It would increase smoothly through point B until point C where it would suddenly drop to point E and continue to point F. Immediately after the drop from C to E, decreasing the frequency even slightly would not reverse the jump. On the other hand, if we were to decrease frequency slowly starting from point F, the gain would follow the dashed line: it would decrease smoothly through point E until point D where it would suddenly jump up to point B, and even increasing the frequency slightly would not reverse the jump.

If we understand the Duffing equation the explanation for the hysteresis should be clear. No circuit is perfectly linear in real life, no matter how well-designed it is, and under certain conditions the small nonlinearities might become large enough to

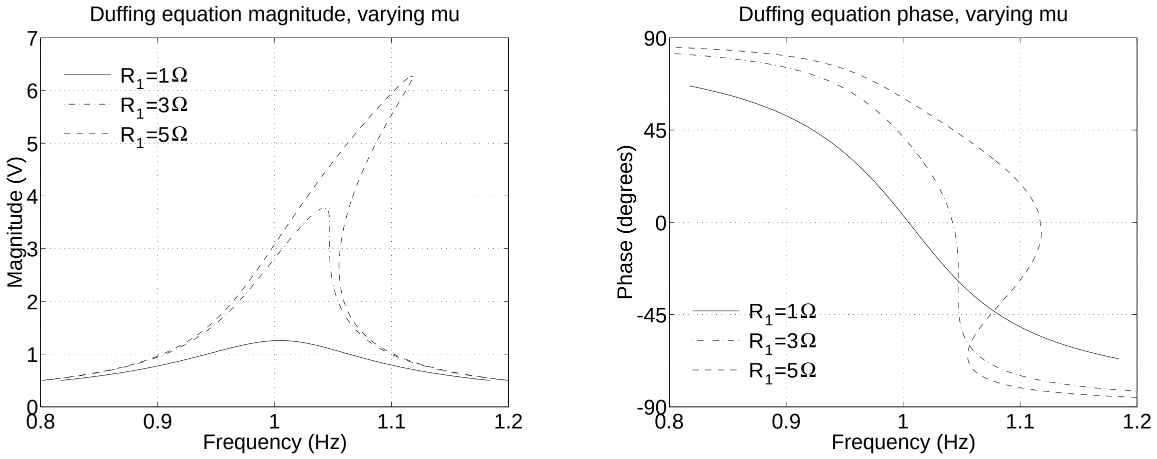


Figure 4.20: Effect of varying damping in Duffing equation.

have a strong effect. We must be wary.

Duffing versus Runge-Kutta and Volterra

We can now see why there are discontinuities in Figure 4.18: for large enough inputs, the responses become multi-valued. In the Figure the initial conditions were such that the output chose the *lower* of the two stable solutions when two solutions existed. This means that as the frequency was increased, the output magnitude rose slowly and the output phase fell slowly until the multiple-solution region of the response when both suddenly jumped down and followed the lower branches of the curves instead of continuing along the upper branches.[3]

To verify the accuracy of the calculated Duffing equation response and demonstrate the existence of the two stable solutions, the filter was simulated with a 200mV input over a range of frequencies using the Runge-Kutta program. It was found by trial and error that using the initial conditions $v_1 = 0\text{V}$, $v_2 = 0\text{V}$ yields solutions on the lower portion of the response curves while the initial conditions $v_1 = -6\text{V}$,

[3] Of course, even if the solution had followed the upper branches, there would *still* be a discontinuity as it "fell off" the peaks of the response curves down to the lower branches.

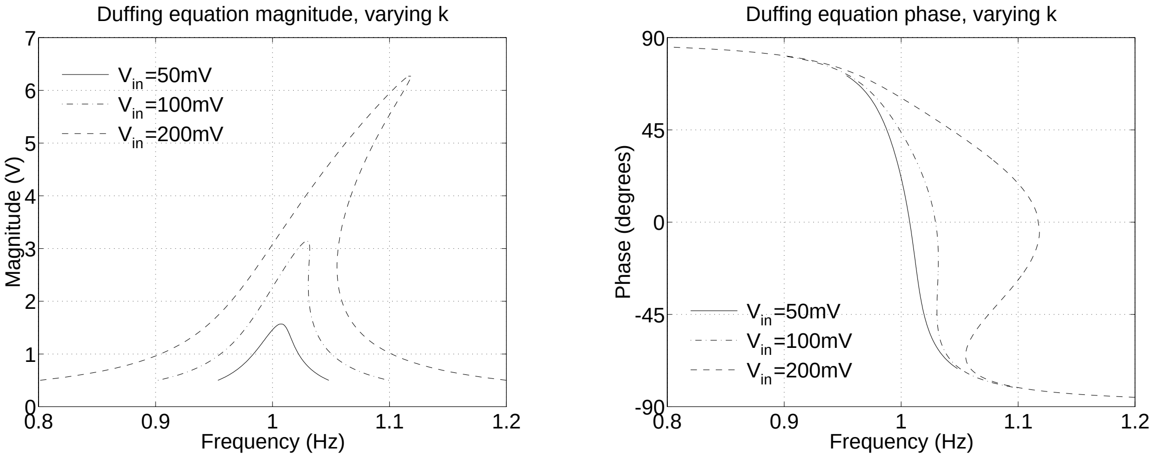


Figure 4.21: Effect of varying input amplitude in Duffing equation.

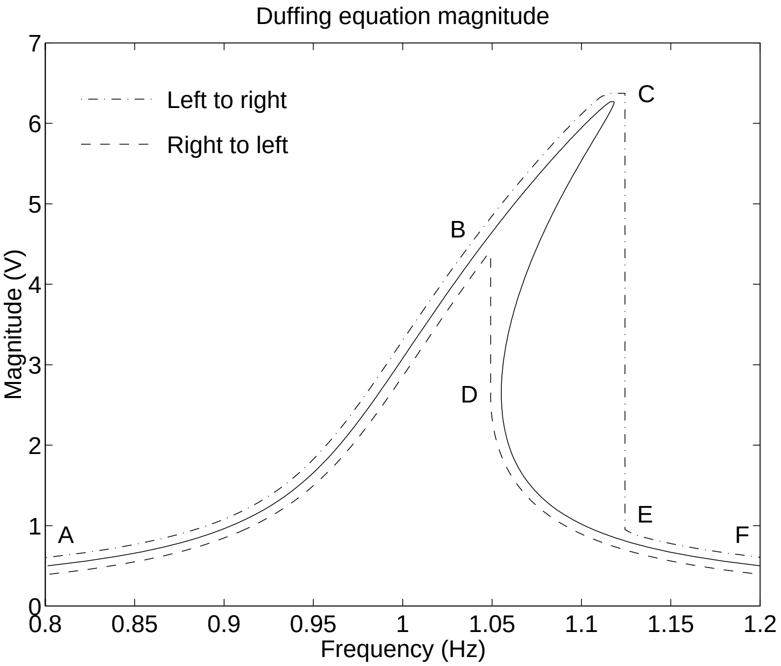


Figure 4.22: Effect of sweeping frequency in Duffing equation.

$v_2 = 6$V yields solutions on the upper portion. The results are illustrated in Figure 4.23. The solid line shows the response calculated using equations (4.35) and (4.34) and the circles show the simulated values. The correspondence is between them good. No pair of initial conditions seems to give rise to the middle solution when three exist; this is as it should be since the middle solution is not stable. If the system were to begin in that state, the slightest perturbation would send it to either the upper or lower solution.

It is not surprising that the error in the Volterra series calculation is large for large inputs; because of our investigation, we now see that for a large input the filter has multiple solutions in a certain frequency range. §3.4.2 discussed this very phenomenon and warned that Volterra series could not be expected to yield accurate results, and the truth of this warning is confirmed.

The author confesses this example was "cooked up": only one of the three TAs was made nonlinear precisely because it makes possible the algebraic analysis carried out in this section. This does not, however, render it any less valid or useful. To demonstrate the power the analysis has given us, we can draw a graph of the maximum allowable filter Q as a function of V_{in} and ϵ_2 that ensures the single-valuedness of the magnitude response.

To generate such a graph, we must return to the positive root for σ, σ_u, in equation (4.35). Returning to Figure 4.22, we recall that the right half of the magnitude response (the line CDEF) is essentially the graph of (σ_u, a) from (4.35) with a bit of shifting and scaling. We can see that the graph is multi-valued if the locus of points (σ_u, a) ever becomes *vertical* — or, to put it another way, if the *slope* of the graph ever becomes *infinite*. The slope is given by $\dfrac{dy}{dx} = \dfrac{da}{d\sigma_u}$, and if we wish to find where this becomes infinite, it is the same as finding where its inverse is zero:

$$\frac{d\sigma_u}{da} = 0 \tag{4.39}$$

Substituting (4.35) into (4.39) and solving for a yields the rather nasty eighth-order

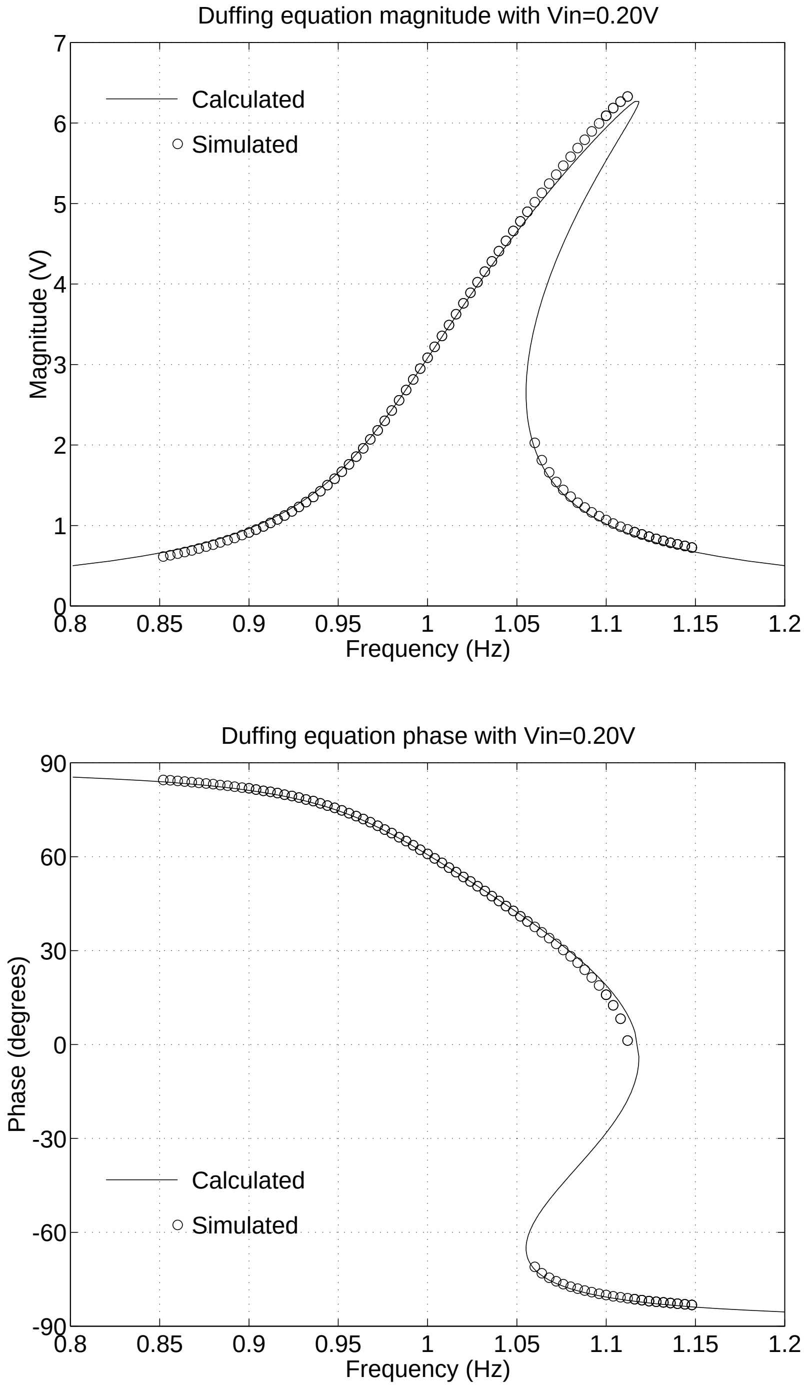


Figure 4.23: Comparison of calculated and simulated Duffing equation.

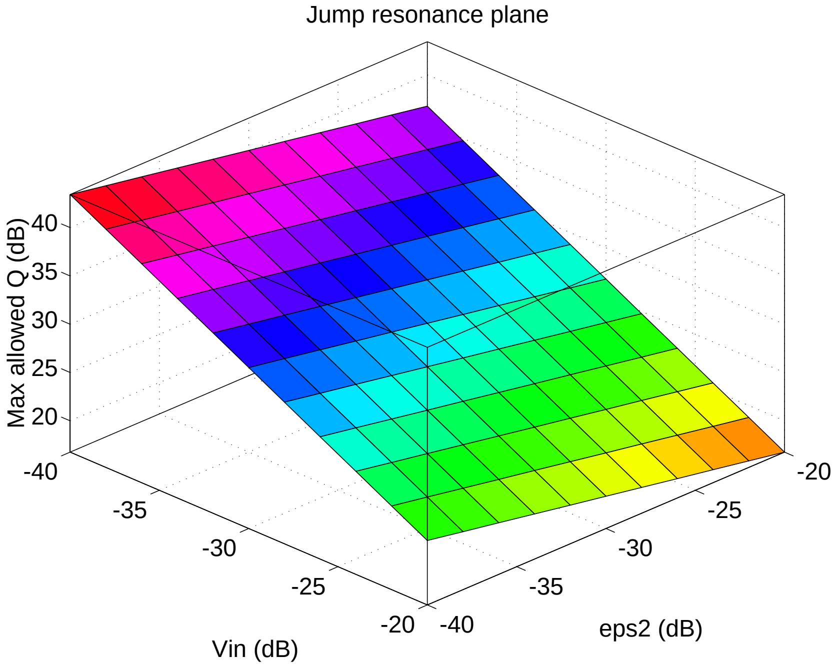


Figure 4.24: Maximum-allowed filter Q before jump resonance in Duffing equation occurs.

polynomial

$$\mu^2 a^8 - \frac{k^2}{4\omega_0^2} a^6 + \frac{k^4}{9\alpha^2\omega_0^2} = 0 \tag{4.40}$$

If this equation has a real root for a in the range $(0, a_{max}) = (0, \frac{k}{2\omega_0\mu})$ from equation (4.37), then we know the graph is multi-valued. If not, the graph is single-valued. Therefore, for a particular pair of (ϵ_2, V_{in}) values, we calculate all the parameters in equation (4.29) and determine the largest value of Q for which (4.40) has no real solutions for a over $(0, a_{max})$. Naturally, numerical root-finding techniques are a prerequisite.

A surface is plotted in Figure 4.24. V_{in} and ϵ_2 have been varied over the range 0.01 to 0.1, and the maximum allowed filter Q in dB, Q_{max}, is also displayed. The graph is consistent with Figure 4.19, Figure 4.20, and Figure 4.21: higher V_{in} and ϵ_2 mean stronger nonlinearity and consequently the Q required for a multi-valued response is lower.

More surprisingly, our surface is planar! We can see that a 20dB-increase in V_{in} results in a 20dB-decrease in Q_{max}. Moreover, a 20dB-increase in ϵ_2 results in a 6.67dB-decrease in Q_{max}. We may therefore conclude

$$Q_{max} \propto \frac{1}{V_{in}\epsilon_2^{1/3}} \tag{4.41}$$

It is not obvious (to the author, at least) how the simple relation in (4.41) falls out of the mathematics, but it certainly agrees with intuition. And the graph gives us a way to tell if we are approaching the region where Volterra series are unlikely to give us accurate numerical results — if we are near the multi-valued response region, Volterra series will probably fail. Truly, our insight *has* been strengthened.

4.5 Two Tone Tests

We shall now examine the performance of a slightly more realistic filter when there are two input sinusoids. As stated at the beginning of §4.4, in a manufactured circuit all TAs would likely have nearly the same nonlinearity and so this time the cubic coefficients will be set to $\epsilon_i = \epsilon_1 = \epsilon_2 = -0.05$. A negative value for ϵ means the *i*-*v* characteristics will bend horizontally rather than vertically for large voltage inputs, and this too is likely to be the case in a manufactured circuit.[4] The other components will retain their values. The new circuit is shown in Figure 4.25.

Let us suppose that the filter is tuned to the desired signal at $f_a = 1$Hz and that there is an interfering signal close by at $f_b = 0.98$Hz. Let us also make their amplitudes equal and fairly small, say $V_a = V_b = 200\mu$V. Because there are two tones now we cannot simply measure the magnitude and phase difference between input and output; we must run the Runge-Kutta simulator and take the FFT of the output.

[4] The circuit examined in §4.4 with positive ϵ_2 is therefore rather unrealistic. A real circuit is much more likely to display a leftward-bending Duffing characteristic rather than the rightward-bending one in Figure 4.22 in §4.4.3.

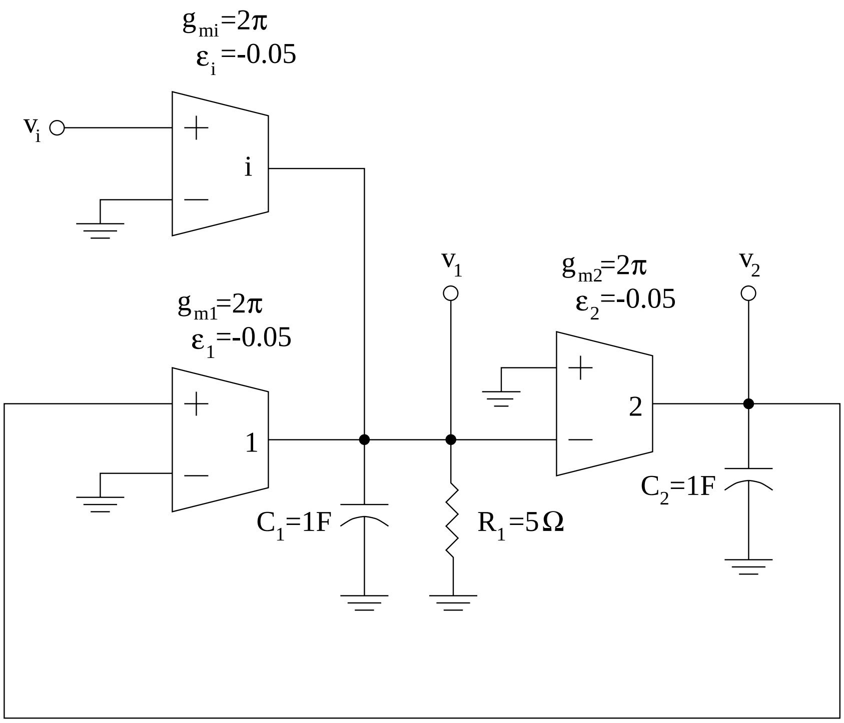


Figure 4.25: Filter for two tone tests.

As long as we follow the guidelines in §4.3, this should give us the magnitudes and phases of each component to good accuracy.

To calculate the expected output component values we must now make use of equation (3.24). To illustrate, we will find all the M_n that contribute to the tone at f_a. These are:

$$\begin{array}{rcrcl}
\frac{V_a}{2}M_1(f_a) & = & 3.1416 \times 10^{-3} & + & j0 \\
\frac{V_a^3}{16}M_3(f_a, f_a, -f_a) & = & 0 & - & j4.6509 \times 10^{-8} \\
\frac{V_a V_b^2}{8}M_3(f_a, f_b, -f_b) & = & 0 & - & j3.6353 \times 10^{-8} \\
\frac{V_a V_b^4}{128}M_5(f_a, f_b, f_b, -f_b, -f_b) & = & -4.0620 \times 10^{-13} & - & j2.3856 \times 10^{-13} \\
\frac{V_a^3 V_b^2}{64}M_5(f_a, f_a, -f_a, f_b, -f_b) & = & -1.2834 \times 10^{-12} & - & j7.7356 \times 10^{-13} \\
\frac{V_a^5}{384}M_5(f_a, f_a, f_a, -f_a, -f_a) & = & -6.8854 \times 10^{-13} & + & j1.5558 \times 10^{-14}
\end{array} \tag{4.42}$$

The sum of these six terms converges rapidly to 6.283185 $\angle -0.002^o$ mV. With more

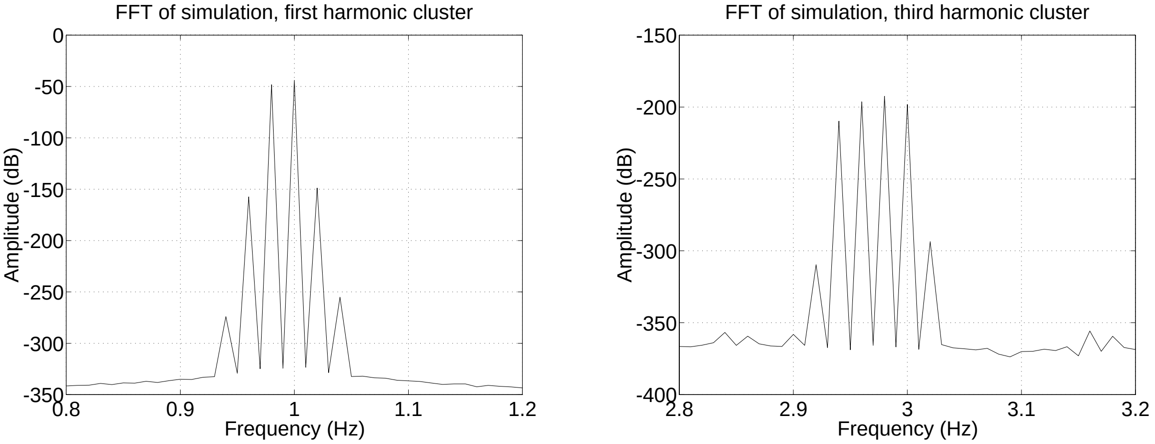


Figure 4.26: FFT of output with two input tones.

than one tone it is easy to miss a contributor to a particular output component. To assist, a program was written that determines all the contributors to a component for any number of input tones.

Graphs of the output frequency spectrum around f_a and $3f_a$ are shown in Figure 4.26. The FFT noise floor is at -340dB which corresponds to an accuracy of 17 decimal places; from the graphs, the tone at f_a is about -40dB, and the tone at $3f_a$ is about -200dB. The values of the amplitudes and phases have been calculated in Table 4.2 and compared with the simulated results. The Table is divided into three sections which correspond to the two first-order components, the six third-order components, and the four fifth-order components depicted in Figure 4.26. (The fifth-order components, incidentally, are the ones on the outer edges of the tone clusters in the Figure.) Noteworthy features of the comparison include:

1. The magnitudes for all first- and third-order components agree to five digits; more remarkably, the fifth-order components agree very well considering their small sizes. The $4f_b - f_a$ component is only 40dB above the FFT noise floor.

2. The phases agree to within 1^o. It is no coincidence that the first-order phase errors are almost identical — this is almost certainly due to the largish time

Table 4.2: Results for two 200μV tones at f_a and f_b.

Freq. (Hz)	Tone	Volterra series calculation (V)	Difference from simulated (V)
1.00	f_a	$6.2832 \quad \times 10^{-3} \quad \angle -0.002^o$	$0.0000 \quad \times 10^{-3} \quad \angle -0.177^o$
0.98	f_b	$3.8881 \quad \times 10^{-3} \quad \angle 51.770^o$	$-0.0001 \quad \times 10^{-3} \quad \angle -0.175^o$
0.96	$2f_b - f_a$	$1.3471 \quad \times 10^{-8} \quad \angle 82.249^o$	$-0.0001 \quad \times 10^{-8} \quad \angle -0.173^o$
1.02	$2f_a - f_b$	$3.6073 \quad \times 10^{-8} \quad \angle 167.012^o$	$0.0000 \quad \times 10^{-8} \quad \angle -0.178^o$
3.00	$3f_a$	$1.2336 \quad \times 10^{-10} \quad \angle 0.681^o$	$0.0000 \quad \times 10^{-10} \quad \angle -0.532^o$
2.94	$3f_b$	$3.2491 \quad \times 10^{-11} \quad \angle 156.002^o$	$-0.0002 \quad \times 10^{-11} \quad \angle -0.526^o$
2.98	$2f_a + f_b$	$2.3725 \quad \times 10^{-10} \quad \angle 52.460^o$	$-0.0000 \quad \times 10^{-10} \quad \angle -0.530^o$
2.96	$f_a + 2f_b$	$1.5208 \quad \times 10^{-10} \quad \angle 104.233^o$	$-0.0001 \quad \times 10^{-10} \quad \angle -0.528^o$
0.94	$3f_b - 2f_a$	$2.0005 \quad \times 10^{-14} \quad \angle 169.291^o$	$-0.0006 \quad \times 10^{-14} \quad \angle -0.175^o$
1.04	$3f_a - 2f_b$	$1.7676 \quad \times 10^{-13} \quad \angle -36.266^o$	$0.0001 \quad \times 10^{-13} \quad \angle -0.176^o$
2.92	$4f_b - f_a$	$3.3187 \quad \times 10^{-16} \quad \angle -174.727^o$	$0.0001 \quad \times 10^{-16} \quad \angle -0.168^o$
3.02	$4f_a - f_b$	$2.1068 \quad \times 10^{-15} \quad \angle 166.512^o$	$0.0001 \quad \times 10^{-15} \quad \angle -0.534^o$

step of 1ms in the Runge-Kutta algorithm. If the time step could be reduced, the phase agreement would improve equally, but it is not feasible to reduce the time step very much because the FFT calculation starts to take too long, and more to the point, run out of memory.

3. Some of the M_n terms for $n > 3$ back in Table 4.1 are complex to derive, and are furthermore far from trivial to calculate correctly on a computer. Skeptics might suspect an error somewhere. But, at least in the case of M_5, if any doubts linger as to the correctness of its derivation, perhaps they can now be laid to rest: the fifth-order components in Table 4.2 are in close enough agreement that the M_5 derived for this filter is almost surely correct.

Of course, any number of additional cases could be simulated and compared, but it would serve little purpose. Only one more result will be shown: the conditions are identical except R_1 is changed from 5Ω to 25Ω, which raises the Q of the filter from about 30 to about 150. Shortly our filters will have high Q and we would like to check the accuracy of Volterra series in those cases. The results are summarized in Table 4.3.

It will be seen that the component at $3f_a$ has risen from 1.2336×10^{-10}V in the $Q = 30$ case to 1.5420×10^{-8}V in the $Q = 150$ case — an increase of $125 = 5^3$ times. Intuitively, this is reasonable: the $3f_a$ component is calculated from $M_3(f_a, f_a, f_a)$, which is the product of three $M_1(f_a)$ terms. Since f_a is the center frequency, $M_1(f_a) \propto Q$, and so if Q increases by a factor of five, the third harmonic should increase by the cube of five.

Apart from that, the errors in the Volterra series calculations are about the same in Table 4.2 and Table 4.3. Again, the magnitudes of the first- and third-order components agree to almost five digits, and the phase errors for individual components are similar (-0.1^o in some cases, -0.5^o in others). Our confidence in Volterra series should be further increased.

Table 4.3: Results for same two tones, but with larger filter Q.

Freq. (Hz)	Tone	Volterra series calculation (V)	Difference from simulated (V)
1.00	f_a	$3.1416 \quad \times 10^{-2} \quad \angle -0.111^o$	$0.0000 \quad \times 10^{-2} \quad \angle -0.166^o$
0.98	f_b	$4.8921 \quad \times 10^{-3} \quad \angle 81.041^o$	$-0.0002 \quad \times 10^{-3} \quad \angle -0.176^o$
0.96	$2f_b - f_a$	$1.1407 \quad \times 10^{-7} \quad \angle 157.741^o$	$-0.0001 \quad \times 10^{-7} \quad \angle -0.186^o$
1.02	$2f_a - f_b$	$1.4364 \quad \times 10^{-6} \quad \angle 107.862^o$	$0.0000 \quad \times 10^{-6} \quad \angle -0.156^o$
3.00	$3f_a$	$1.5420 \quad \times 10^{-8} \quad \angle -0.197^o$	$0.0001 \quad \times 10^{-8} \quad \angle -0.499^o$
2.94	$3f_b$	$6.4761 \quad \times 10^{-11} \quad \angle -116.742^o$	$-0.0007 \quad \times 10^{-11} \quad \angle -0.528^o$
2.98	$2f_a + f_b$	$7.4625 \quad \times 10^{-9} \quad \angle 80.957^o$	$-0.0001 \quad \times 10^{-9} \quad \angle -0.509^o$
2.96	$f_a + 2f_b$	$1.2038 \quad \times 10^{-9} \quad \angle 162.109^o$	$-0.0001 \quad \times 10^{-9} \quad \angle -0.518^o$
0.94	$3f_b - 2f_a$	$2.7804 \quad \times 10^{-13} \quad \angle -45.068^o$	$-0.0051 \quad \times 10^{-13} \quad \angle -0.044^o$
1.04	$3f_a - 2f_b$	$4.9577 \quad \times 10^{-11} \quad \angle -146.892^o$	$0.0001 \quad \times 10^{-11} \quad \angle -0.185^o$
2.92	$4f_b - f_a$	$4.4390 \quad \times 10^{-15} \quad \angle -40.373^o$	$-0.0028 \quad \times 10^{-15} \quad \angle -0.721^o$
3.02	$4f_a - f_b$	$2.0970 \quad \times 10^{-12} \quad \angle 107.984^o$	$0.0001 \quad \times 10^{-12} \quad \angle -0.044^o$

4.6 Three Tone Tests

At this point we are ready to begin evaluating the performance of weakly nonlinear BPFs in the presence of more than one interfering tone. While it is true that at most two tones are required to produce compression, desensitization, and intermodulation *separately*, three tones at once will produce all three effects *simultaneously*. We wish to examine all three types of distortion at once.

So far we have been considering fairly specific numerical examples but we would like to generalize our results. True, we are still considering a particular filter, but we would like to identify the variables that are likely to change from one application to another and examine distortion *trends* as a function of these variables.

What are some of these variables? Six spring to mind:

- the center frequency ω_0,
- the quality factor Q,
- the gain at band center,
- the nonlinearity strength ϵ,
- the signal strength,
- the channel separation.

Let us define these more precisely and write out formulae for them.

Center frequency From equation (4.2), we recall that $\omega_0 = \sqrt{\dfrac{g_{m1}g_{m2}}{C_1C_2}}$ in radians per second and $f_0 = \dfrac{\omega_0}{2\pi}$ in hertz.

Quality factor This, too, is from (4.2): $Q = R_1C_1\omega_0$.

Gain at band center For the linear BPF, the gain peaks at f_0 and has value $A_0 = g_{mi}R_1$, also from (4.2).

Nonlinearity strength For each TA, we have $i = g_m v + \epsilon v^3$, and define the percentage nonlinearity as

$$\begin{aligned} \text{NL\%} &= \frac{\epsilon v^3}{g_m v} \times 100\% \text{ at } v = 10\text{mV} & (4.43) \\ &= \frac{0.01\epsilon}{g_m} \text{ in percent} & (4.44) \end{aligned}$$

Why define it at an input of 10mV? It is somewhat arbitrary, but 10mV is approximately the maximum amplitude signal that an AMPS cellular phone handset can expect to receive.

Signal amplitude In §2.2.2, we defined the desired tone as the one at f_a with amplitude V_a, and the interfering tones as the ones at f_b and f_c with amplitudes V_b and V_c. We will adopt the same definitions here.

Channel separation Define the percentage channel separation as

$$\text{CS\%} = \frac{\triangle f_{min}}{f_0} \times 100\% \qquad (4.45)$$

where $\triangle f_{min}$ is the minimum possible channel separation. So, for example, in AMPS, $\triangle f_{min} = 30$KHz; in FM radio, $\triangle f_{min} = 200$KHz. Furthermore, we will assume the interferers are in one of the three configurations depicted in Figure 4.27: f_a will always be at the center frequency f_0, f_b will be $\triangle f_{min}$ away from f_a, and f_c is $\triangle f_{min}$ away from either f_b or f_c as shown in the Figure.

The nice thing about these definitions is that they can be independent of f_0. That is, a given set of values of Q, A_0, NL%, CS%, and (V_a, V_b, V_c) can be satisfied at *any* value of f_0. More importantly, the key values of M_1 and M_3 remain the same no matter what the value of f_0.

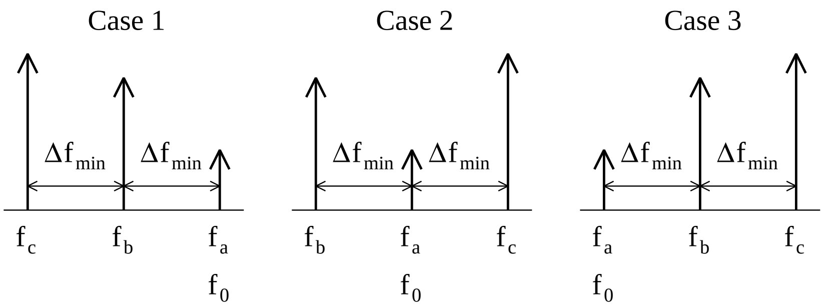


Figure 4.27: Possible configurations for desired and interfering signals.

An example makes this clearer. Consider the conditions under which the results from Table 4.2 in §4.5 were generated. f_0 was 1Hz, and

$$\begin{aligned} Q &= R_1 C_1 \omega_0 = 10\pi \approx 31.4 \\ A_0 &= g_{mi} R_1 = 10\pi \approx 31.4 \\ \text{NL\%} &= \frac{0.01\epsilon}{g_m} = 7.95 \times 10^{-5}\% \\ \text{CS\%} &= \frac{\triangle f_{min}}{f_0} = 2\% \\ V_a &= V_b = 200\mu\text{V} \end{aligned} \tag{4.46}$$

This gave gain and distortion term values of

$$\begin{aligned} \tfrac{V_a}{2} M_1(f_a) &= 3.1416 \times 10^{-3} && + \; j0 \\ \tfrac{V_a^3}{16} M_3(f_a, f_a, -f_a) &= -7.5000 \times 10^{-13} && - \; j4.6509 \times 10^{-8} \\ \tfrac{V_a V_b^2}{8} M_3(f_a, f_b, -f_b) &= -1.5000 \times 10^{-12} && - \; j3.6353 \times 10^{-8} \end{aligned} \tag{4.47}$$

We could now set f_0 to any frequency, and recalculate R_1, g_m, etc., so that all the values in (4.46) were the same. This would give exactly the same values for the three M_n terms in (4.47). In a sense, then, we can remove the significance of center frequency by normalizing it out of the equations.

So, then, we will explore the effects of varying each of Q, A_0, NL%, CS%, and (V_a, V_b, V_c) on the following distortion terms:

- The compression term $T_c = M_3(f_a, f_a, -f_a)$,
- The desensitization terms $T_{d1} = M_3(f_a, f_b, -f_b)$ and $T_{d2} = M_3(f_a, f_c, -f_c)$,
- The intermodulation terms $T_{i1} = M_3(f_b, f_b, -f_c)$ (which arises in cases one and three in Figure 4.27) and $T_{i2} = M_3(-f_a, f_b, f_c)$ (which arises in case two).

Keeping the other four variables fixed we will plot the magnitudes of the distortion terms $(T_c, T_{d1}, T_{d2}, T_{i1}, T_{i2})$ as a function of the fifth variable. It is fortuitous that our definitions make our graphs independent of f_0 because then they are more general — and useful. The default values of Q, A_0, NL%, and CS% will be those in (4.46), and we will use $(V_a, V_b, V_c) = (0.1\text{mV}, 0.5\text{mV}, 1\text{mV})$.

The formulae for M_1 and M_3 from equations (4.14) and (4.17) are repeated here and rewritten slightly in terms of the variables under discussion.

$$\begin{aligned} M_1(f_1) &= \frac{\frac{g_{mi}}{C_1}s}{s^2 + \frac{1}{R_1C_1}s + \frac{g_{m1}g_{m2}}{C_1C_2}} \\ &= \frac{A_0 \times j\omega_1}{j\omega_1 + \frac{Q}{\omega_0}[\omega_0^2 - \omega_1^2]} \text{ where } \omega_1 = 2\pi f_1 \end{aligned} \tag{4.48}$$

$$\begin{aligned} M_3(f_1, f_2, f_3) &= \frac{6\epsilon_i - 6M_1(f_1)M_1(f_2)M_1(f_3)\left[\dfrac{\epsilon_2 g_{m1}}{C_2\sum_3 j\omega_i} + \dfrac{\epsilon_1 g_{m2}^3}{C_2^3\prod_3 j\omega_i}\right]}{C_1\sum_3 j\omega_i + \frac{1}{R_1} + \dfrac{g_{m1}g_{m2}}{C_2\sum_3 j\omega_i}} \\ &= \frac{6\epsilon_i - 6M_1(f_1)M_1(f_2)M_1(f_3)\left[\dfrac{\epsilon_2 g_{m1}}{C_2\sum_3 j\omega_i} + \dfrac{\epsilon_1 g_{m2}^3}{C_2^3\prod_3 j\omega_i}\right]}{\dfrac{C_1}{\sum_3 j\omega_i}\left[(\sum_3 j\omega_i)^2 + \frac{\omega_0}{Q}(\sum_3 j\omega_i) + \omega_0^2\right]} \\ &= \frac{6\epsilon_i - 6M_1(f_1)M_1(f_2)M_1(f_3)\left[\dfrac{\epsilon_2 g_{m1}}{C_2 j\omega_0} + \dfrac{j\epsilon_1 g_{m2}^3}{C_2^3\prod_3 \omega_i}\right]}{\frac{\omega_0 C_1}{Q}} \\ &\quad \text{where } \omega_i = 2\pi f_i \end{aligned} \tag{4.49}$$

The last simplification assumes $f_1 + f_2 + f_3 = f_0$, which holds for all the distortion terms.

4.6.1 Filter Q

The denominator of (4.48) tells us how the linear transfer function $M_1(f_1)$ depends on Q. At a frequency other than ω_0, for small Q, the imaginary part dominates and gain is constant; as Q increases, eventually it will become large enough to cause the real part to dominate and to be inversely proportional to Q. At ω_0, the real part is always zero, so gain is constant. More compactly,

$$\text{For } \omega_1 \neq \omega_0,\ |M_1(f_1)| \ \text{ is } \begin{cases} \text{constant for small } Q \\ \propto \frac{1}{Q} \text{ for large } Q \end{cases} \qquad (4.50)$$

$$\text{For } \omega_1 = \omega_0,\ |M_1(f_1)| \ \text{ is } \ \text{constant for all } Q \qquad (4.51)$$

The distortion terms' behavior is governed by equation (4.49). For small Q, the numerator is constant and the denominator is inversely proportional to Q, which makes $M_3 \propto Q$.

- For the compression term T_c the numerator remains constant for any value of Q because there is a product of three $M_1(f_0)$ terms, and we know $M_1(f_0)$ is constant with Q from (4.51). Thus, $T_c \propto Q$ for all Q.

- For the other distortion terms, increasing Q causes the numerator to change depending on the values of f_1, f_2, and f_3: for any $f_i \neq f_0$, we have $M_1(f_i) \propto \frac{1}{Q}$, and this offsets the Q in the denominator. So M_3 will decrease with Q in the case of the T_d and T_i terms. Eventually, the product of M_1 terms will become small enough that the $6\epsilon_i$ term in the numerator dominates, which makes $M_3 \propto Q$ once again.

The behavior is illustrated in Figure 4.28: T_c rises at a constant $20\frac{\text{dB}}{\text{dec}}$, and the other distortion terms all rise at $20\frac{\text{dB}}{\text{dec}}$ for low Q, fall for medium Q, and rise again at $20\frac{\text{dB}}{\text{dec}}$ for high Q. The T_{i1} term falls at $40\frac{\text{dB}}{\text{dec}}$ because:

- the numerator contains $M_1(f_b)M_1(f_b)M_1(-f_c)$, which means there are three terms in the numerator falling at $20\frac{\text{dB}}{\text{dec}}$,

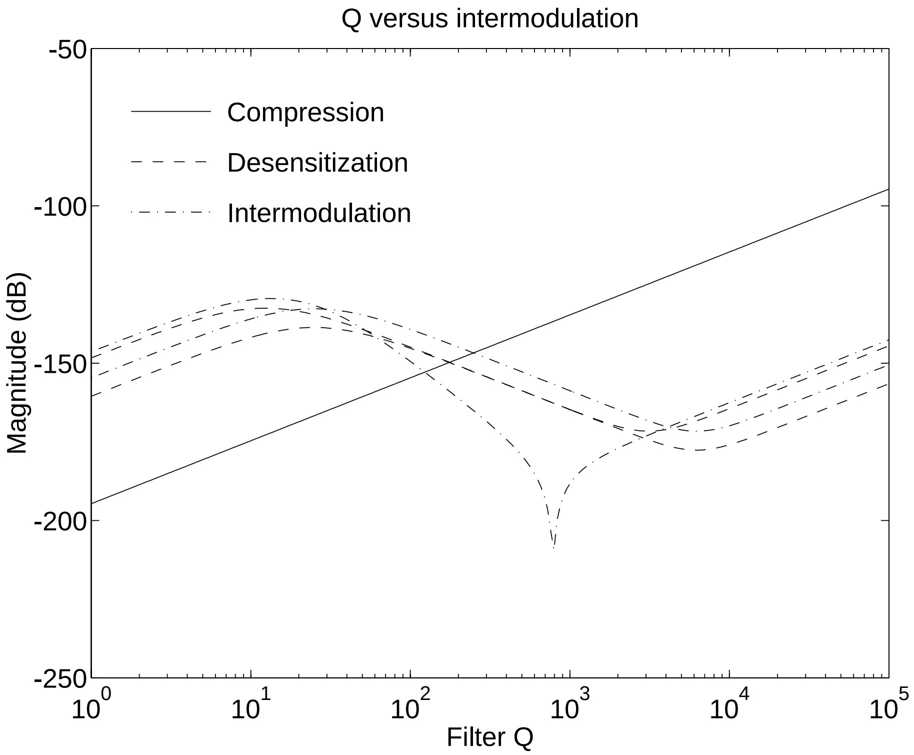


Figure 4.28: Effect of increasing Q on distortion.

- the denominator contains Q, which causes a rise of $20\frac{\text{dB}}{\text{dec}}$,
- the overall effect is that T_{i1} falls by $3 \times 20 - 20 = 40\frac{\text{dB}}{\text{dec}}$.

The other distortion terms all contain one $M_1(f_a)$ term in the numerator, so the numerator has a net $40\frac{\text{dB}}{\text{dec}}$ drop, and the $20\frac{\text{dB}}{\text{dec}}$ rise caused by the Q in the denominator means that overall, the other distortion terms drop at $40 - 20 = 20\frac{\text{dB}}{\text{dec}}$.

The dip in T_{i1} on the graph is interesting because it implies there is an optimum Q at which to operate. However, such a Q may not be possible to reach in practice due to constraints such as limited accuracy in a Q-tuning circuit.

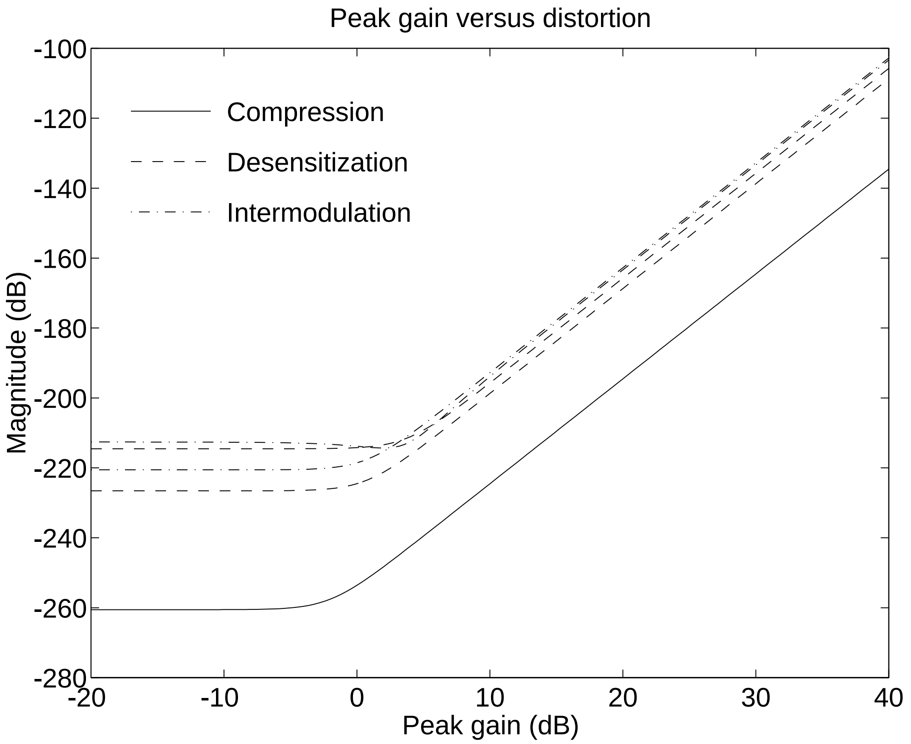


Figure 4.29: Effect of increasing A_0 on distortion.

4.6.2 Peak Filter Gain

In (4.48) we can see that $M_1 \propto A_0$, and so in (4.49), for small A_0 the $6\epsilon_i$ term will dominate and make M_3 constant; increasing A_0 will eventually cause the three M_1 terms to dominate, giving a rise of $20\frac{\text{dB}}{\text{dec}}$ for each M_1 term. This is confirmed in the plot in Figure 4.29: the graphs are all flat for small gains, then they begin to rise at $60\frac{\text{dB}}{\text{dec}}$ as the gain increases.

This may be interpreted by saying that for low A_0 the input stage nonlinearity dominates, while at high A_0 the nonlinearity in the rest of the filter becomes significant.

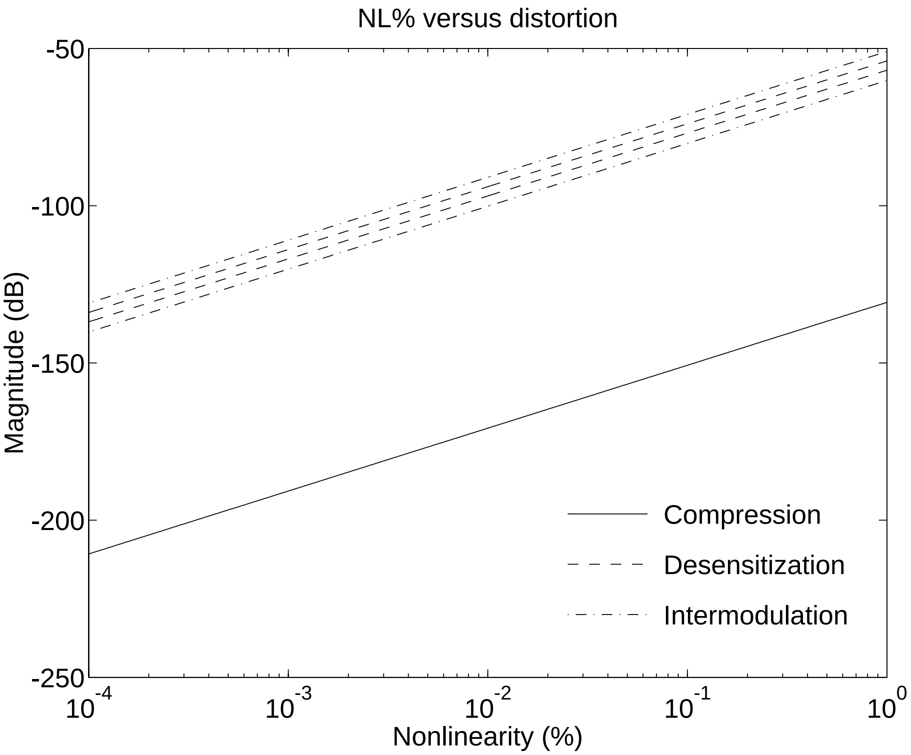


Figure 4.30: Effect of increasing NL% on distortion.

4.6.3 Nonlinearity Strength

The linear transfer function M_1, of course, does not depend on any of the ϵ. Only the nonlinear transfer functions contain ϵ terms, and in (4.49), the numerator is directly proportional to a sum of terms involving ϵ_i, ϵ_1, and ϵ_2. Thus, M_3, and hence distortion, varies directly with ϵ, and since NL% $\propto \epsilon$ we have $M_3 \propto$ NL%. The graph in Figure 4.30 is thus not very exciting: all the distortion terms rise monotonically at $20\frac{\mathrm{dB}}{\mathrm{dec}}$.

4.6.4 Channel Separation

This is perhaps the most important and interesting parameter. As we move the interferers closer to the desired signal, how is the distortion affected? Regrettably our

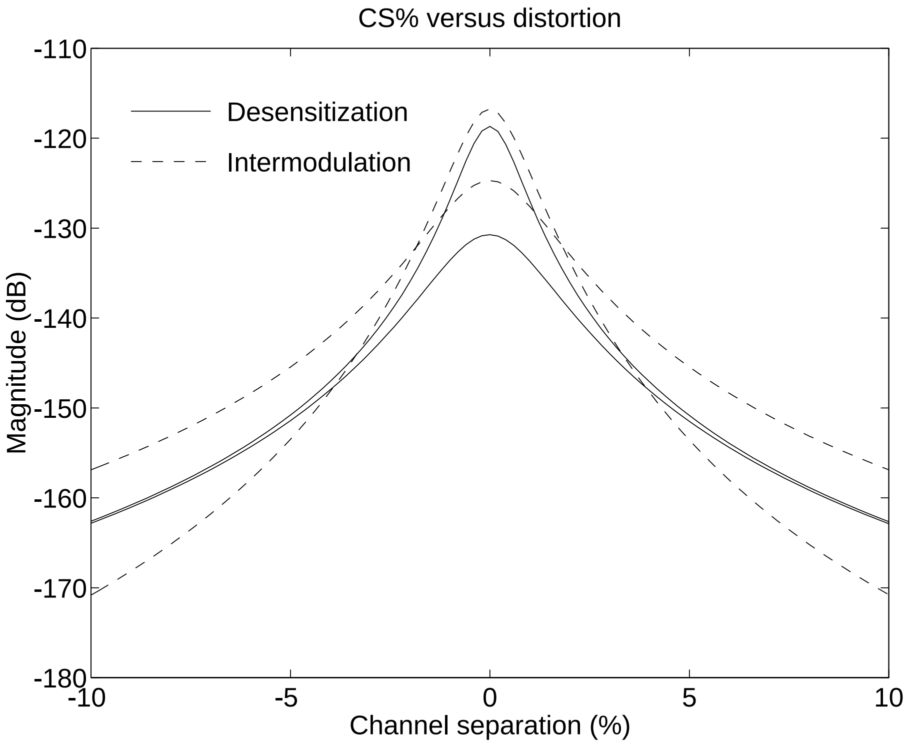


Figure 4.31: Effect of increasing CS% on distortion.

intuition is correct: the closer the adjacent channels, the worse the distortion. This can be understood by observing that M_1 peaks at ω_0 and gets smaller as frequency gets farther away from ω_0; the product of the three M_1 terms in (4.49), then, gets larger when f_1, f_2, and f_3 get closer to ω_0.

A graph of some typical results is shown in Figure 4.31. T_c is not included, of course, because its value is independent of CS%. For close channels, the distortion is between 30dB and 50dB worse than for channels far away.

4.6.5 Signal Strength

The final parameter, signal strength, is rather like NL% in that the results are not very startling. The distortion terms are multiplied by various signal strengths and the dependence is obvious in the formula. So, for example, we have that $T_c = \frac{V_a^3}{16}M_3(f_a, f_a, -f_a)$, and increasing V_a will cause a threefold increase in T_c. As well, $T_{i1} = \frac{V_b^2 V_c}{16}M_3(f_b, f_b, -f_c)$, and so increasing V_a has no effect on T_{i1} while increasing V_b by an amount x will make a change $2x$ in T_{i1}. And so on. The idea is simple enough that a graph is not necessary to illustrate the dependencies.

4.6.6 Summary

Table 4.4 summarizes the results from the previous five sections. None of them are particularly counterintuitive. In an actual circuit we might wonder how distortion changes when *more than* one of the variables changes at the same time. For example, altering the value of R_1 in our example filter changes both $Q = \omega_0 C_1 R_1$ and $A_0 = g_{mi} R_1$ simultaneously. The answer can be obtained from *combining* the results of §4.6.1 and §4.6.2. So, for example, if $Q = 10$ and $A_0 = 10$dB, raising R_1 would make the compression term T_c rise by $80\frac{\text{dB}}{\text{dec}}$, 20 from Q and 60 from A_0.

The next chapter will examine a circuit architecture that attempts to reduce the magnitudes of the distortion terms.

Table 4.4: Summary of graph slopes in dB per decade for three input tones.

	T_c	T_{d1}	T_{d2}	T_{i1}	T_{i2}
small Q	20	20	20	20	20
medium Q	20	-20	-20	-40	-20
large Q	20	20	20	20	20
small A_0	0	0	0	0	0
large A_0	60	60	60	60	60
NL%	20	20	20	20	20
small CS%	–	large magnitude			
large CS%	–	small magnitude			
V_a	60	20	20	0	20
V_b	0	40	0	40	20
V_c	0	0	40	20	20

Chapter 5

An Alternate Filtering Architecture

5.1 Parallel Filters with Feedback

We have seen how weak nonlinearity in a bandpass filter affects the distortion of a desired tone in the presence of interfering tones. Consider now the architecture shown in Figure 5.1, which consists of three parallel (for now, linear) BPFs with transfer functions $A(s)$, $B(s)$, $C(s)$ and center frequencies ω_{0A}, ω_{0B}, ω_{0C}. The outputs are summed and fed back through an amplifier with gain k.

We shall assume that $A(s)$ is centered at the desired signal and the $B(s)$ and $C(s)$ are each centered at an interferer; that is,

$$f_{0A} = f_a, \ f_{0B} = f_b, \ f_{0C} = f_c \tag{5.1}$$

The issue of dynamically tuning each filter is examined for a discrete-time version of this architecture in a paper by Padmanabhan [Pad91] who demonstrates that tuning each filter to track a particular signal is possible. This thesis assumes that tuning has already taken place.

First we will derive the linear transfer functions for the new architecture. Then we

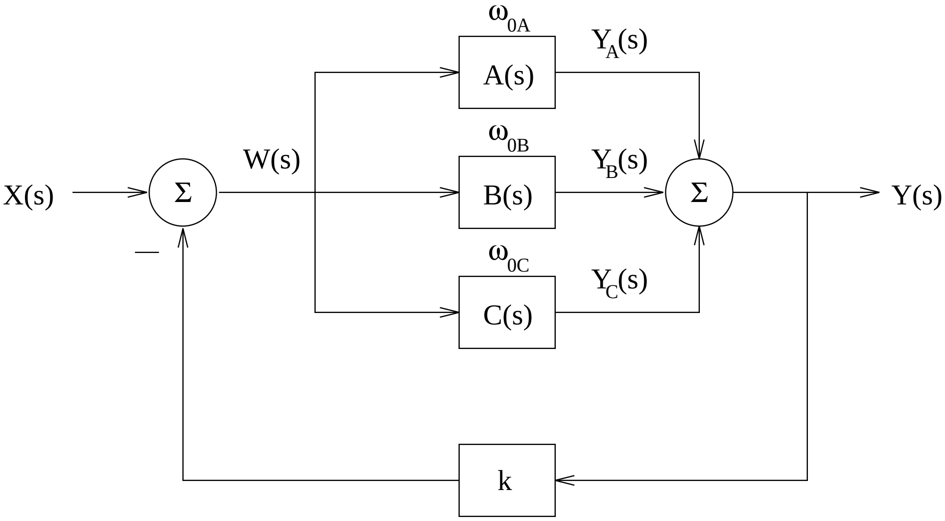


Figure 5.1: Parallel filter implementation.

will allow the filters to be weakly nonlinear and derive the Volterra transfer functions for the system. We will then determine if there is a reduction in distortion for the three filters in parallel over a single filter. We will refer to the three filters with feedback as the "3filt" configuration and a single filter with no feedback as the "1filt" configuration.

5.2 Linear Analysis

5.2.1 Algebraic Derivation

Let us derive the linear transfer functions for the circuit in Figure 5.1. In the frequency domain we have

$$W(s) = X(s) - kY(s) \tag{5.2}$$

$$Y(s) = (A(s) + B(s) + C(s))W(s) \tag{5.3}$$

Substituting (5.3) in (5.2) and solving gives

$$\frac{W}{X} = \frac{1}{1+k(A+B+C)} \tag{5.4}$$

This in (5.3) gives

$$\frac{Y}{X} = \frac{A+B+C}{1+k(A+B+C)}$$

The transfer function to the Y_A output can be found by observing

$$\frac{Y_A}{W} = A \tag{5.5}$$

and putting (5.5) in (5.4):

$$\frac{Y_A}{X} = \frac{A}{1+k(A+B+C)} \tag{5.6}$$

5.2.2 Graphical Interpretation

When the three filters are moderately-high Q (greater than, say, 50), the effect is to *notch out* the signals at ω_{0B} and ω_{0C} at the output Y_A.

Let us quantify this statement. The high-Q assumption means that $|A(\omega_{0A})|$ is much larger than $|B(\omega_{0A})|$ and $|C(\omega_{0A})|$. From (5.6) the transfer function to the Y_A output is then

$$\begin{aligned} \left|\frac{Y_A}{X}\right|_{\omega_{0A}} &= \left|\frac{A}{1+k(A+B+C)}\right|_{\omega_{0A}} \\ &\approx \left|\frac{A}{1+kA}\right|_{\omega_{0A}} \\ &\approx \begin{cases} \frac{1}{k}, & |kA(\omega_{0A})| \gg 1 \\ |A(\omega_{0A})|, & |kA(\omega_{0A})| \ll 1 \end{cases} \end{aligned} \tag{5.7}$$

So, depending on the value of $|kA(\omega_{0A})|$, the gain at the desired signal's frequency will lie between $\frac{1}{k}$ and $A_{0A} = |A(\omega_{0A})|$, the gain at the center frequency of $A(s)$.

At the interferer frequencies ω_{0B} and ω_{0C}, we will assume a large Q for $B(s)$ and $C(s)$, which means $|B(\omega_{0B})|$ dominates $|A(\omega_{0B})|$ and $|C(\omega_{0B})|$, and $|C(\omega_{0C})|$ dominates $|A(\omega_{0C})|$ and $|B(\omega_{0C})|$. Therefore

$$\begin{aligned} \left|\frac{Y_A}{X}\right|_{\omega_{0B}} &\approx \left|\frac{A}{1+kB}\right|_{\omega_{0B}} \\ &\approx \begin{cases} |\frac{A(\omega_{0B})}{kB(\omega_{0B})}|, & |kB(\omega_{0B})| \gg 1 \\ |A(\omega_{0B})|, & |kB(\omega_{0B})| \ll 1 \end{cases} \end{aligned} \tag{5.8}$$

and

$$\begin{aligned} \left|\frac{Y_A}{X}\right|_{\omega_{0C}} &\approx \left|\frac{A}{1+kC}\right|_{\omega_{0C}} \\ &\approx \begin{cases} |\frac{A(\omega_{0C})}{kC(\omega_{0C})}|, & |kC(\omega_{0C})| \gg 1 \\ |A(\omega_{0C})|, & |kC(\omega_{0C})| \ll 1 \end{cases} \end{aligned} \tag{5.9}$$

It is clear that the depths of the notches in $|\frac{Y_A}{X}|$ at the interferer frequencies depend upon the feedback factor k and the center-frequency gain of the interferer filters $A_{0B} = |B(\omega_{0B})|$ and $A_{0C} = |C(\omega_{0C})|$. The notch depths will be proportional to $|kB(\omega_{0B})|$ and $|kC(\omega_{0C})|$ so long as these two quantities are large compared to one.

Some graphs are helpful at this point. The left graph in Figure 5.2 depicts the case where $A_{0A} = A_{0B} = A_{0C} = 10$, $Q_A = Q_B = Q_C = 2\pi 10 \approx 63$, $k = 1$, and the interferers are 2% away from the desired signal (CS% = 2). The solid line shows the $|\frac{Y_A}{X}|$ transfer function with the three filters in parallel and the dotted line shows the transfer function for $A(s)$ alone. The 3filt output has slight notches in it at the interferer frequencies but they are not very pronounced. For this value of CS% the individual filters need higher Q before the notches start to become deep, and such a case is shown in the right graph in Figure 5.2 where $Q_A = Q_B = Q_C = 2\pi 100 \approx 630$. Again, the solid line is for 3filt and the dotted line is for $A(s)$ alone. The notches are now obvious: they are about 20dB deep because we have $|kA_{0B}| = 10$ and $|kA_{0C}| = 10$ and thus from equations (5.8) and (5.9), the 3filt transfer function at the interferer frequencies will be about $\frac{|A|}{10}$, or 20dB below $|A|$. Moreover, from equation (5.7) the

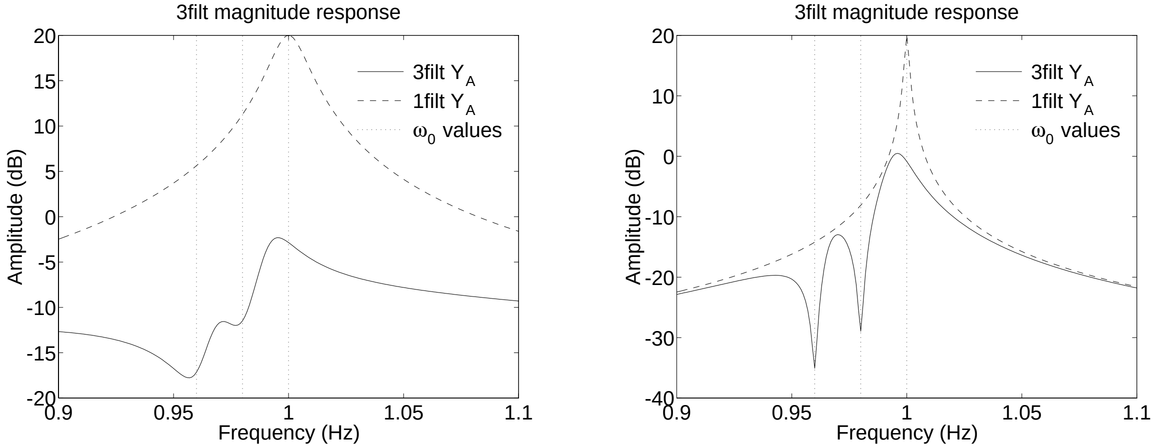


Figure 5.2: Magnitude responses of $Q = 63$ (left) and $Q = 630$ (right) three-filter architecture. $\omega_{0A} = 1.00$Hz, $\omega_{0B} = 0.98$Hz, $\omega_{0C} = 0.96$Hz.

gain at ω_{0A} is about $\dfrac{1}{k} = 1$.

To recapitulate, the effect of satisfying

$$k|A(\omega_{0A})| \gg 1,\ k|B(\omega_{0B})| \gg 1,\ k|C(\omega_{0C})| \gg 1 \tag{5.10}$$

is to, first, create deep notches at the interferer frequencies, and second, to make the gain at the desired signal frequency equal to $\dfrac{1}{k}$.

We could go on varying parameters and plotting graphs all day, but instead let us address the important question: does notching out the interferers improve distortion? Specifically, consider once again the 3filt graph on the right of Figure 5.2. It has been replotted in Figure 5.3 along with two 1filt graphs: both have A_0 values of 0dB but one graph has a Q of about 63 while the other has a Q of about 630.

- The low-Q 1filt graph has the same gain at the desired signal frequency but 20dB *more* gain at each of the interferers.

- The high-Q 1filt graph has the same gain at the desired signal and interferer frequencies but has a higher "effective" Q than the 3filt graph. That is, the 3filt graph is similar to the low-Q graph except for the notches at the interferers, so

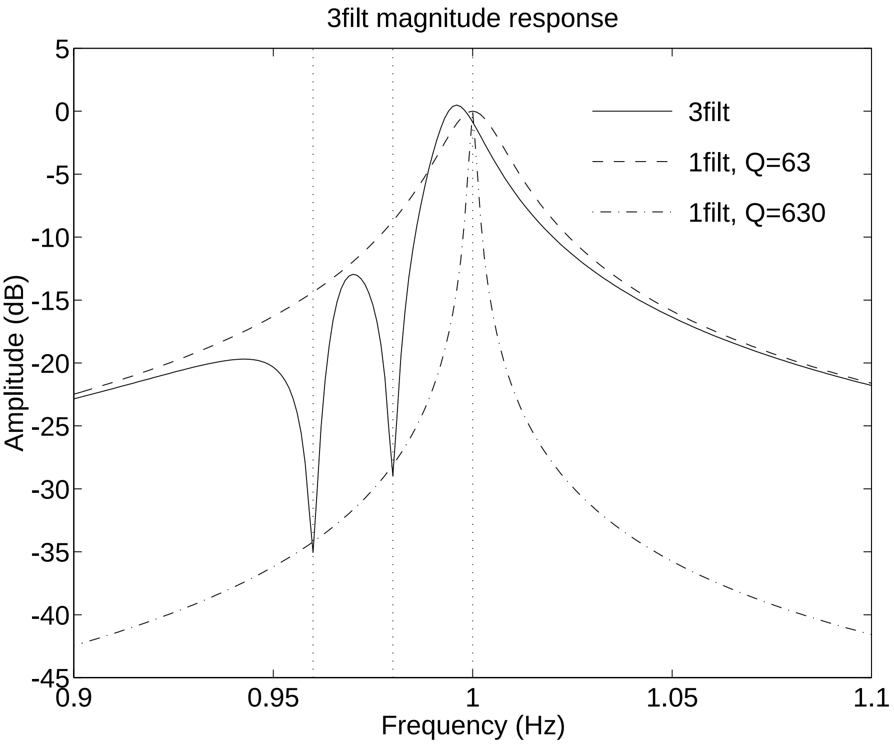


Figure 5.3: Three parallel filters compared with two single filters. $\omega_{0A} = 1.00\text{Hz}$, $\omega_{0B} = 0.98\text{Hz}$, $\omega_{0C} = 0.96\text{Hz}$.

it is as though it has a Q of 63 even though it is comprised of filters with a high Q ($Q = 630$).

We may argue hand-wavingly that in the first case, 3filt, by reducing the *linear* amplitude of the interferers, will also reduce their *nonlinear* amplitudes. In the second case, we may argue equally hand-wavingly from §4.6.1 that having a lower effective Q will make the 3filt distortion lower than the 1filt. Can we solidify our argument?

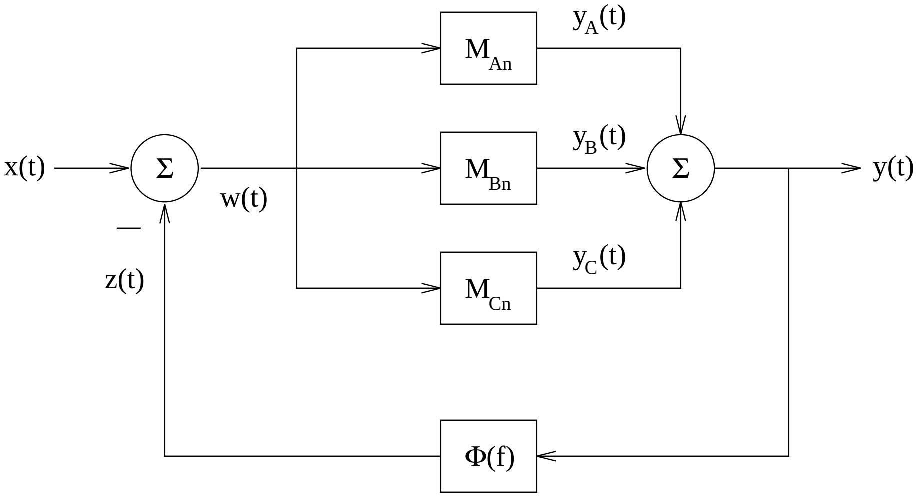


Figure 5.4: Nonlinear filters with feedback.

5.3 Nonlinear Analysis

5.3.1 Algebraic derivation

Let us now derive the Volterra transfer functions for the filters with feedback. A diagram of the system is shown in Figure 5.4, where the time-domain signals are: input $x(t)$, output $y(t)$, individual outputs $y_A(t)$, $y_B(t)$, $y_C(t)$, feedback $z(t)$, and filter input $w(t)$. We will assume the filter Volterra transfer functions M_{An}, M_{Bn}, M_{Cn} are known and that the feedback amplifier $\Phi(f)$ is linear and has a frequency-dependent gain. Lastly, we will assume ideal summers. The various nonlinear transfer functions will be written

$$
\begin{array}{rlrl}
w \to y_A : & M_{An} \qquad & x \to y_A : & H_{An} \\
w \to y_B : & M_{Bn} \qquad & x \to y_B : & H_{Bn} \\
w \to y_C : & M_{Cn} \qquad & x \to y_C : & H_{Cn} \\
x \to w : & K_n \qquad & x \to y : & G_n
\end{array}
$$

Most are shown in Figure 5.5; H_{An} is the one we are most interested in.

To reduce the complexity of the derivation, we will first consider the system in

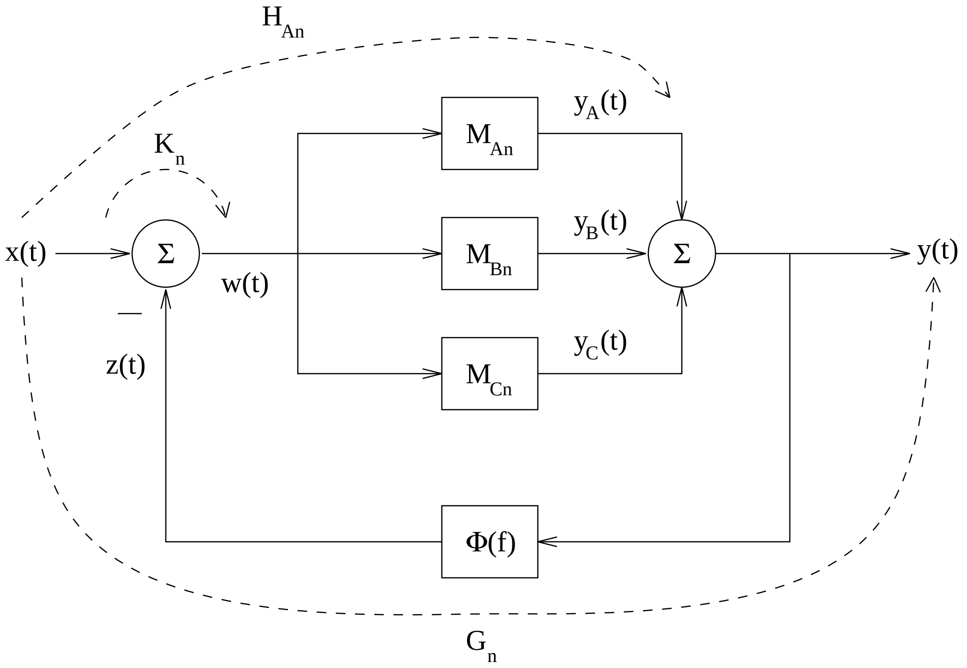


Figure 5.5: Volterra transfer functions for nonlinear filters with feedback.

Figure 5.6. The three summed filters have been replaced by a single nonlinear element M_n; it is trivial to see that

$$M_n = M_{An} + M_{Bn} + M_{Cn} \tag{5.11}$$

That is, the Volterra kernel for the summed output is the same as the sum of the individual Volterra kernels.

We wish to find formulae for general n for the $x \rightarrow w$ transfer functions K_n and the $x \rightarrow y$ transfer functions G_n. The derivations that follow are similar to those in [Bed71], only these ones are more detailed and they extend the authors' work.

Single-Tone Input: K_1 and G_1

Let us start with

$$x(t) = \exp(j\omega_1 t) = |[\omega_1]| \tag{5.12}$$

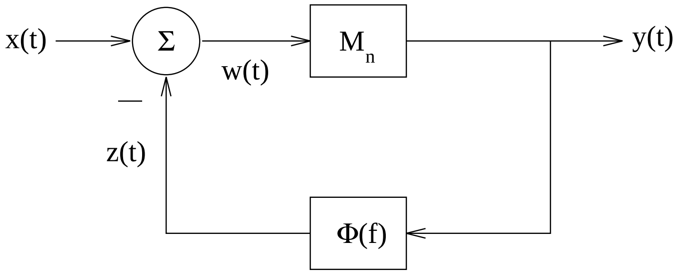


Figure 5.6: Simplified feedback architecture.

where $|[x]|$ is the shorthand notation for $\exp(jxt)$ first used in §3.2.1. Using the harmonic input method, we know that

$$w(t) = K_1(f_1)\exp(j\omega_1 t) = K_1(f_1)|[\omega_1]| \tag{5.13}$$

$$y(t) = G_1(f_1)\exp(j\omega_1 t) = G_1(f_1)|[\omega_1]| \tag{5.14}$$

where K_1 and G_1 have yet to be determined. Note that in Figure 5.6,

$$z(t) = \int_{-\infty}^{\infty} \phi(v) y(t-v) dv \tag{5.15}$$

(5.14) in (5.15) yields

$$\begin{aligned} z(t) &= \int_{-\infty}^{\infty} \phi(v) G_1(f_1) \exp(j\omega_1(t-v)) dv \\ &= G_1(f_1)\exp(j\omega_1 t) \int_{-\infty}^{\infty} \phi(v) \exp(-j\omega_1 v) dv \end{aligned}$$

The integral is simply the Fourier transform of $\phi(t)$ at $\omega_1 = 2\pi f_1$, and so

$$\begin{aligned} z(t) &= G_1(f_1)\exp(j\omega_1 t)\Phi(f_1) \\ &= G_1(f_1)\Phi(f_1)|[\omega_1]| \end{aligned} \tag{5.16}$$

We can also see in Figure 5.6 that

$$w(t) = x(t) - z(t) \tag{5.17}$$

Putting (5.12), (5.13), and (5.16) in (5.17) gives

$$\begin{aligned} K_1(f_1)|[\omega_1]| &= |[\omega_1]| - G_1(f_1)\Phi(f_1)|[\omega_1]| \\ K_1(f_1) &= 1 - G_1(f_1)\Phi(f_1) \end{aligned}$$

That is,

$$\begin{aligned} w(t) &= K_1(f_1)|[\omega_1]| \;\; \text{(from equation (5.13))} \\ &= (1 - G_1(f_1)\Phi(f_1))|[\omega_1]| \end{aligned} \tag{5.18}$$

We have now found K_1 in terms of G_1. To find the latter, note that $y(t)$ is simply $w(t)$ passed through M_n, and so using (5.18),

$$y(t) = (1 - G_1(f_1)\Phi(f_1))M_1(f_1)|[\omega_1]| \tag{5.19}$$

Putting (5.14) in (5.19) gives

$$\begin{aligned} G_1(f_1)|[\omega_1]| &= (1 - G_1(f_1)\Phi(f_1))M_1(f_1)|[\omega_1]| \\ G_1(f_1) &= \frac{M_1(f_1)}{1 + \Phi(f_1)M_1(f_1)} \end{aligned}$$

It scarcely need be pointed out that the overall linear system transfer function G_1 agrees with that which would be derived via traditional frequency-domain means.

Two-Tone Input: K_2 and G_2

Now we make

$$x(t) = \exp(j\omega_1 t) + \exp(j\omega_2 t) = |[\omega_1]| + |[\omega_2]| \tag{5.20}$$

These two tones will circulate through the system and interact due to the nonlinearity, yielding

$$w(t) = K_1(f_1)|[\omega_1]| + K_1(f_2)|[\omega_2]| + K_2(f_1, f_2)|[\omega_1 + \omega_2]| + \cdots \tag{5.21}$$

The output will be

$$y(t) = G_1(f_1)|[\omega_1]| + G_1(f_2)|[\omega_2]| + G_2(f_1, f_2)|[\omega_1 + \omega_2]| + \cdots \tag{5.22}$$

The reader might ask, "Won't there be an infinite number of tones at the output? For example, $|[\omega_1]|$ and $|[\omega_1+\omega_2]|$ interact to produce a tone at $|[2\omega_1+\omega_2]|$. Why isn't this tone listed in (5.22)?" The answer is, it turns out that tones not listed in (5.22) are not needed for deriving a closed-form G_2 or K_2. Let us proceed with equation (5.22) as is for now.

$z(t)$ is once again $y(t)$ passed through $\Phi(f)$. If we substitute (5.22) in (5.15), the first two terms of $y(t)$ in (5.22) will result in $z(t)$ terms like that in (5.16), and the third term of $y(t)$ will result in

$$\begin{aligned} z(t) &= \int_{-\infty}^{\infty} \phi(v) G_2(f_1,f_2)\exp(j(\omega_1+\omega_2)(t-v))dv \\ &= G_2(f_1,f_2)\exp(j(\omega_1+\omega_2)t)\int_{-\infty}^{\infty}\phi(v)\exp(-j(\omega_1+\omega_2)v)dv \\ &= G_2(f_1,f_2)\exp(j(\omega_1+\omega_2)t)\Phi(f_1+f_2) \\ &= G_2(f_1,f_2)\Phi(f_1+f_2)|[\omega_1+\omega_2]| \end{aligned}$$

So

$$z(t) = G_1(f_1)\Phi(f_1)|[\omega_1]| + G_1(f_2)\Phi(f_2)|[\omega_2]| + G_2(f_1,f_2)\Phi(f_1+f_2)|[\omega_1+\omega_2]| \quad (5.23)$$

Substituting (5.20), (5.21), and (5.23) in (5.17) gives

$$\begin{aligned} & K_1(f_1)|[\omega_1]| + K_1(f_2)|[\omega_2]| + K_2(f_1,f_2)|[\omega_1+\omega_2]| \\ = \; & |[\omega_1]| + |[\omega_2]| - G_1(f_1)\Phi(f_1)|[\omega_1]| - G_1(f_2)\Phi(f_2)|[\omega_2]| \\ & - G_2(f_1,f_2)\Phi(f_1+f_2)|[\omega_1+\omega_2]| \end{aligned}$$

Equating terms with $|[\omega_1]|$, $|[\omega_2]|$, and $|[\omega_1+\omega_2]|$, respectively, gives K_1 and K_2:

$$\begin{aligned} K_1(f_1) &= 1 - G_1(f_1)\Phi(f_1) && (5.24) \\ K_1(f_2) &= 1 - G_1(f_2)\Phi(f_2) && (5.25) \\ K_2(f_1,f_2) &= -G_2(f_1,f_2)\Phi(f_1+f_2) && (5.26) \end{aligned}$$

(5.24) and (5.25) agree with the K_1 found in the last section. To find G_2 now, we must find which combinations of terms in $w(t)$ in (5.21) give a component at $|[\omega_1+\omega_2]|$.

Two such combinations exist: $|[\omega_1]|$ and $|[\omega_2]|$ in $w(t)$ produce a $|[\omega_1+\omega_2]|$ term at $y(t)$ through $M_2(f_1, f_2)$, and $|[\omega_1+\omega_2]|$ in $w(t)$ produces a $|[\omega_1+\omega_2]|$ term at $y(t)$ through $M_1(f_1+f_2)$. The amplitudes of these terms at $y(t)$ will be

$$\begin{aligned} |[\omega_1]|,\, |[\omega_2]|: \qquad & K_1(f_1)K_1(f_2)M_2(f_1,f_2) && (5.27) \\ |[\omega_1+\omega_2]|: \qquad & K_2(f_1,f_2)M_1(f_1+f_2) && \\ = \quad & -G_2(f_1,f_2)\Phi(f_1+f_2)M_1(f_1+f_2) && (5.28) \end{aligned}$$

from (5.26). Thus, the overall $|[\omega_1+\omega_2]|$ term at $y(t)$ is the sum of (5.27) and (5.28):

$$(K_1(f_1)K_1(f_2)M_2(f_1,f_2) - G_2(f_1,f_2)\Phi(f_1+f_2)M_1(f_1+f_2))|[\omega_1+\omega_2]| \tag{5.29}$$

But from (5.22) the output component at $|[\omega_1+\omega_2]|$ is also

$$G_2(f_1,f_2)|[\omega_1+\omega_2]| \tag{5.30}$$

Equating (5.29) and (5.30) and solving gives

$$G_2(f_1,f_2) = \frac{K_1(f_1)K_1(f_2)M_2(f_1,f_2)}{1+\Phi(f_1+f_2)M_1(f_1+f_2)}$$

Thus, even though we ignored certain tones in (5.22), the ones we did include were sufficient to obtain a closed form for G_2. It depends on K_1, which in turn depends on G_1, so it is recursive.

Three-Tone Input: K_3 and G_3

Proceeding in the same way as before,

$$x(t) = |[\omega_1]| + |[\omega_2]| + |[\omega_3]| \tag{5.31}$$

$w(t)$ is starting to become more complicated now:

$$\begin{aligned} w(t) &= K_1(f_1)|[\omega_1]| + K_1(f_2)|[\omega_2]| + K_1(f_3)|[\omega_3]| \\ &+ K_2(f_1,f_2)|[\omega_1+\omega_2]| + K_2(f_1,f_3)|[\omega_1+\omega_3]| + K_2(f_2,f_3)|[\omega_2+\omega_3]| \\ &+ K_3(f_1,f_2,f_3)|[\omega_1+\omega_2+\omega_3]| + \cdots \end{aligned} \tag{5.32}$$

As before, we can write

$$\begin{aligned}
K_1(f_1) &= 1 - G_1(f_1)\Phi(f_1) \\
K_1(f_2) &= 1 - G_1(f_2)\Phi(f_2) \\
K_1(f_3) &= 1 - G_1(f_3)\Phi(f_3) \\
K_2(f_1, f_2) &= -G_2(f_1, f_2)\Phi(f_1 + f_2) \\
K_2(f_1, f_3) &= -G_2(f_1, f_3)\Phi(f_1 + f_3) \\
K_2(f_2, f_3) &= -G_2(f_2, f_3)\Phi(f_2 + f_3) \\
K_3(f_1, f_2, f_3) &= -G_3(f_1, f_2, f_3)\Phi(f_1 + f_2 + f_3)
\end{aligned}$$

To find $G_3(f_1, f_2, f_3)$, we find all the terms in (5.32) that interact to give a component at $|[\omega_1 + \omega_2 + \omega_3]|$, add them, and equate them $G_3(f_1, f_2, f_3)$. The interacting terms and their coefficients are

$$\begin{aligned}
|[\omega_1]|,\ |[\omega_2]|,\ |[\omega_3]|: &\quad K_1(f_1)K_1(f_2)K_1(f_3)M_3(f_1, f_2, f_3) \\
|[\omega_1 + \omega_2]|,\ |[\omega_3]|: &\quad K_2(f_1, f_2)K_1(f_3)M_2(f_1 + f_2, f_3) \\
|[\omega_1 + \omega_3]|,\ |[\omega_2]|: &\quad K_2(f_1, f_3)K_1(f_2)M_2(f_1 + f_3, f_2) \qquad (5.33) \\
|[\omega_2 + \omega_3]|,\ |[\omega_1]|: &\quad K_2(f_2, f_3)K_1(f_1)M_2(f_2 + f_3, f_1) \\
|[\omega_1 + \omega_2 + \omega_3]|: &\quad K_3(f_1, f_2, f_3)M_1(f_1 + f_2 + f_3) \\
= &\quad -G_3(f_1, f_2, f_3)\Phi(f_1 + f_2 + f_3)M_1(f_1 + f_2 + f_3)
\end{aligned}$$

The overall $G_3(f_1, f_2, f_3)$ is therefore

$$\begin{aligned}
G_3(f_1, f_2, f_3) &= [1 + \Phi(f_1 + f_2 + f_3)M_1(f_1 + f_2 + f_3)]^{-1} \\
&\quad \times \left\{ \sum\nolimits_3' K_2(f_1, f_2)K_1(f_3)M_2(f_1 + f_2, f_3) \right. \\
&\quad \left. + K_1(f_1)K_1(f_2)K_1(f_3)M_3(f_1, f_2, f_3) \right\}
\end{aligned}$$

where the $\sum_N'$ notation from §3.2.3 reappears.

General G_n and K_n

We can follow the pattern of our work and extrapolate the general formula for the input-output Volterra transfer function G_n:

$$\begin{aligned} G_n(f_1,\ldots,f_n) &= [1+\Phi(f_1+\ldots+f_n)M_1(f_1+\ldots+f_n)]^{-1} \\ &\quad \times\sum_{l=2}^{n}\sum_{(v;l,n)}\sum\nolimits_N' K_{v_1}(f_1,\ldots,f_{v_1}) \\ &\quad \times K_{v_2}(f_{v_1},\ldots,f_{v_1+v_2})\times\ldots \\ &\quad \times K_{v_l}(f_\mu,\ldots,f_n) \\ &\quad \times M_l(\sum_{i=1}^{v_1}f_i,\sum_{i=v_1}^{v_1+v_2}f_i,\ldots,\sum_{i=\mu}^{n}f_i) \end{aligned} \tag{5.34}$$

where $\mu = n - f_{v_l} + 1$ as in §3.2.3. The general formula for the "error" Volterra transfer function K_n is

$$K_n(f_1,\ldots,f_n) = -G_n(f_1,\ldots,f_n)\Phi(f_1+\ldots+f_n) + \begin{cases} 1, & n=1 \\ 0, & n>1 \end{cases} \tag{5.35}$$

The astute reader will observe that we have $x(t)$ connected to $w(t)$ through K_n and $w(t)$ connected to $y(t)$ through M_n — in other words, we have the series connection of the two nonlinear systems depicted in Figure 5.7. It would be useful to have a closed form for the Volterra series of such a connection, because in 3filt the transfer function we wish to find, H_{An}, is made up of a series connection of K_n and M_{An}.

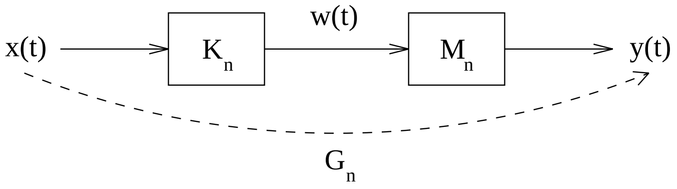


Figure 5.7: Series connection of two nonlinear systems.

Fortunately, we can extrapolate from (5.33) and write the general form for the Volterra transfer function G_n of a series connection of K_n and M_n:

$$\begin{aligned} G_n(f_1,\ldots,f_n) &= \sum_{l=1}^{n}\sum_{(v;l,n)}\sum\nolimits'_N K_{v_1}(f_1,\ldots,f_{v_1}) \\ &\quad \times K_{v_2}(f_{v_1},\ldots,f_{v_1+v_2})\times\ldots \\ &\quad \times K_{v_l}(f_\mu,\ldots,f_n) \\ &\quad \times M_l(\sum_{i=1}^{v_1} f_i, \sum_{i=v_1}^{v_1+v_2} f_i,\ldots,\sum_{i=\mu}^{n} f_i) \end{aligned} \tag{5.36}$$

With unsymmetric kernels, such a nice closed form is not possible to write [Rugh81].

General H_{An}

Returning now to Figure 5.5, recall that we wanted the transfer function from $x \rightarrow y_A$, H_{An}. We can write it as the series connection of K_n and M_{An} using (5.36):

$$\begin{aligned} H_{An}(f_1,\ldots,f_n) &= \sum_{l=1}^{n}\sum_{(v;l,n)}\sum\nolimits'_N K_{v_1}(f_1,\ldots,f_{v_1}) \\ &\quad \times K_{v_2}(f_{v_1},\ldots,f_{v_1+v_2})\times\ldots \\ &\quad \times K_{v_l}(f_\mu,\ldots,f_n) \\ &\quad \times M_{Al}(\sum_{i=1}^{v_1} f_i, \sum_{i=v_1}^{v_1+v_2} f_i,\ldots,\sum_{i=\mu}^{n} f_i) \end{aligned} \tag{5.37}$$

We are really only interested in H_{A1} and H_{A3}, so let us write out these terms explicitly making liberal use of equations (5.11), (5.34) and (5.35).

$$\begin{aligned} H_{A1}(f_1) &= K_1(f_1)M_{A1}(f_1) \\ &= (1-\Phi(f_1)G(f_1))M_{A1}(f_1) \\ &= (1-\frac{\Phi(f_1)M_1(f_1)}{1+\Phi(f_1)M_1(f_1)})M_{A1}(f_1) \\ &= \frac{M_{A1}(f_1)}{1+\Phi(f_1)[M_{A1}(f_1)+M_{B1}(f_1)+M_{C1}(f_1)]} \end{aligned} \tag{5.38}$$

This is, of course, the same as equation (5.6). We will write H_{A3} as

$$H_{A3}(f_1,f_2,f_3) = K_1(f_1)K_1(f_2)K_1(f_3)M_{A3}(f_1,f_2,f_3)$$

$$
\begin{aligned}
&+ K_3(f_1,f_2,f_3)M_{A1}(f_1+f_2+f_3) \\
=\ & K_1(f_1)K_1(f_2)K_1(f_3)M_{A3}(f_1,f_2,f_3) && (5.39)\\
&- \Phi(f_1+f_2+f_3)G_3(f_1,f_2,f_3)M_{A1}(f_1+f_2+f_3) && (5.40)
\end{aligned}
$$

where

$$
\begin{aligned}
K_1(f) &= \frac{1}{1+\Phi(f)[M_{A1}(f)+M_{B1}(f)+M_{C1}(f)]} && (5.41)\\
G_3(f_1,f_2,f_3) &=
\end{aligned}
$$

$$
\frac{K_1(f_1)K_1(f_2)K_1(f_3)[M_{A3}(f_1,f_2,f_3)+M_{B3}(f_1,f_2,f_3)+M_{C3}(f_1,f_2,f_3)]}{1+\Phi(f_1+f_2+f_3)[M_{A1}(f_1+f_2+f_3)+M_{B1}(f_1+f_2+f_3)+M_{C1}(f_1+f_2+f_3)]} \quad (5.42)
$$

are written out for convenience.

5.3.2 Numerical Interpretation

Right now, let us compare 3filt to 1filt for distortion to see if there is any improvement and leave the questions for later.

Table 5.1 compares the three architectures from Figure 5.3: a 3filt built with three $Q = 630$ filters to produce 20dB-notching of the interferers, a 1filt with $Q = 63$, and a 1filt with $Q = 630$. The feedback is a constant $\Phi(f) = 1$ and the nonlinearity in the TAs is set to $\epsilon = -0.05$. f_a is the desired signal while f_b and f_c are the interferers. The numbers in the columns one, two, and four are the magnitudes of the Volterra transfer functions (e.g., the "f_a compression" row has $|H_{A3}(f_a, f_a, -f_a)|$ for 3filt and $|M_{A3}(f_a, f_a, -f_a)|$ for the two different 1filts), and the improvement realized by 3filt is expressed in dB. This is the ratio of the amplitudes of the components that would be found in a numerical simulation of 3filt versus 1filt; for example, we would find the magnitude of the 3filt compression term was 22.91dB smaller than the high-Q 1filt compression term. Or, at least, that is the claim for now; we will verify it later.

Indeed, 3filt shows a marked improvement over both 1filt implementations. We can get a full 20dB more suppression of the interferers than the single low-Q filter,

Table 5.1: Distortion improvement with three parallel filters.

Term	3filt value	Low-Q 1filt value	3filt win (dB)	High-Q 1filt value	3filt win (dB)
f_a linear gain	0.906	1.000	-0.86	1.000	-0.86
f_b linear gain	0.036	0.366	+20.23	0.039	+0.85
f_c linear gain	0.018	0.191	+20.70	0.019	+0.85
f_a compression	4.796	6.708	+2.91	67.082	+22.91
f_a, f_b desensitization	0.070	3.111	+32.92	30.000	+52.60
f_a, f_c desensitization	0.018	3.009	+44.52	30.000	+64.50
f_b, f_c intermodulation	0.064	2.865	+33.01	29.998	+53.41

and our distortion terms are the same or more than an order of magnitude smaller. Moreover, our 3filt distortion terms are all very much smaller than they are with the high-Q filter. Can we see why the improvement results?

We can if we look at equations (5.39), (5.40), (5.41), and (5.42). Let us start with K_1 in equation (5.41). We observe that for each distortion term, the frequency arguments f_1, f_2, f_3 in $H_{A3}(f_1, f_2, f_3)$ are limited to six possibilities: $\pm f_a$, $\pm f_b$, $\pm f_c$. This means that the f in $K_1(f)$ will take on one of these six values. But $K_1(f)$ has $M_{A1}(f) + M_{B1}(f) + M_{C1}(f)$ in the denominator, and f is at the center frequency of one of the three filters. Because the three filters are high Q, the $M_1(f)$ term which has f at its center frequency will dominate the other two. That is,

$$M_{A1}(f) + M_{B1}(f) + M_{C1}(f) \approx \begin{cases} M_{A1}(f), & f = \pm f_a \\ M_{B1}(f), & f = \pm f_b \\ M_{C1}(f), & f = \pm f_c \end{cases} \tag{5.43}$$

Furthermore, equation (5.10) is satisfied for this 3filt. We have that

$$|\Phi(f_a) M_{A1}(f_a)| = |\Phi(f_b) M_{B1}(f_b)| = |\Phi(f_c) M_{C1}(f_c)| = 10 \tag{5.44}$$

Using (5.43) and (5.44) in (5.41) tells us

$$\begin{aligned} |K_1(f)| &\approx \frac{1}{1+10} \text{ at } f = \pm f_a,\, \pm f_b,\, \pm f_c \\ &\approx 0.1 \end{aligned} \tag{5.45}$$

Thus the first line of H_{A3}, equation (5.39)

$$K_1(f_1)K_1(f_2)K_1(f_3)M_{A3}(f_1, f_2, f_3)$$

will be three orders of magnitude smaller than $M_{A3}(f_1, f_2, f_3)$ alone. Let us quickly calculate some values for the compression term to check our arguments.

$$\begin{aligned} |M_{A3}(f_a, f_a, -f_a)| &= \begin{cases} 6.708 \times 10^0, & \text{1filt low } Q \\ 6.708 \times 10^1, & \text{1filt high } Q \\ 6.000 \times 10^4, & \text{3filt} \end{cases} \\ |K_1(f_a)K_1(f_a)K_1(-f_a)M_{A3}(f_a, f_a, -f_a)| &= 4.463 \times 10^1, \text{ 3filt} \end{aligned}$$

These results are consistent: the high-Q 1filt has 20dB more distortion than the low-Q 1filt because Q is ten times higher, and the M_A filter in 3filt has 60dB more distortion than the high-Q 1filt because A_0 is ten times higher. And we do indeed see three orders of magnitude reduction in M_{A3} when it is multiplied by the three K_1 values. It is easy to calculate that this 60dB reduction exists for the desensitization and intermodulation terms as well.

Now, the overall value for H_{A3} is the difference of the equations (5.39) and (5.40). We have examined the first equation already, so let us consider the second equation

$$\Phi(f_1 + f_2 + f_3)G_3(f_1, f_2, f_3)M_{A1}(f_1 + f_2 + f_3)$$

or, because $\Phi(f) = 1$,

$$G_3(f_1, f_2, f_3)M_{A1}(f_1 + f_2 + f_3)$$

We will consider the three separate parts of this equation: the numerator of G_3, the denominator of G_3, and the M_{A1} term.

1. In equation (5.42), we see the numerator of G_3 is made up of a product of K_1 terms and a sum of M_3 terms. We already know from (5.45) that the K_1 terms all evaluate to about 0.1, so that the sum of M_3 terms will be reduced by three orders of magnitude.

2. The denominator of G_3 is the same as the denominator of $K_1(f_a)$ because $f_1 + f_2 + f_3 = f_a$ for all the distortion terms. So this denominator evaluates to about 10.

3. The M_{A1} term also evaluates to 10 because, again, $f_1 + f_2 + f_3 = f_a$, and $M_{A1}(f_a) = 10$ for the 3filt. So this M_{A1} term approximately cancels the term in the denominator of G_3.

The approximate cancellation of M_{A1} with the denominator of G_3 reduces (5.40) to the numerator of G_3 alone:

$$K_1(f_1)K_1(f_2)K_1(f_3)[M_{A3}(f_1,f_2,f_3) + M_{B3}(f_1,f_2,f_3) + M_{C3}(f_1,f_2,f_3)] \tag{5.46}$$

For the compression term $H_{A3}(f_a, f_a, -f_a)$, notice what happens to (5.46): M_{A3} greatly dominates M_{B3} and M_{C3}, and so (5.46) becomes approximately

$$K_1(f_a)K_1(f_a)K_1(-f_a)M_{A3}(f_a, f_a, -f_a) \tag{5.47}$$

which means

$$\begin{aligned} H_{A3}(f_a, f_a, -f_a) &\approx \text{equation (5.39)} - \text{equation (5.47)} \\ &= K_1(f_a)K_1(f_a)K_1(-f_a)M_{A3}(f_a, f_a, -f_a) \\ &\quad - K_1(f_a)K_1(f_a)K_1(-f_a)M_{A3}(f_a, f_a, -f_a) \\ &= 0 \end{aligned} \tag{5.48}$$

$H_{A3}(f_a, f_a, -f_a)$ is not exactly zero because of slight phase shifts, but still, equations

(5.39) and (5.40) will be nearly the same. Let us verify this quickly:

$$\begin{aligned}
K_1(f_a)K_1(f_a)K_1(-f_a)M_{A3}(f_a,f_a,-f_a) &= 2.421 - j44.564 \\
\text{numerator of } G_3(f_a,f_a,-f_a) &= 2.421 - j44.562 \\
\text{denominator of } G_3(f_a,f_a,-f_a) &= 11.020 - j0.604 \\
M_{A1}(f_a+f_a-f_a) &= 10 \\
G_3(f_a,f_a,-f_a)M_{A1}(f_a+f_a-f_a) &= 4.400 - j40.196
\end{aligned}$$

Thinking in vectorial terms, the first and last lines are in almost the same direction, resulting in a small difference:

$$\begin{aligned}
\text{net } H_{A3}(f_a,f_a,-f_a) &= 2.421 - j44.562 \\
&\quad - (4.400 - j40.196) \\
&= -1.979 - j4.366
\end{aligned}$$

For the desensitization and intermodulation terms, M_{A3} in equation (5.46) will no longer dominate M_{B3} and M_{C3}, but whichever term *is* largest, the M_3 sum will still be greatly reduced by the product of the three K_1 terms.

Thus, the reason for the reduction of distortion in 3filt is the three K_1 terms multiplying each of the two H_{A3} components, (5.39) and (5.40). This means we can choose individual filters in 3filt with higher Q and A_0 values; although M_3 for such filters is quite large compared to 1filt implementations with lower Q and A_0, the feedback serves to reduce M_3's severity — greatly so for distortion terms other than the compression term, as Table 5.1 shows.

5.4 Extracting Distortion Terms from Numerical Simulations

It is essential that we devise a method of measuring the distortion terms from simulations, and perhaps from measurements taken in the laboratory. Let us direct our attention to this problem.

Much effort has been focused on measuring Volterra transfer functions. Wiener himself proposed a time-domain method [Wien58], [Lee64] which is useful when the M_n are orthogonal for zero mean white Gaussian input. The harmonic input method [Culb84] may be applied directly in the laboratory as long as n is not too large. These methods are not that quick for getting an overall picture of a Volterra transfer function, and other papers such as [Boyd83] propose methods of quickly measuring $M_2(f_1, f_2)$ at many (f_1, f_2) pairs.

But none of these methods can measure distortion easily. The problem is that the M_3 distortion terms are at the same frequency as the M_1 linear term, and so on a spectrum analyzer the tones overlap. Almost all measurement techniques proposed in the literature do not consider the problem of overlapping tones. Wiener's method might work except that it is cumbersome when the kernels are not orthogonal.

Still, on a spectrum analyzer we can certainly *observe* the effects of compression, desensitization, and the like: if we increase the amplitude of a tone at f_b, we will eventually see a drop in the magnitude of a tone at f_a. Measuring only changes in magnitude, as opposed to changes in magnitude and phase, is there a way to calculate distortion?

5.4.1 Using Magnitude Information Only

It is worth briefly examining the difficulties with this. Suppose we have a weakly nonlinear filter with a single tone f_a at its input of amplitude V_{a1}. If V_{a1} is small enough, the contribution of M_5 will be negligible, and from (4.19) the equation for the output tone at $\omega_a = 2\pi f_a$ will be

$$e^{j\omega_a t}\left[\frac{V_{a1}}{2}M_1(f_a) + \frac{V_{a1}^3}{16}M_3(f_a, f_a, -f_a)\right] + e^{-j\omega_a t}\left[\frac{V_{a1}}{2}M_1(-f_a) + \frac{V_{a1}^3}{16}M_3(-f_a, -f_a, f_a)\right] \tag{5.49}$$

As stated in §4.4.1, if $M_1(f_a) = a + jb$ and $M_3(f_a, f_a, -f_a) = c + jd$, then (5.49) becomes

$$\begin{aligned} & e^{j\omega_a t}\left[\frac{V_{a1}}{2}(a+jb) + \frac{V_{a1}^3}{16}(c+jd)\right] + e^{-j\omega_a t}\left[\frac{V_{a1}}{2}(a-jb) + \frac{V_{a1}^3}{16}(c-jd)\right] \\ = \; & e^{j\omega_a t}\left[(\frac{aV_{a1}}{2} + \frac{cV_{a1}^3}{16}) + j(\frac{bV_{a1}}{2} + \frac{dV_{a1}^3}{16})\right] \\ & + e^{-j\omega_a t}\left[(\frac{aV_{a1}}{2} + \frac{cV_{a1}^3}{16}) - j(\frac{bV_{a1}}{2} + \frac{dV_{a1}^3}{16})\right] \end{aligned} \tag{5.50}$$

Applying (4.20) to (5.50) means the output at ω_a will be

$$\sqrt{\left(\frac{aV_{a1}}{2} + \frac{cV_{a1}^3}{16}\right)^2 + \left(\frac{bV_{a1}}{2} + \frac{dV_{a1}^3}{16}\right)^2} \cos\left(\omega_a t + \arctan\frac{\frac{aV_{a1}}{2} + \frac{cV_{a1}^3}{16}}{\frac{bV_{a1}}{2} + \frac{dV_{a1}^3}{16}}\right) \tag{5.51}$$

On a spectrum analyzer we will only have the magnitude part of (5.51). Label this magnitude V_{out1}. Then

$$V_{out1}^2 = \left(\frac{aV_{a1}}{2} + \frac{cV_{a1}^3}{16}\right)^2 + \left(\frac{bV_{a1}}{2} + \frac{dV_{a1}^3}{16}\right)^2 \tag{5.52}$$

This equation has four unknowns: a, b, c, and d. Theoretically we could try four different input amplitudes V_{a1}, V_{a2}, V_{a3}, V_{a4}, measure the four different output amplitudes V_{out1}, V_{out2}, V_{out3}, V_{out4}, then solve the resulting four (nonlinear) equations.

Let us construct an example: we will arbitrarily make

$$\begin{aligned} M_1(f_a) &= a + jb = 1 + j3 = \sqrt{10}\ \angle 71.57^o \\ M_3(f_a, f_a, -f_a) &= c + jd = 2 + j4 = \sqrt{20}\ \angle 63.43^o \end{aligned} \tag{5.53}$$

Choosing values for V_a of $(1, 2, 3, 5)$, we can find the four corresponding values of V_{out}^2 in (5.52) to be $(\frac{221}{64}, 29, \frac{9621}{64}, \frac{117125}{64})$. This means our four equations are

$$\begin{aligned} V_{out1}, V_{a1}: \quad \frac{221}{64} &= \left(\frac{1}{2}a + \frac{1}{16}c\right)^2 + \left(\frac{1}{2}b + \frac{1}{16}d\right)^2 \\ V_{out2}, V_{a2}: \quad 29 &= \left(a + \frac{1}{2}c\right)^2 + \left(b + \frac{1}{2}d\right)^2 \\ V_{out3}, V_{a3}: \quad \frac{9621}{64} &= \left(\frac{3}{2}a + \frac{27}{16}c\right)^2 + \left(\frac{3}{2}b + \frac{27}{16}d\right)^2 \\ V_{out4}, V_{a4}: \quad \frac{117125}{64} &= \left(\frac{5}{2}a + \frac{125}{16}c\right)^2 + \left(\frac{5}{2}b + \frac{125}{16}d\right)^2 \end{aligned} \tag{5.54}$$

Table 5.2: Calculated values from Maple.

Solution	$a + jb$
1	$2.3737 \; - \; j2.0894 \; = \; \sqrt{10} \; \angle \; -41.35^o$
2	$-1.4895 \; + \; j2.7895 \; = \; \sqrt{10} \; \angle \; 118.10^o$
3	$1.9180 \; - \; j2.5142 \; = \; \sqrt{10} \; \angle \; -52.66^o$
4	$-2.0194 \; + \; j2.4335 \; = \; \sqrt{10} \; \angle \; 129.69^o$

Solution	$c + jd$
1	$2.0953 \; - \; j3.3998 \; = \; \sqrt{20} \; \angle \; -49.48^o$
2	$-2.6432 \; + \; j3.6075 \; = \; \sqrt{20} \; \angle \; 126.23^o$
3	$3.1880 \; - \; j3.1363 \; = \; \sqrt{20} \; \angle \; -44.53^o$
4	$-2.3404 \; + \; j3.8108 \; = \; \sqrt{20} \; \angle \; 121.56^o$

Using Maple to solve these four symbolic equations numerically with fsolve() yields four different solutions for (a, b, c, d) — none of which agree with our chosen values! Table 5.2 summarizes the results.

Even though we did not succeed in finding the exact (a, b, c, d) values, Maple *did* succeed in finding the correct magnitudes and *relative* phase for $a + jb$ and $c + jd$. The magnitudes in (5.53) were $\sqrt{10}$ and $\sqrt{20}$, and Maple found these. And, M_1 leads M_3 by 8.13^o in (5.53), and the solutions in the Table all exhibit a phase difference of $\pm 8.13^o$. For our purposes, magnitude and relative phase are good enough: obviously we need the correct magnitudes, but the exact angles are unimportant.

This example has not fully delved the complexity of using magnitude information only; a couple more problems are included in the following summary of the disadvantages of this method.

1. The equations involved are nonlinear and so do not have closed-form solutions.

We require numerical solution techniques, and these are not always reliable.

2. We need a total of four measurements to extract the compression term. Although it has not been shown here, the other distortion terms will require at least two measurements each to extract. It would be nice to improve on this.

3. With a bit more experimentation, it can be shown that when (a, b, c, d) vary by orders of magnitude it becomes much more difficult for numerical algorithms to converge to a solution. We have seen that M_1 and M_3 can easily differ by three orders of magnitude, so this is a severe disadvantage.

How can we correct these deficiencies?

5.4.2 Using Both Magnitude and Phase Information

If we know both the magnitude and the phase of each component, extracting distortion terms becomes trivial. Both pieces of data are obtainable from an FFT of a simulation or from a network analyzer (rather than a spectrum analyzer) in the laboratory.

As in the previous section, assume we have a single-tone input of amplitude V_{a1} and frequency f_a and that $M_1(f_a) = a + jb$, $M_3(f_a, f_a, -f_a) = c + jd$, and M_5 is negligible. Then we can write the output tone at ω_a using (5.50):

$$e^{j\omega_a t}\left[(\frac{aV_{a1}}{2} + \frac{cV_{a1}^3}{16}) + j(\frac{bV_{a1}}{2} + \frac{dV_{a1}^3}{16})\right] + e^{-j\omega_a t}\left[(\frac{aV_{a1}}{2} + \frac{cV_{a1}^3}{16}) - j(\frac{bV_{a1}}{2} + \frac{dV_{a1}^3}{16})\right] \tag{5.55}$$

To write (5.55) in rectangular coordinates with time suppressed, we may apply (4.20) and think of the result as a phasor, and we arrive at

$$\begin{aligned} & 2\left(\frac{aV_{a1}}{2} + \frac{cV_{a1}^3}{16}\right) + j2\left(\frac{bV_{a1}}{2} + \frac{dV_{a1}^3}{16}\right) \\ = & \left(aV_{a1} + \frac{cV_{a1}^3}{8}\right) + j\left(bV_{a1} + \frac{dV_{a1}^3}{8}\right) \end{aligned} \tag{5.56}$$

But recall that we know the output's magnitude and phase. We can write it in rectangular coordinates as, say, $V_{Rout1} + jV_{Iout1}$, and this equals (5.56):

$$V_{Rout1} + jV_{Iout1} = \left(aV_{a1} + \frac{cV_{a1}^3}{8}\right) + j\left(bV_{a1} + \frac{dV_{a1}^3}{8}\right) \tag{5.57}$$

The real part of (5.57) involves the unknowns a and c while the imaginary part involves b and d. Now suppose we alter the amplitude of the input to V_{a2} and that the output becomes $V_{Rout2} + jV_{Iout2}$. M_1 and M_3 don't change, so (a, b, c, d) are the same. Rewriting (5.57) with the new input and output yields

$$V_{Rout2} + jV_{Iout2} = \left(aV_{a2} + \frac{cV_{a2}^3}{8}\right) + j\left(bV_{a2} + \frac{dV_{a2}^3}{8}\right) \tag{5.58}$$

Again, (5.58) has a and c in the real part and b and d in the imaginary part. Separating real and imaginary parts of (5.57) and (5.58) and pairing each part:

$$\begin{array}{rcl} V_{Rout1} & = & aV_{a1} + \frac{cV_{a1}^3}{8} \\ V_{Rout2} & = & aV_{a2} + \frac{cV_{a2}^3}{8} \end{array}, \quad \begin{array}{rcl} V_{Iout1} & = & bV_{a1} + \frac{dV_{a1}^3}{8} \\ V_{Iout2} & = & bV_{a2} + \frac{dV_{a2}^3}{8} \end{array} \tag{5.59}$$

Each pair of equations is linear in the unknown variables, and so both are easy to solve for (a, b, c, d). The final result is

$$M_1(f_a) = a + jb = \frac{V_{a1}^3 V_{Rout2} - V_{a2}^3 V_{Rout1}}{V_{a1}^3 V_{a2} - V_{a1} V_{a2}^3} + j\frac{V_{a1}^3 V_{Iout2} - V_{a2}^3 V_{Iout1}}{V_{a1}^3 V_{a2} - V_{a1} V_{a2}^3} \tag{5.60}$$

$$M_3(f_a, f_a, -f_a) = c + jd = 8\frac{V_{a1} V_{Rout2} - V_{a2} V_{Rout1}}{V_{a1} V_{a2}^3 - V_{a1}^3 V_{a2}} + j8\frac{V_{a1} V_{Iout2} - V_{a2} V_{Iout1}}{V_{a1} V_{a2}^3 - V_{a1}^3 V_{a2}} \tag{5.61}$$

So: to find the compression term we need only solve two linear equations instead of solving four nonlinear equations. The equations are explicit and do not require a numerical equation solver. And large differences in magnitude between M_1 and M_3 no longer matter. We have overcome all the disadvantages of using magnitude information alone.

What about the other distortion terms? Consider desensitization first. Suppose the input has two tones at f_a and f_b of amplitude V_{a1} and V_{b1}. Assuming the tones are small enough so that M_5 terms are negligible, the output tone at ω_a can be written using the first three terms of (4.39) as

$$\begin{aligned}
& V_{Rout1} + jV_{Iout1} \\
= \; & e^{j\omega_a t}\left[\frac{V_{a1}}{2}M_1(f_a) + \frac{V_{a1}^3}{16}M_3(f_a, f_a, -f_a) + \frac{V_{a1}V_{b1}^2}{8}M_3(f_a, f_b, -f_b)\right] \\
& + e^{-j\omega_a t}\left[\frac{V_{a1}}{2}M_1(-f_a) + \frac{V_{a1}^3}{16}M_3(-f_a, -f_a, f_a) + \frac{V_{a1}V_{b1}^2}{8}M_3(-f_a, -f_b, f_b)\right] \\
= \; & \left(aV_{a1} + c\frac{V_{a1}^3}{8} + e\frac{V_{a1}V_{b1}^2}{4}\right) + j\left(bV_{a1} + d\frac{V_{a1}^3}{8} + f\frac{V_{a1}V_{b1}^2}{4}\right) \qquad (5.62)
\end{aligned}$$

where $M_3(f_a, f_b, -f_b) = e + jf$. If we have already found (a, b, c, d) then theoretically we could solve the real part of (5.62) for e and the imaginary part for f, but it is no doubt clear to the reader that errors will tend to propagate this way. That is, if (a, b, c, d) are slightly off, e and f will be off as well because they depend on a through d. We can greatly reduce error propagation by making a second measurement: keep V_a the same, but alter V_b to some other value V_{b2}. This will yield

$$V_{Rout2} + jV_{Iout2} = \left(aV_{a1} + c\frac{V_{a1}^3}{8} + e\frac{V_{a1}V_{b2}^2}{4}\right) + j\left(bV_{a1} + d\frac{V_{a1}^3}{8} + f\frac{V_{a1}V_{b2}^2}{4}\right) \qquad (5.63)$$

Subtracting (5.62) from (5.63), equating real and imaginary parts, and solving for e and f gives

$$M_3(f_a, f_b, -f_b) = e + jf = \frac{4}{V_{a1}}\frac{V_{Rout2} - V_{Rout1}}{V_{b2}^2 - V_{b1}^2} + j\frac{4}{V_{a1}}\frac{V_{Iout2} - V_{Iout1}}{V_{b2}^2 - V_{b1}^2} \qquad (5.64)$$

So doing two measurements is sufficient to remove the dependence of e and f on (a, b, c, d). If we were using magnitude information only, doing two measurements would suffice for calculating e and f but it would not remove their dependence on (a, b, c, d) — in fact, *any* number of measurements will not remove the dependence because the equations are nonlinear and the dependence cannot be subtracted out as it was in equation (5.64). Thus, a further advantage of using both magnitude and phase is that we can stem calculation error carry-through.

The procedure for measuring the other desensitization term, $M_3(f_a, f_c, -f_c)$, is identical with that for $M_3(f_a, f_b, -f_b)$. The last kind of distortion term we have to measure is intermodulation: $M_3(f_b, f_b, -f_c)$. This one turns out to be the easiest to find. If we apply two tones f_b and f_c with magnitudes V_b and V_c at the input, then if the M_5 terms are negligible, the output at ω_a will be

$$\begin{aligned} V_{Rout} + jV_{Iout} &= e^{j\omega_a t}\left[\frac{V_b^2 V_c}{16} M_3(f_b, f_b, -f_c)\right] + e^{-j\omega_a t}\left[\frac{V_b^2 V_c}{16} M_3(-f_b, -f_b, f_c)\right] \\ &= g\frac{V_b^2 V_c}{8} + jh\frac{V_b^2 V_c}{8} \qquad (5.65) \end{aligned}$$

where $M_3(f_b, f_b, -f_c) = g + jh$ and we are freely mixing time-explicit and time-suppressed forms. We can write M_3 directly from (5.65) as

$$M_3(f_b, f_b, -f_c) = g + jh = 8\frac{V_{Rout}}{V_b^2 V_c} + j8\frac{V_{Iout}}{V_b^2 V_c} \qquad (5.66)$$

A single measurement followed by application of (5.66) will give us the intermodulation term value.

5.5 Examples of Numerical Extraction

We will use both the Runge-Kutta program and SPICE to simulate some 3filt configurations and extract the Volterra distortion terms using the formulae developed in the previous section.

5.5.1 Simple 3filt

Let us start with the 3filt from §5.3.2. We will try to verify our calculations in the first column of Table 5.1. We will be using an expanded Runge-Kutta program that implements the 3filt circuit mathematically, and in conjunction with this we will use Matlab to calculate the FFT of the $y_A(t)$ output. This will give us the magnitude and phase of the tone at ω_a from which we can extract the distortion terms.

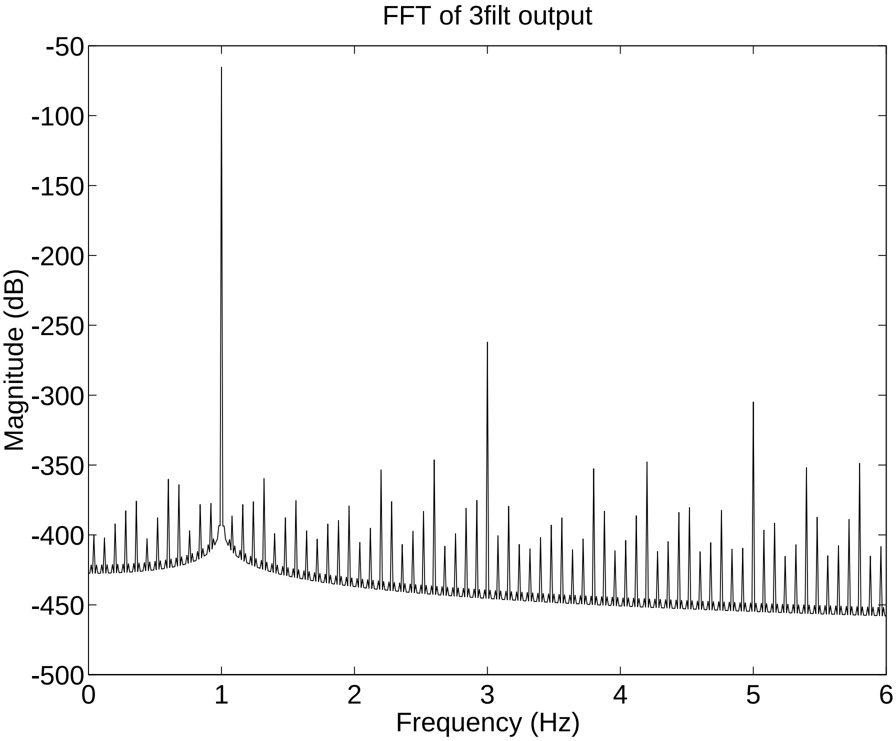


Figure 5.8: FFT of 5000 second 3filt simulation.

Recall that the three filters have $A_0 = 10$, $Q = 630$, $\epsilon = -0.05$, and the channels are 2% apart: $f_{0A} = 1.00\text{Hz}$, $f_{0B} = 0.98\text{Hz}$, $f_{0C} = 0.96\text{Hz}$. The feedback is a constant $\Phi(f) = 1$. The calculated values for linear gain and compression are

$$\begin{aligned} H_{A1}(f_a) &= 0.90470 + j0.04960 = 0.90606 \quad \angle 3.14^o \\ H_{A3}(f_a, f_a, -f_a) &= -1.97946 - j4.36825 = 4.79582 \quad \angle -114.38^o \end{aligned} \tag{5.67}$$

To measure these values from simulation we must choose two different amplitudes V_{a1} and V_{a2} that are small enough so that M_5 is negligible.[1] We will try $V_{a1} = 0.6\text{mV}$, $V_{a2} = 1.2\text{mV}$, and a time step of 1ms. How long should we simulate for?

Figure 5.8 shows the FFT of the final 100 seconds of a 5000 second simulation

[1] We must also remember the points from §4.3: to choose a suitable time step and to simulate for long enough for transients to die away.

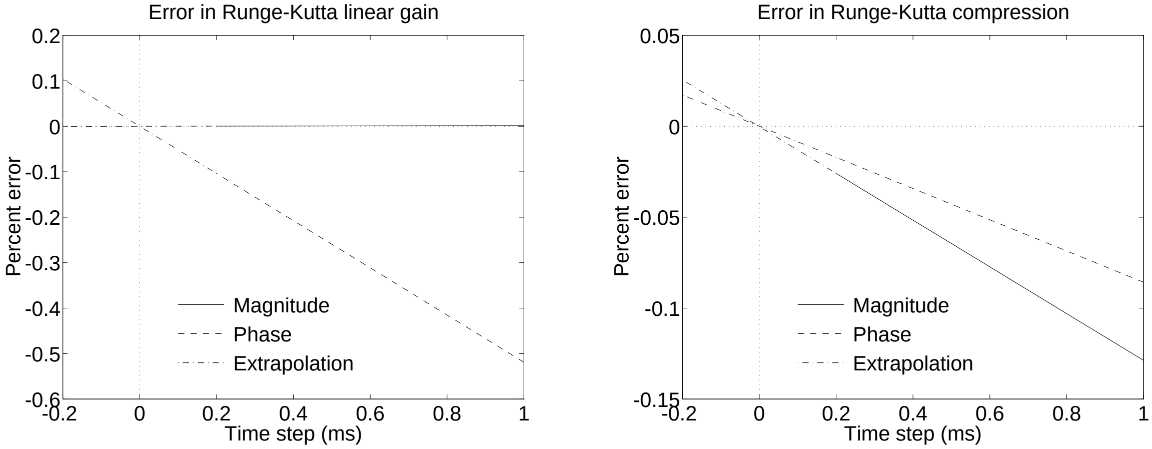


Figure 5.9: Runge-Kutta error versus time step for 3filt.

with V_{a1}. The noise floor is very low and the third harmonic (the tone at 3Hz) is well below the fundamental. It looks as though these are good measurement conditions. Changing the input to V_{a2} and applying equations (5.60) and (5.61) to the measured output tone for each input condition yields

$$\begin{aligned} H_{A1}(f_a) &= \quad 0.04934 - j0.90473 = 0.90607 \quad \angle{-86.88^o} \\ H_{A3}(f_a, f_a, -f_a) &= -4.36601 + j1.96944 = 4.78965 \quad \angle 155.72^o \end{aligned} \tag{5.68}$$

The magnitudes are quite close to the calculated ones in (5.67), although they are not exact, and the phases seem to be shifted clockwise by 90^o. This is not a serious deficiency — again, as long as the *relative* phases agree, we have nothing to worry about.

Harkening back to §4.4 for a moment, it was shown that step size in the Runge-Kutta algorithm makes a difference: in Figure 4.11, we saw that phase varied linearly with time step and that extrapolating the time step to zero gave extremely close agreement between simulation and calculation. The good news is, the same thing happens with the 3filt simulation. Figure 5.9 shows the difference between the calculated and simulated magnitudes and phases (with the 90^o shift taken into account) for the linear gain and compression terms — and now *both* errors go to zero linearly as time step does! The smallest time step that can be simulated reasonably is 0.2ms, so the curve for steps smaller than this is extrapolated. Not only does the extraction method work, but it agrees perfectly with the calculation.

Moving on to the desensitization terms, the calculated values for this 3filt are

$$\begin{aligned} H_{A3}(f_a, f_b, -f_b) &= 0.00693 - j0.06998 = 0.07032 \angle -84.34^o \\ H_{A3}(f_b, f_c, -f_c) &= -0.00038 - j0.01787 = 0.01787 \angle -91.22^o \end{aligned}$$

Setting $V_{a1} = 0.6\text{mV}$, choosing $(V_{b1}, V_{b2}) = (V_{c1}, V_{c2}) = (0.6\text{mV}, 1.2\text{mV})$, and using the last 100 seconds of a 5000-second simulation with a time step of 1ms, we can measure the tone at both input conditions and apply (5.64) to find desensitization values of

$$\begin{aligned} H_{A3}(f_a, f_b, -f_b) &= -0.06999 - j0.00692 = 0.07033 \angle -174.36^o \\ H_{A3}(f_a, f_c, -f_c) &= -0.01787 + j0.00039 = 0.01788 \angle 178.76^o \end{aligned}$$

The agreement between calculation and simulation is very good if the 90^o phase shift is kept in mind. Lastly, the calculated intermodulation term is

$$H_{A3}(f_b, f_b, -f_c) = 0.00890 - j0.06341 = 0.06403 \angle -82.01^o$$

A single simulation with $(V_b, V_c) = (6\text{mV}, 6\text{mV})$ and application of (5.66) leads to

$$H_{A3}(f_a, f_b, -f_b) = -0.06342 - j0.00887 = 0.06404 \angle -172.04^o$$

Clearly, the proposed distortion-term measurement method works well. Not only that, but the H_{An} derived in §5.3.1 is correct.

5.5.2 Realistic 3filt

Now comes the challenging part: to apply the measurement technique to a more realistic circuit in SPICE. We will investigate the circuit in [Nguy92] that was subsequently designed at this institution and manufactured in a 0.8μm BiCMOS process. The essence of the circuit is shown in Figure 5.10; it is a monolithic BPF in which both the center frequency f_0 and the quality factor Q can be tuned. A brief explanation of the circuit operation follows.

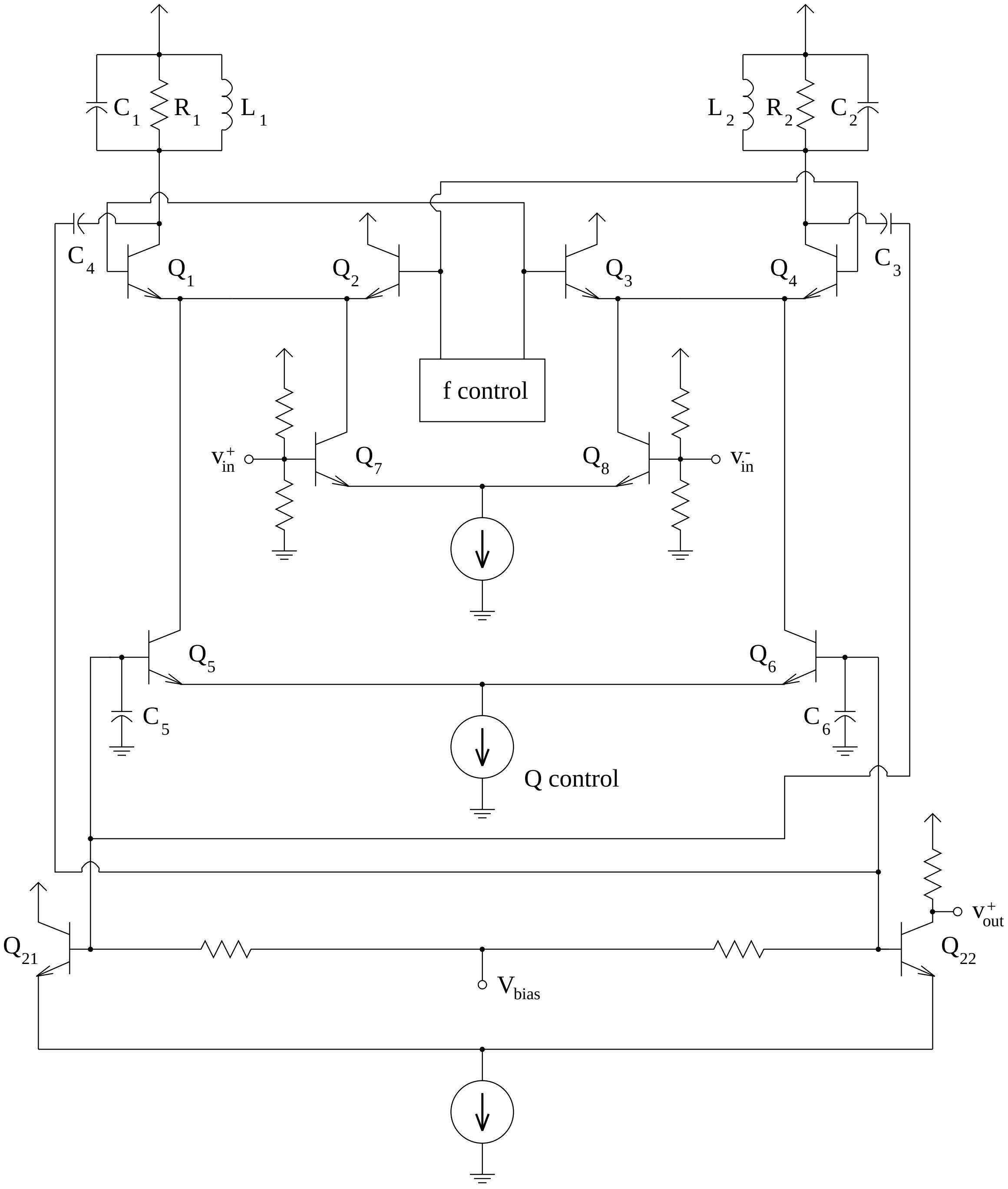


Figure 5.10: Realistic circuit for testing extraction method.

Circuit Operation

The input is applied at the bases of differential pair transistors Q_7 and Q_8. Q_7 modulates the current in resonant circuit $L_1C_1R_1$ through differential pair Q_1/Q_2, and the output is capacitively coupled through C_4 to output transistor Q_{22}. Q_8 operates with the other resonant circuit $L_2C_2R_2$. Differential-to-single-ended conversion is accomplished at the differential pair formed by transistors Q_{21} and Q_{22}.

The two tank circuits are designed with slightly different resonant frequencies, in this case approximately $f_{01} = 1.70$GHz and $f_{02} = 1.95$GHz, and the frequency control circuitry determines their relative influence. If transistors Q_1 and Q_3 are turned on full, then Q_2 and Q_4 will be off and the center frequency will be $f_0 \approx f_{01}$. If Q_2 and Q_4 are on full then Q_1 and Q_3 will be off and $f_0 \approx f_{02}$. f_0 can be varied smoothly between f_{01} and f_{02} by appropriate setting of the frequency control.

Monolithic inductors are notorious for their low Q values, typically between three and eight [Nguy90], and the R_1 and R_2 in the tank circuits represent the resistive loss of each inductor. Transistors Q_5 and Q_6 are positive feedback devices with negative resistance that offset R_1 and R_2. When their bias current is increased, the system poles are moved from the left-half plane towards the $j\omega$-axis and ultimately into the right-half plane. Capacitor pairs (C_3, C_5) and (C_4, C_6) control the point at which the poles cross the $j\omega$-axis as well as set the bias level for Q_5 and Q_6.

Simulation Conditions

Given that the circuit is tunable between about 1.70GHz and 1.95GHz we must choose a center frequency f_0 for simulation that will be FFT-friendly. Let us arbitrarily say we want an input tone at f_0 to appear in the 100th FFT bin; this means we must choose its period to be an integer multiple of the simulation time step *and* that we much take 100 output periods of data for our FFT. Our time step also has a constraint in that it must be a small enough fraction of the period to give us accurate results (recall Figure 4.4) but large enough that our simulations do not take too long.

To satisfy all the conditions we can choose an input period of $T_0 = 546\text{ps}$ and a time step of 2ps (0.36% of the period). This makes our filter center frequency $f_0 = \dfrac{1}{T_0} = 1.831501832\text{GHz}$. Nine decimal places are necessary to ensure low FFT noise floors as was brought out in §4.3.3. Taking $100T_0 = 54.6\text{ns}$ of the transient output data for our FFT will then put a tone at f_0 into the 100th FFT bin as desired.

We have now chosen our desired signal frequency f_a. What about the interferer frequencies f_b and f_c? We have been using CS% = 2% until now; let us attempt a more aggressive value of 1%. This means that our frequencies will be

$$\begin{aligned} f_a &= f_0 &&= 1.831501832\text{GHz} \\ f_b &= 0.99f_0 &&= 1.813186813\text{GHz} \\ f_c &= 0.98f_0 &&= 1.794871795\text{GHz} \end{aligned} \tag{5.69}$$

The tones will appear in FFT bins 100, 99, and 98, respectively, and they are all within the tuning range of the filter.

It would be most useful to do a comparison like the one in §5.3.2 where we find the distortion for a high-Q 1filt, a low-Q 1filt, and a 3filt. To this end three different SPICE files were constructed, one for each structure. After trial and error manipulation of the frequency control voltage and Q control current the ac analyses depicted in Figure 5.11 were obtained. The relevant parameters for each filter are:

$$\begin{array}{rrcllcll} \text{Low-}Q\text{ 1filt:} & A_0 &=& 38.1\text{dB}, & Q &=& 90.1 & \\ \text{High-}Q\text{ 1filt:} & A_0 &=& 38.0\text{dB}, & Q &=& 817.6 & \\ \text{3filt:} & \text{all } A_0 &=& 58.0\text{dB}, & \text{all } Q &=& 817.6, & \Phi(f) = 0.01 \\ & \text{gain at } f_0 &=& 38.9\text{dB} & & & & \end{array} \tag{5.70}$$

The conditions are quite similar to those in §5.3.2:

- the filters all have approximately the same gain at f_a,
- the low-Q 1filt has about ten times less Q than the high-Q 1filt,
- the "effective" 3filt Q is about the same as that of the low-Q 1filt (that is, their gains follow each other approximately except for the notches at the interferers),

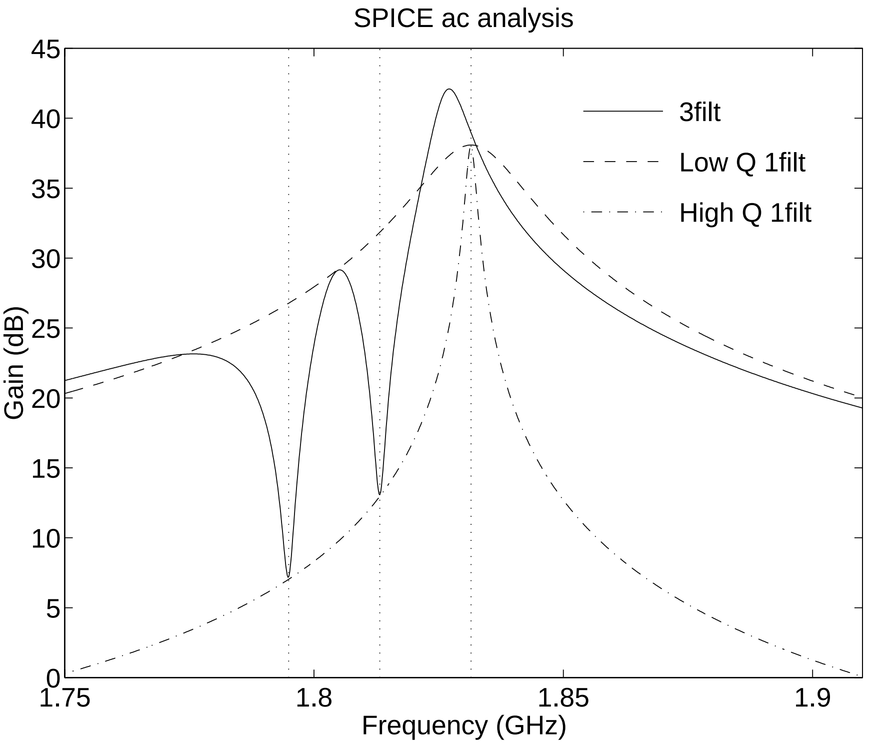


Figure 5.11: SPICE ac analyses of three realistic filter configurations.

- the interferers are notched about 20dB more by 3filt than the low-Q 1filt, and
- the interferers have about the same gain in 3filt and the high-Q 1filt.

The peak 3filt gain occurs slightly away from f_a, and this is an artifact of the exact k and A_0 values in the linear transfer function. This suggests that off-tuning f_{0A} to move the gain peak to f_a is worth considering for future work. For now, let the simulations begin!

Slightly Different Extraction Method

The main difference between this simulation and the one in §5.3.2 is that now we don't have formulae for what the *expected* distortion values are. This difficulty is easily overcome, however. Consider the method for extracting the compression term:

Table 5.3: Low-Q 1filt linear gain and compression terms.

Input pair	$\lvert M_1(f_a)\rvert$	$\lvert M_3(f_a, f_a, -f_a)\rvert$
(V_{a1}, V_{a2})	7.8503×10^1	5.7356×10^8
(V_{a1}, V_{a3})	7.8503×10^1	5.7351×10^8
(V_{a1}, V_{a4})	7.8503×10^1	5.7336×10^8
(V_{a2}, V_{a3})	7.8503×10^1	5.7348×10^8
(V_{a2}, V_{a4})	7.8503×10^1	5.7333×10^8
(V_{a3}, V_{a4})	7.8503×10^1	5.7327×10^8

we pick two different input amplitudes V_{a1} and V_{a2}, simulate both, then take the output tone for each case and apply equations (5.60) and (5.61). No matter what two amplitudes we pick the calculated compression value $M_3(f_a, f_a, -f_a)$ should remain the same as long as M_5 is small enough to be disregarded.

Therefore, the way to check our results is to pick *three* input amplitudes V_{a1}, V_{a2}, and V_{a3} and apply equations (5.60) and (5.61) to each pair of amplitudes (V_{a1}, V_{a2}), (V_{a1}, V_{a3}), and (V_{a2}, V_{a3}). The calculated M_3 values should agree if we do things properly. For additional certainty we could pick *four* input amplitudes and apply the equations to the six possible amplitude pairings.

Let us try it with the low-Q 1filt: choosing the values $V_{a1} = 5\mu$V, $V_{a2} = 7\mu$V, $V_{a3} = 10\mu$V, $V_{a4} = 15\mu$V, and using the output voltage from the transient simulation data between 1550ns and 1604.6ns, taking its FFT, and applying the the compression formulae to each pair of results leads to the values shown in Table 5.3. The linear gain terms agree to at least five decimal places while the compression terms agree to about three. The proposed method seems to work.

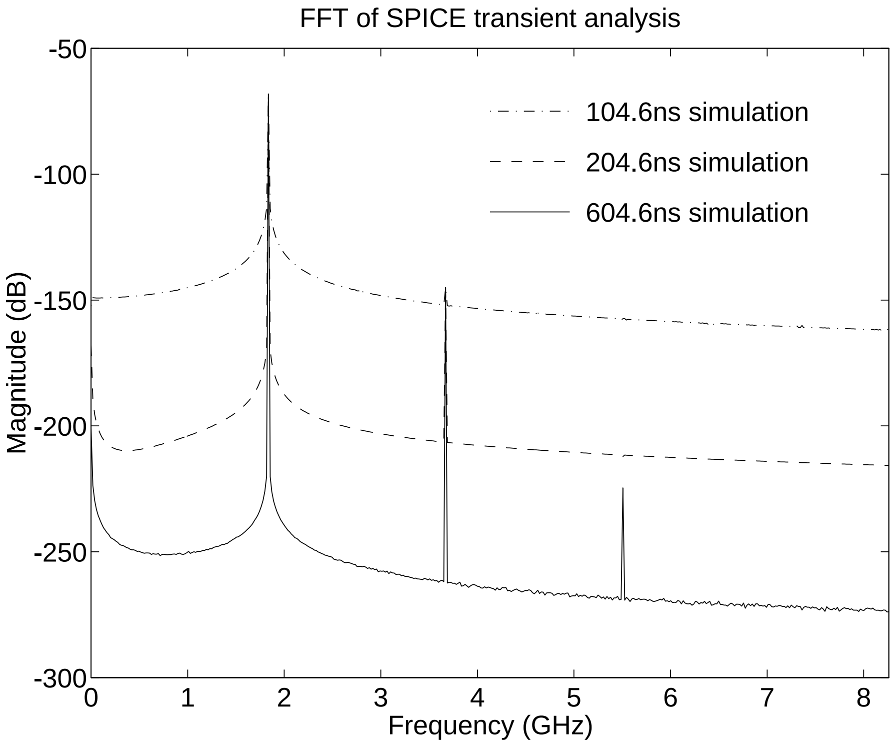


Figure 5.12: FFTs of simulations that are too short for reliable extraction.

Things to Watch Out For

Care must be taken when extracting data in this way. This section highlights some things that can go wrong.

First and foremost the transients *must* have settled. (This statement has been repeated often enough that it could almost be the chorus for this thesis.) In case the point has not been made clearly enough yet a graph showing the FFTs of some short simulations are shown in Figure 5.12. The simulations that last for 100ns or 200ns have quite high FFT noise floors; what happens when we try to extract the compression term from simulations like this?

Each of the three simulation lengths depicted in the Figure was simulated for three input amplitudes $V_{a1} = 5\mu\text{V}$, $V_{a2} = 7\mu\text{V}$, $V_{a3} = 10\mu\text{V}$. The compression term

Table 5.4: Low-Q 1filt compression term as a function of simulation length.

Input	Simulation length			
pair	104.6ns	204.6ns	604.6ns	1604.6ns
(V_{a1}, V_{a2})	4.8845×10^8	5.7310×10^8	5.7355×10^8	5.7356×10^8
(V_{a1}, V_{a3})	4.9920×10^8	5.7305×10^8	5.7349×10^8	5.7351×10^8
(V_{a2}, V_{a3})	5.0442×10^8	5.7302×10^8	5.7347×10^8	5.7348×10^8

extracted from each pair for each simulation time are shown in Table 5.4. The last column of the Table is repeated from Table 5.3 for reference. If all we did was simulate to 104.6ns we might assume a compression term magnitude of about 5.0×10^8, but this is clearly incorrect: the longer simulations all agree that it is 5.73×10^8. Ensuring our transients have settled is even more crucial now that we have no formulae to compare our extracted values to. The last paragraph of §4.3.2 alluded to this fact: for high-Q circuits, long simulations are a prerequisite for meaningful results.

Long simulations are not the only criterion. The input amplitudes we use can be neither too small nor too large. If they are too small, the M_3 which we are trying to extract becomes buried in the FFT noise. If they are too large, M_5 starts to become significant which will throw off any M_3 calculation.

The best illustration of errors induced by small inputs is in the extraction of the (f_a, f_b) desensitization term for 3filt. Table 5.5 shows the extracted M_3 term for two different sets of input conditions: in the first,

$$V_a = 3\mu\text{V and } (V_{b1}, V_{b2}, V_{b3}) = (3\mu\text{V}, 5\mu\text{V}, 7\mu\text{V})$$

while in the second

$$V_a = 10\mu\text{V and } (V_{b1}, V_{b2}, V_{b3}) = (5\mu\text{V}, 10\mu\text{V}, 15\mu\text{V})$$

Equation (5.64) is applied to each of the three pairs of output values to obtain the

Table 5.5: 3filt desensitization term with inputs too small.

V_a Value	Input pair	$\lvert M_3(f_a, f_b, -f_b)\rvert$
3μv	(V_{b1}, V_{b2})	2.2956×10^7
	(V_{b1}, V_{b3})	2.7539×10^7
	(V_{b2}, V_{b3})	3.2946×10^7
10μV	(V_{b1}, V_{b2})	2.4728×10^7
	(V_{b1}, V_{b3})	2.4610×10^7
	(V_{b2}, V_{b3})	2.4541×10^7

Table 5.6: Low-Q 1filt compression term with inputs too large.

Input pair	$\lvert M_3(f_a, f_a, -f_a)\rvert$
(V_{a1}, V_{a2})	5.1897×10^8
(V_{a1}, V_{a3})	4.7530×10^8
(V_{a2}, V_{a3})	4.4922×10^8

desensitization magnitude. The wild variation in $|M_3|$ for the $V_a = 3\mu$V case is because both V_a and V_b are too small for the desensitization to be accurately observed. The larger inputs yield a much more stable value for $|M_3|$.

Lastly, an example of inputs being too large is shown in Table 5.6. Attempting to extract the compression term of the low-Q 1filt with inputs of $(V_{a1}, V_{a2}, V_{a3}) = (0.1\text{mV}, 0.2\text{mV}, 0.3\text{mV})$ yields the steadily decreasing values shown in the Table. We know the true value to be around 5.73×10^8 from Table 5.3; the problem in Table 5.6 is that M_5 is starting to skew the calculated M_3 values.

Overall Results

Being careful to avoid the errors mentioned in the previous subsection, the distortion terms for each of the three filters were extracted. The results are summarized in Table 5.7. A few comments are in order.

1. The f_a linear gain values are the ones extracted from simulations rather than those from the ac analyses. The values disagree slightly because the ac analyses predicted the center frequencies slightly inaccurately; further simulations reveal the high-Q 1filt center frequency to be at 1.8311GHz rather than 1.8315GHz, for example. The ac analyses were close enough, however.

2. All the distortion terms have much larger magnitudes that those in Table 5.1 because, from §4.6.2, the A_0 values are all so much higher.

3. The general trends observed in the distortion term magnitudes in Table 5.1 exist here as well: in all three configurations the compression term is the largest, followed by the desensitization and intermodulation terms except for 3filt where the intermodulation term is larger than the desensitization term.

4. The overall 3filt improvement is worse than that predicted in Table 5.1. This is probably due to the combination of smaller value of CS% (1% and not 2%) and A_0 (almost 40dB instead of 0dB). More discussion on this point follows in §5.6.

5. The compression term for 3filt is slightly worse than that of the low-Q 1filt. This is due to imperfect cancellation of the two terms in equation (5.48) as explained in §5.3.2. 3filt does show a marked improvement over the rather high compression value for the high-Q 1filt.

6. The desensitization terms for 3filt are quite a bit better than both 1filts just as in Table 5.1. The intermodulation term does *not* show as strong an improvement, but it shows an improvement nonetheless.

Table 5.7: Results of realistic filter simulation.

Term	3filt value	Low-Q 1filt value	3filt win (dB)	High-Q 1filt value	3filt win (dB)
f_a linear gain	8.42×10^1	7.85×10^1	+0.6	64.6×10^1	+2.3
f_b linear gain	4.50×10^0	3.91×10^1	+18.8	4.45×10^0	−0.1
f_c linear gain	2.31×10^0	2.18×10^1	+19.5	2.25×10^0	−0.2
f_a compression	1.09×10^9	5.74×10^8	−5.6	2.56×10^{11}	+47.4
f_a, f_b desensitization	2.45×10^7	1.39×10^8	+15.0	1.23×10^9	+34.0
f_a, f_c desensitization	6.21×10^6	4.14×10^7	+16.5	2.73×10^8	+32.9
f_b, f_c intermodulation	2.21×10^7	3.74×10^7	+4.6	4.26×10^7	+5.7

7. In §4.1.2 we stated it was acceptable to ignore dc offset or square terms in the nonlinearity because it does not affect the distortion terms we are interested in. This is true for this circuit too: even-power nonlinearities are definitely present as can be seen from the presence of a second harmonic in Figure 5.12. But this does not affect our extracted values: with or without even-power nonlinearities, these particular distortion terms do not change.

8. The extraction of the Volterra distortion terms for this circuit was not undertaken with any particular application in mind. That is, the author cannot think of a practical application for the particular 3filt simulated here. The point was to prove that the extraction method works even when we know nothing about the nonlinearities in the circuit — we can still extract distortion coefficients from simulations.

5.6 General Trends

We shall finish this chapter by looking at some general distortion trends for 3filt compared to 1filt. All along we have been calculating the improvement in dB that 3filt affords over 1filt, and we shall continue in the same vein. We will be comparing the three filters in Figure 5.3.

5.6.1 Improvement Surface

We have been looking at the ratio of $\dfrac{H_{A3}(f_1, f_2, f_3)}{M_{A3}(f_1, f_2, f_3)}$ at fairly specific points. For example, in Table 5.1 we calculate this ratio in four places: the compression point, the two desensitization points, and the intermodulation point. We see that at these four points the 3filt distortion is better than both 1filts. Does the same hold true over a larger range of (f_1, f_2, f_3) triples?

As an example, a plot of $\dfrac{M_{A3}(1, f_1, -f_2)}{H_{A3}(1, f_1, -f_2)}$ in dB as a function of both f_1 and f_2 is displayed in Figure 5.13. The comparison is between 3filt and the high-Q 1filt

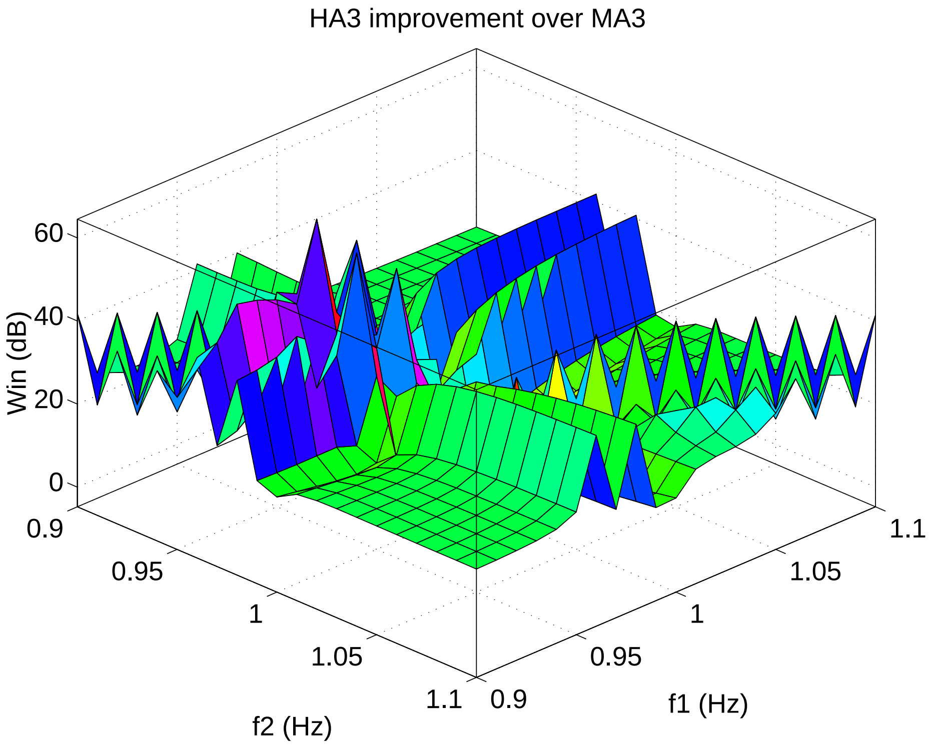


Figure 5.13: Improvement surface for $H_{A3}(1, f_1, -f_2)$ versus $M_{A3}(1, f_1, -f_2)$.

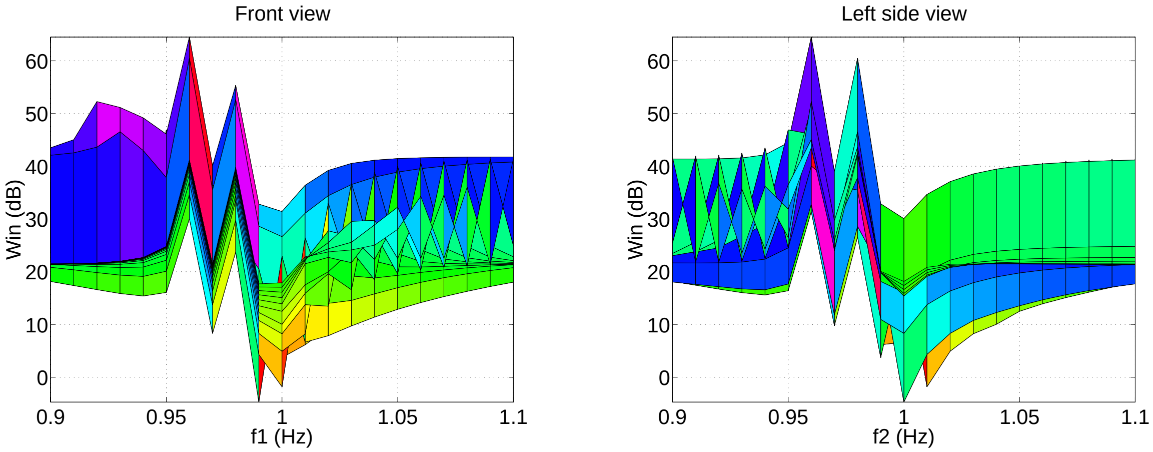


Figure 5.14: Front and left side views of above surface.

in Figure 5.3, and f_1 and f_2 span ranges around $f_0 = 1$Hz. The surface is not very smooth and rather difficult to visualize, but its features can be summarized as follows.

1. The flat parts of the surface are at 20dB. In most places around f_0, then, H_{A3} is an order of magnitude smaller than M_{A3}.

2. The horizontal ridges in the surface occur at f_{0B} and f_{0C}; along these lines, H_{A3} is very much smaller than M_{A3}.

3. The spikes along the diagonal occur for $f_1 = -f_2$. This means that the desensitization at the center frequency $H_{A3}(1, f_1, -f_1)$ is better for 3filt for *any* value of f_1. This quite nice result follows from equations (5.43), (5.44), and (5.45) — even if the interferers are not at the notches, the desensitization is better.

4. There is a small region where H_{A3} is larger than M_{A3}, i.e., where 3filt distortion is worse than 1filt, and it occurs close to the compression point $(1, 1, -1)$. It is visible in Figure 5.14 which shows the front and side view of the surface. (Regrettably, these views do not help in visualizing Figure 5.13 any better, but they *do* show the positive and negative peaks well.) This is due to imperfect vector cancellation in (5.48).

5.6.2 Varying CS%

An interesting exercise would be to vary the channel separation and see how the distortion terms in 3filt compare. Such a comparison is done in Figure 5.15. The left graph compares 3filt to the high-Q 1filt, the right graph 3filt to the low-Q 1filt. Naturally, the center frequencies of the 3filt interferer filters f_{0B} and f_{0C} are adjusted simultaneously with CS% so the notches always coincide with the interferers. The x-axis is CS% on a logarithmic scale, and the y-axis is the 3filt improvement in dB.

Predictably, 3filt distortion is best when CS% is large. Overall, the high-Q 1filt performs about 20dB worse than the low-Q one. The observed trends can be explained by considering the general formula for H_{A3}, equations (5.39) through (5.42).

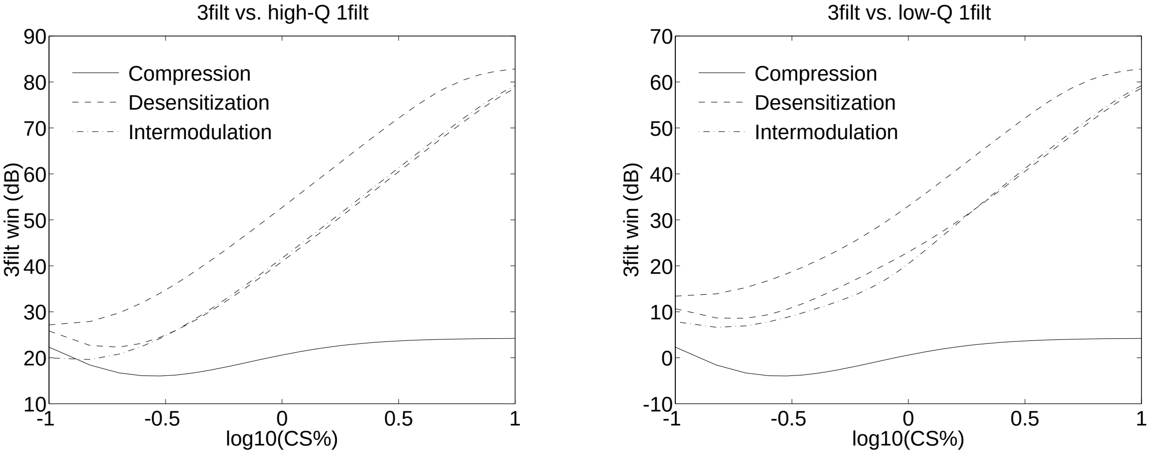


Figure 5.15: Distortion in 3filt versus two 1filts as a function of CS%.

1. For CS% more than 1%, the 3filt compression improvement remains fairly constant. This is because M_{A3} in (5.42) dominates. As CS% drops, M_{B3} and M_{C3} in that same equation start to contribute more and more, raising G_3 and hence H_{A3}.

2. The desensitization and intermodulation terms get worse in H_{A3} $40\frac{\text{dB}}{\text{dec}}$ faster than they do in M_{A3}. The decrease can be explained by noting that as the (f_1, f_2, f_3) arguments in the (M_{A3}, M_{B3}, M_{C3}) terms in (5.42) get closer together, these terms will increase — just as they did in §4.6.4. In 1filt there is only one M_3 term, but in 3filt there are *three* M_3 terms in (5.42), all increasing simultaneously.

3. The worsening H_{A3} win seems to level off for very small channel separations, but this is an artifact of what happens in the linear equation. The 3filt notches start to meld into one another when the channels are very close as can be seen in Figure 5.16. The nicely-defined linear transfer function for CS% $= 2\%$ degenerates into the 1filt-esque curve of CS% $= 0.1\%$; the gain at f_{0A} starts to fall away from $\frac{1}{k}$, and hence H_{A3} stops falling as quickly relative to M_{A3}. Of course, by this point the 3filt is really no longer a 3filt, so making CS% very

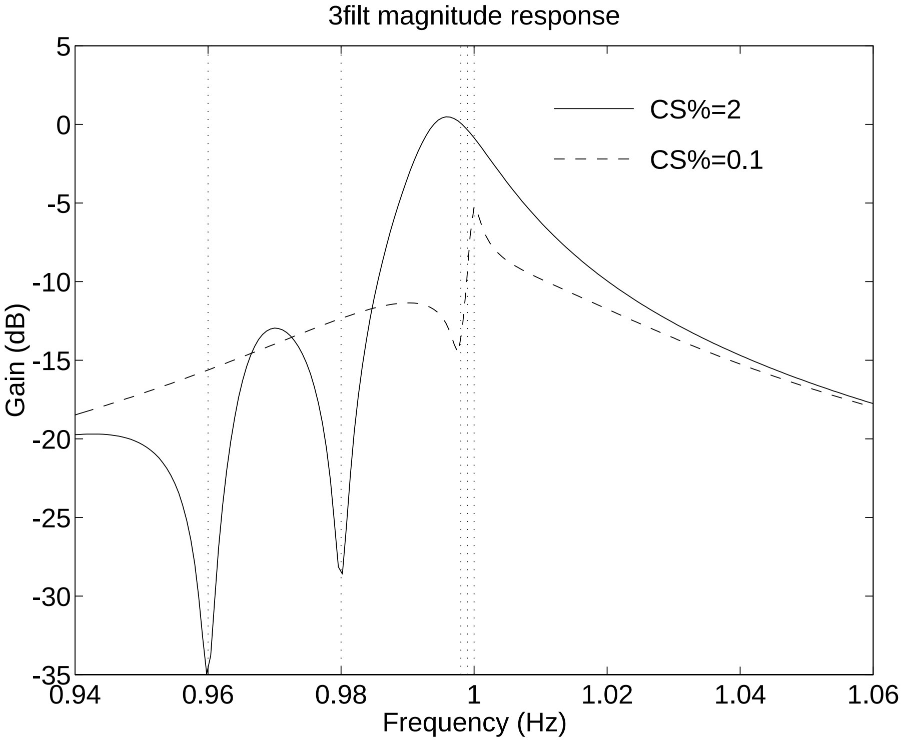


Figure 5.16: Degeneration of 3filt transfer function for small CS%.

small is rather pointless.

The reader can verify that the values in Table 5.1 appear on the graphs in Figure 5.15. The Table had CS% $= 2\%$ which corresponds on the graph to $\log_{10} 2 = 0.3$ on the x-axis.

5.6.3 Varying Gain at f_a

What happens to distortion in 3filt and 1filt when the transfer functions in Figure 5.3 are shifted vertically? To find out, A_0 is varied in 1filt and A_0 and k are varied simultaneously in 3filt. (Recall from §5.2.2 that raising the A_0 of each filter in 3filt doesn't alter the gain at f_{0A}: this gain is set by $\dfrac{1}{k}$ as long as equation (5.10) is satisfied. To keep the same shape of linear transfer function for 3filt as in Figure 5.3 requires

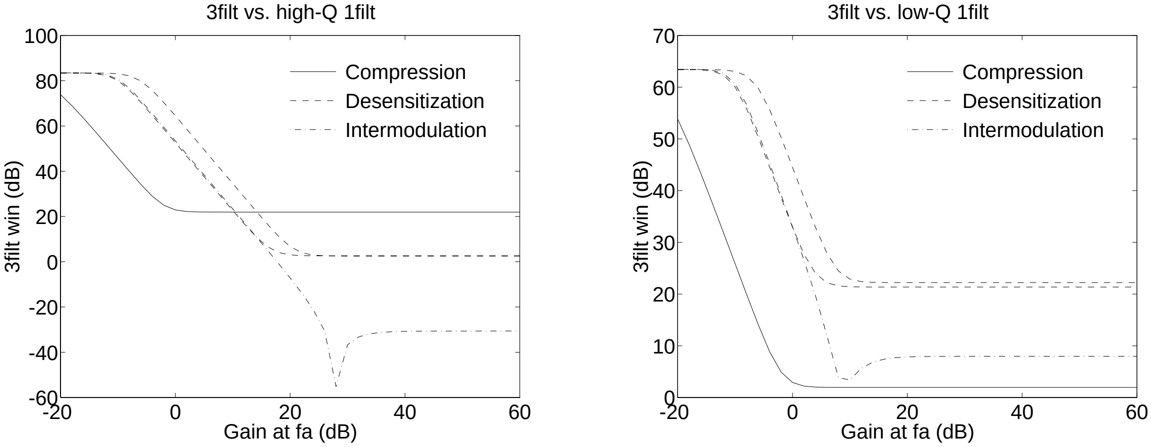


Figure 5.17: Distortion in 3filt versus two 1filts as a function of A_0.

changing A_0 and k in opposite directions and keeping their product constant.)

Figure 5.17 compares 3filt to the two different 1filts. Once again, the results from Table 5.1, which were for $A_0 = 0\text{dB}$, appear on the graphs. The results are easiest to explain by referring to §4.6.2. There we observed that for low peak gains 1filt has constant distortion which eventually begins to rise by $60\frac{\text{dB}}{\text{dec}}$. 3filt, being made up of 1filts, exhibits the same behavior — except the $60\frac{\text{dB}}{\text{dec}}$ rise begins for a *lower* peak gain. This explains the shape of the curves: at very low A_0, both 1filt and 3filt have constant distortion as a function of A_0, and 3filt has much less distortion due to the feedback, etc., discussed in §5.3.2. As A_0 rises, 3filt distortion starts to rise by $60\frac{\text{dB}}{\text{dec}}$ while that of 1filt stays constant, the net effect of which is to make 3filt's advantage fall by $60\frac{\text{dB}}{\text{dec}}$. For high enough A_0 the two 1filts' distortion starts to rise at $60\frac{\text{dB}}{\text{dec}}$, the same rate as 3filt, so the lines in Figure 5.17 level off.

It is interesting that for any amount of gain, 3filt always beats the low-Q 1filt. The intermodulation distortion in 3filt, however, is significantly worse than that of the high-Q 1filt for high gains.

A Note on the Realistic 3filt

The 3filt advantage over 1filt in Table 5.7 for the realistic circuit of §5.5.2 is not nearly as great as might have been hoped from Table 5.1, particularly in terms of the intermodulation performance. Thanks to §5.6.2 and §5.6.3 we now see two possible reasons why: the channel separation and the gain at f_a in §5.5.2 were quite a bit higher than in §5.3.2. We may reasonably conclude that the realistic 3filt was misdesigned.

We cannot say much more in the absence of specific information on the nonlinearities in the realistic filter. If we had more time, and if we really wished to design a good 3filt circuit using the realistic filter, we could plot graphs similar to Figure 5.15 and Figure 5.17 for such a circuit. This would require a staggering amount of computing time, however.

5.7 Summary of 3filt Properties

To summarize, three parallel filters with feedback can have less distortion than a single filter. §5.3 contains a Volterra series analysis of this phenomenon, and §5.6 indicates that three filters outperform a single filter as long as the center frequency gains are not too high nor the channels too closely spaced.

A final very brief discussion on applications: for AMPS, it is unlikely that a 3filt could operate at the front end of a cellular phone handset. 30kHz channels at a 900MHz RF corresponds to CS% = 0.003% — far too close. Application to AMPS IF or to commercial FM radio front-end, where 800kHz channel separation at a 100MHz RF corresponds to CS% = 0.8%, seems much more viable.

Chapter 6

Measurements in the Laboratory

6.1 Introduction

Our last task is to measure the distortion terms for an actual circuit in the laboratory. As stated in §5.4 attempts in the literature at measuring Volterra kernels invariably involve a spectrum analyzer, but we have seen the difficulties inherent in this for measuring distortion in §5.4.1. We will almost certainly need a network analyzer so we can use both the magnitude and phase information thereby obtained in the formulae in §5.4.2.

6.1.1 Circuit

A readily-available circuit was the one described in [Shov93]. A biquad filter was designed using tunable TAs in a 0.8μm BiCMOS process. The circuit diagram is shown in Figure 6.1. The filter has a differential input and differential low pass and band pass outputs. The TAs contain both bipolar and MOS transistors and their transconductances are voltage-tunable over about a decade. The linear band pass transfer function is

$$V_{BP}(s) = \frac{\frac{G_{mi}}{C}s - \frac{G_{m21}G_{mb}}{C^2}}{s^2 + \frac{G_{m22}}{C}s + \frac{G_{m12}G_{m21}}{C^2}} \qquad (6.1)$$

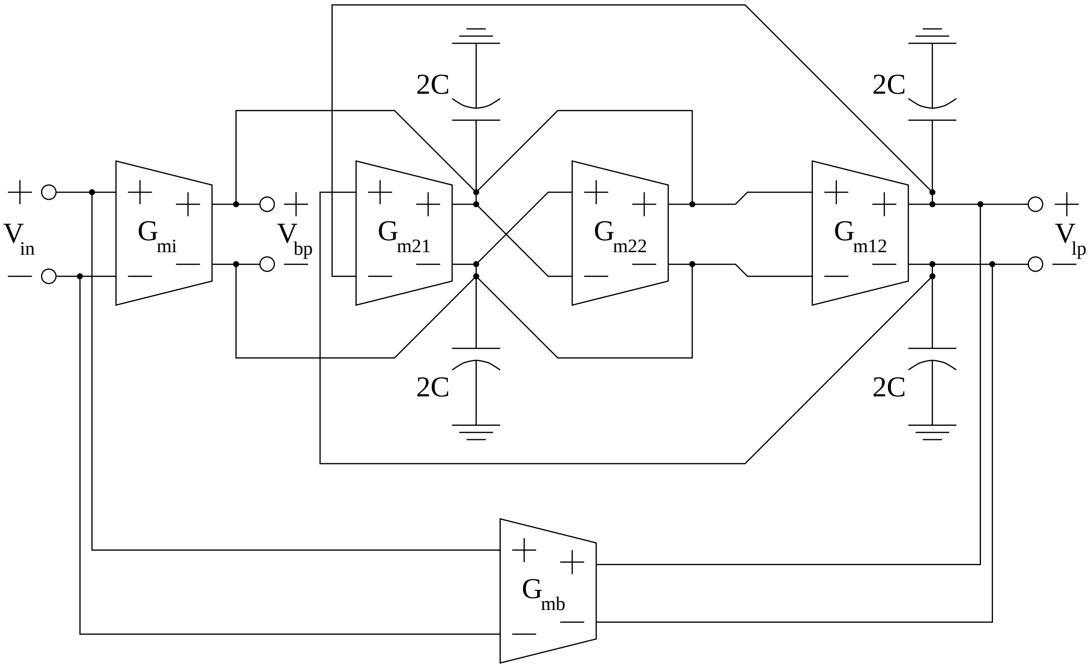


Figure 6.1: Biquad filter circuit.

The filter's center or corner frequency can be adjusted by varying G_{m12} and G_{m21} while filter Q is tuned with G_{m22}. Center frequency gain or dc gain A_0 can be adjusted with G_{mi}.

The circuit was manufactured and packaged and a chip mounted on a printed circuit board is used for this thesis. Power supplies are set to $V_{dd} = 2.5$V and $V_{ss} = -2.5$V; the individual TA g_m tuning is accomplished with simple potentiometer resistive dividers. Band pass filters with center frequencies from about 15MHz to 250MHz are quite easy to construct. Stable filter Q of 200 is also possible to achieve although the filter is verging on oscillating for so high a Q. Q can also be made negative turning the filter into an oscillator which, it turns out, injection locks beautifully.

6.1.2 Measurement Goals

Throughout this thesis we have been investigating a band pass filter structure with a biquadratic linear transfer function. Although Figure 6.1 is a different circuit from Figure 4.1, its linear transfer function (6.1) is almost equivalent to equation (4.2), the transfer function of the filter we have been studying. Furthermore, the fact that this circuit can be made to injection lock confirms something we already knew: it contains nonlinearities. The input of each TA contains a MOS differential pair, and as stated in §4.1.2 therefore has an i-v characteristic with odd symmetry. Since (6.1) is a biquadratic form and since the active devices in Figure 6.1 contain cubic-type nonlinearities, it seems plausible that this filter might exhibit the same properties as the filter structure we have been investigating. In particular, the general trends from §4.6 might hold.

Our goal in these measurements should then be twofold: first, to show that the measurement technique in §5.4.2 can be applied to an actual circuit, and second, to see if the distortion term trends in §4.6 hold for this implementation of a biquadratic band pass filter.

6.1.3 Test Equipment and Setup

We will use the Hewlett-Packard 4195A network analyzer, which is capable of measurements between 10Hz and 500MHz, in S-parameter mode. Two S-parameter measurement attachments are required, so the Hewlett-Packard 35676-66301 50Ω signal divider operable to 200MHz is well-suited to the task.

When in S_{21} mode, the 4195A generates a tone of a particular amplitude and frequency at transmit port one and measures the amplitude and phase of the tone returned at receive port two. It can be configured to either sweep the tone's frequency or to emit a tone at a constant frequency. It is this latter configuration we will be using to generate f_a, the desired tone. To add an interfering tone at f_b we will use the Rohde and Schwarz SMHU signal generator and a simple resistive power splitter

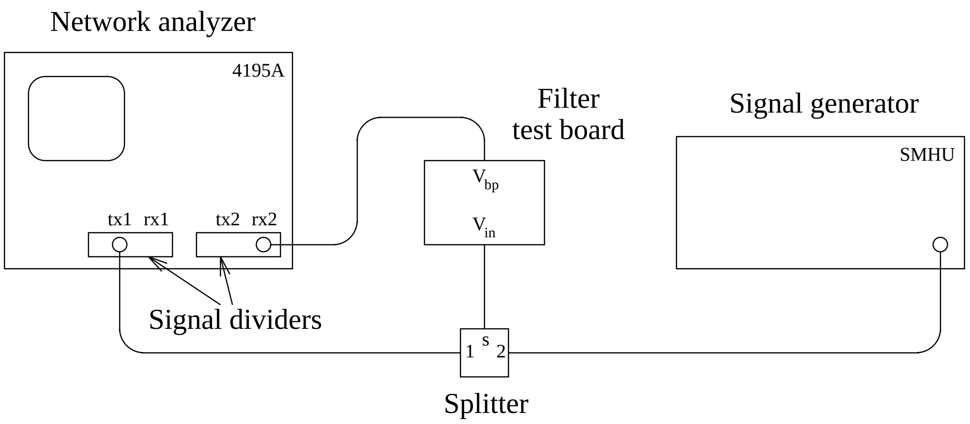


Figure 6.2: Test equipment setup.

to sum f_a and f_b.

Finally, the power supply we will use is the Hewlett-Packard E3630A triple-output power supply that can generate positive and negative voltages simultaneously, and we will connect everything with fairly short lengths of semi-rigid coaxial cable. A diagram of the test setup is shown in Figure 6.2.

Although the filter is differential, single-ended measurements mean fewer connections and less complexity, and since we are only interested in whether the measurements will work we will perform single-ended measurements.

6.2 Initial Measurements

6.2.1 Measuring Gain and Phase

We would like the network analyzer to measure and display the gain and phase characteristics of the filter. However, in between the network analyzer and filter are the signal dividers, splitter, signal generator, and coaxial cables with their own gain and phase characteristics. For this reason a calibration step is performed before taking any filter measurements. First, the measurement frequency range is entered into the analyzer; then, the entire circuit *except* for the filter is connected, i.e., the cables for the filter input and output are connected directly to one another. The gain and phase of *this* setup are measured. Once the measurement is complete the analyzer normalizes the measured gain and phase at each frequency to zero — removing the effect of all the components other than the filter. The filter can now be reattached and characterized without the rest of the components interfering.

To start we will set the filter for a Q of about 100 and a center frequency f_0 close to 100MHz, which are nice numbers for broadcast FM. A typical set of band pass gain and phase characteristics is shown in Figure 6.3. The analyzer's output signal amplitude is -46.0dBm (about 1.58mV) and its resolution bandwidth is RBW = 100Hz. We have achieved $f_0 = 99.0$MHz, $A_0 = 2.20$dB, and a -3dB-bandwidth of about 750kHz so we can calculate

$$Q = \frac{f_0}{-3\text{dB bandwidth}} = \frac{99.0\text{MHz}}{750\text{kHz}} \approx 130$$

Two items are worthy of mention here.

1. The gain characteristic is nice and symmetrical in this small frequency range but the center frequency gain A_0 is quite a bit lower than simulated [Shov93]. This is because the on-chip output driver is a small transistor which cannot drive a 50Ω-load very well.

2. The phase characteristic looks as it should except for a vertical shift of about -30^o. This is because of transmission-line effects due to the coaxial cables; calibrating the network analyzer does not remove the phase shift because simply bending the cables is enough to change the phase. This does not have serious repercussions for our measurements, however.

The effects of gain compression are quite striking and can be observed by simply increasing the amplitude of the input signal. A graph of the gain with input levels of -36dBm, -26dBm, and -16dBm is shown in Figure 6.4. Even for an input level of

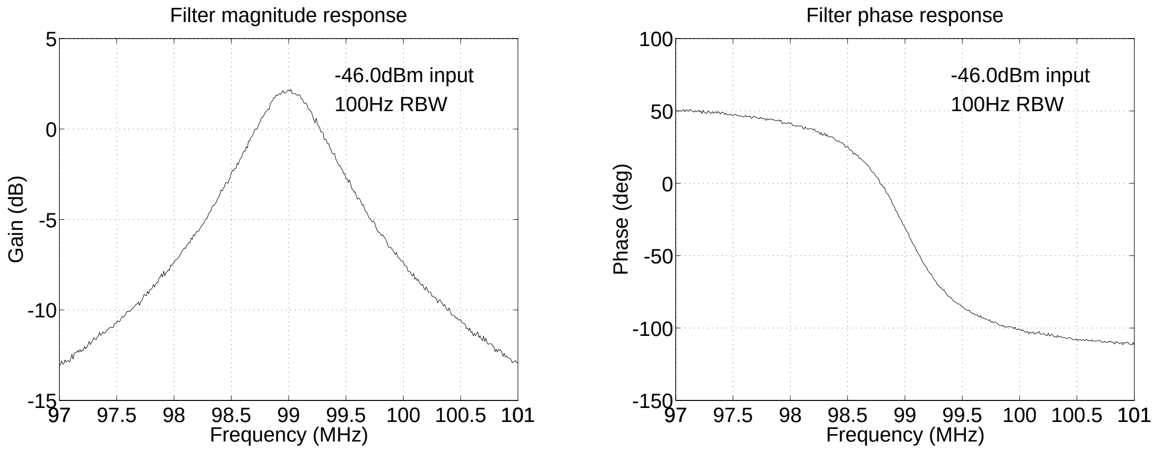


Figure 6.3: Measured gain and phase characteristics.

$-36\text{dBm} = 5.0\text{mV}$ the center frequency gain has fallen slightly from Figure 6.3. The reduction becomes very severe — almost 10dB — for the largest input level.

Most satisfying of all is the slight leftward asymmetry that begins to creep into the curve for large inputs. In the section on the Duffing equation, §4.4.3, we showed in Figure 4.19 that a negative cubic nonlinearity (an "S-shaped" curve) bends the gain to the left while a positive cubic nonlinearity (an "N-shaped" curve) bends the gain to the right. The TA i-v characteristic for this filter is S-shaped, so the observed leftward-bending should come as no surprise.

6.2.2 Difficulties with Extracting Distortion Terms

Let us now attempt to apply the extraction method proposed in §5.4.2 to this particular filter. Several factors make its practical application rather more difficult than its application to a numerical simulation. We will examine some of the problems and how to minimize them in this section.

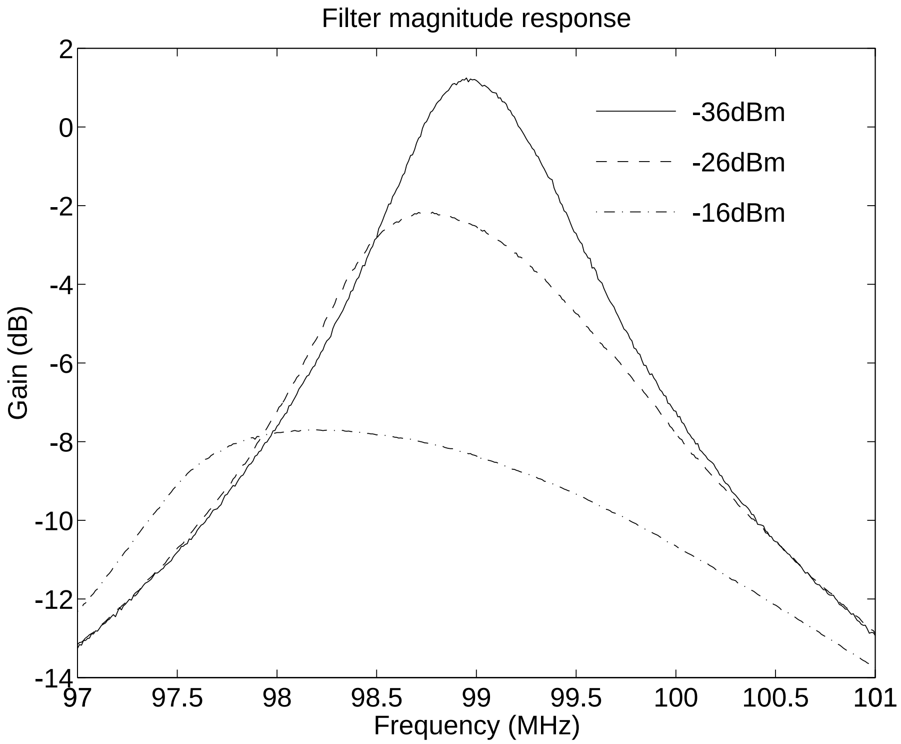


Figure 6.4: Compression in measured gain characteristic.

Noise

Measurement noise is probably the single worst problem. Between one measurement and the next the gain and phase at a particular frequency move up and down slightly due to thermal noise and random fluctuations. As an example, here are the measured gain and phase values when four measurements in a row at the same frequency are performed.

$$\begin{array}{rcccc} \text{Gain (dB):} & \text{-10.8711} & \text{-10.8102} & \text{-10.8201} & \text{-10.8465} \\ \text{Phase (deg):} & \text{-82.0846} & \text{-81.9801} & \text{-81.5393} & \text{-81.9344} \end{array} \tag{6.2}$$

True, the gain values agree to three decimal places and the phase changes by less than 1^o, but in our work in §5.5 our simulations had over ten digits of accuracy. If our extraction technique requires many decimal places we could be in trouble. The change in phase from one measurement to the next tends to be worse near high group-delay (i.e., high phase slope) regions like f_0.

There are two ways we can reduce measurement noise. The first is to set the network analyzer resolution bandwidth to a low value; this improves accuracy at a cost of slower measurements. The second is to take the average of multiple measurements.

It is not clear that taking the average of a set of gain and phase values is very meaningful: perhaps we should be converting from polar to rectangular coordinates and averaging (x, y) values instead. When the gains and phases change by as little as those in (6.2) it does not matter much which we do:

$$\begin{aligned}\text{Average in polar coordinates} &= -10.8370\angle - 81.8846^o \\ \text{Average in rectangular coordinates} &= -10.8369\angle - 81.8848^o\end{aligned}$$

In any case, averaging multiple measurements *will* assist somewhat but not as much as might be hoped because of the next problem.

Drift

This particular filter has quite a tendency to drift, particularly near f_0. A good illustration of this behavior can be seen in the following data: twenty gain and phase measurements at three different input amplitudes were taken and averaged, and this was repeated three times in succession for a 98MHz input frequency. The results are shown in Table 6.1. Notice that for increasing input amplitude the gain and phase decrease monotonically in all three measurements, but at the start of each new measurement f_0 and/or A_0 drifted slightly so that the absolute gain and phase values differ.

The obvious way to overcome drift is to take all measurements quickly. But this trades off with the averaging to reduce measurement noise: doing more measurements in hopes of being able to average out noise takes longer, by which time the filter has drifted a little. There will exist an optimum number of measurements that is large

Table 6.1: An illustration of filter drift.

Measurement number	Gain and phase		
	-36.0dBm input	-34.0dBm input	-32.0dBm input
1	$-1.205 \quad \angle -144.09^o$	$-1.287 \quad \angle -146.72^o$	$-1.481 \quad \angle -151.00^o$
2	$-1.157 \quad \angle -147.48^o$	$-1.328 \quad \angle -149.22^o$	$-1.598 \quad \angle -151.71^o$
3	$-1.277 \quad \angle -141.79^o$	$-1.380 \quad \angle -144.23^o$	$-1.599 \quad \angle -147.09^o$

enough to keep measurement noise low yet small enough to prevent drift from being too severe.

Component Loss

To apply the extraction method of §5.4.2 we need to know the exact amplitude of our input signals. In the laboratory we must keep in mind that both the signal dividers and the splitter have some loss in them. Fortunately this loss can be measured directly: we connect the network analyzer through the divider and splitter to a spectrum analyzer, make the network analyzer emit a single tone at a fixed frequency, and measure the amplitude of the tone on the spectrum analyzer.

In this setup we find that a setting of xdBm in the network analyzer has an amplitude of about $x - 13.5$dBm on the spectrum analyzer. This is logical: the nominal insertion loss in the signal divider is listed as 10dBm, and the loss in the splitter will be slightly over 3dBm. In our calculations, therefore, we must remember that the tone from the network analyzer is about 13dBm lower once it reaches the filter. The interfering tone experiences some loss as well, due only to the splitter. This loss can be measured to be about 3dBm.

Signal Amplitude

In our extractions on the circuit in §5.5.2 we observed that input signal amplitudes could be neither too large nor too small: too large and M_5 becomes significant, too small and M_3 becomes buried in FFT noise. The same results hold true for practical applications. Signals that are too small have an additional problem in practice: the network analyzer has difficulty measuring small received signals accurately which will increase measurement noise.

There is no way to determine what signal amplitude is best except trial and error.

6.3 Some Measurement Results

The following sections apply to measurements taken on the filter with the transfer function depicted in Figure 6.3.

6.3.1 Measurement Procedure

The 4195A network analyzer has a BASIC interpreter built into it. The immediately obvious advantage is that we can automate any averaging we do — we can write a BASIC program to repeatedly measure gain and phase at one frequency, then calculate the average and display the results.

But we can take it quite a lot further than that. The BASIC is rudimentary but has enough features to make it powerful: subroutines, arrays, and the full line of mathematical functions like square root and arctangent. To extract the compression term, for example, we could automate the entire extraction procedure in a manner like this:

- Set the network analyzer frequency to band center.
- Set the network analyzer output amplitude to a particular value, do one or more gain and phase measurements, and remember the average values.

- Repeat the previous step for one or more amplitudes.
- Taking into account component loss, apply equations (5.60) and (5.61) to each pair of average gain and phase values, and display the results.

The advantages of writing a program to do this are numerous:

1. Automation means speed: we can do many measurements quickly which will help alleviate drift problems.
2. The chance of a calculation error is greatly reduced.
3. We can play with input amplitudes, number of measurements to average, or any other parameter and rerun the program. This gives us a way to determine good input conditions with a minimum of effort.

6.3.2 Initial Results

At the approximate center frequency $f = 99$MHz, let us determine what size inputs are suitable for measuring the linear gain $M_1(f)$ and the compression $M_3(f, f, -f)$. Table 6.2 shows the results for a single measurement at various pairs of input amplitudes. Recalling that Volterra series values are complex numbers $x + jy$, we may write the amplitude as $20 \log(x^2 + y^2)$ in dB and the phase as $\arctan \frac{y}{x}$; this has been done in the Table.[1]

The compression term phase jumps around a fair amount between measurements for the $(-46, -40)$ amplitude pair because these input signals are too small: the compression phase becomes much more constant for larger input pairs. For the largest input pair $(-34, -28)$, the compression gain is around 109dB and not 114dB as it is for the other pairs. This is because M_5 is starting to become significant, and we can demonstrate this as follows.

[1] It might seem strange to record M_3 in dB when it is not really a "gain" in the normal sense. The only purpose in doing so is to make it logarithmic.

Table 6.2: Initial compression measurement results.

Input amplitudes (dBm)		Meas. 1	Meas. 2	Meas. 3	Meas. 4
$(-46, -40)$	M_1 (dB)	0.68	0.89	0.89	0.52
	M_1 (^{o})	-13.3	-14.7	-14.7	-15.0
	M_3 (dB)	116.2	116.8	117.4	118.0
	M_3 (^{o})	-157.6	-168.2	138.9	-112.9
$(-40, -34)$	M_1 (dB)	0.59	0.67	0.54	0.71
	M_1 (^{o})	-19.8	-18.8	-20.3	-21.3
	M_3 (dB)	114.2	115.8	113.2	115.2
	M_3 (^{o})	-138.6	-143.2	-148.2	-148.1
$(-34, -28)$	M_1 (dB)	0.43	0.54	0.61	0.73
	M_1 (^{o})	-21.5	-21.6	-21.7	-21.7
	M_3 (dB)	109.1	109.2	109.5	109.7
	M_3 (^{o})	-157.5	-158.5	-161.1	-161.8
$(-43, -33)$	M_1 (dB)	0.72	0.72	0.53	0.59
	M_1 (^{o})	-17.0	-20.0	-21.8	-22.3
	M_3 (dB)	114.5	114.2	113.8	112.5
	M_3 (^{o})	-145.1	-148.5	-153.3	-150.5

First, it is not difficult to derive a relationship between signal amplitude in dBm and signal amplitude in volts. The formulae are

$$\begin{aligned} V_x &= \sqrt{0.1 \times 10^{\frac{y}{10}}}, & y \text{ in dBm} \\ y &= 10 + 20\log_{10} V_x, & V_x \text{ in V} \end{aligned} \tag{6.3}$$

Second, if we know $M_1 = A_1\angle\theta_1$ and $M_3 = A_3\angle\theta_3$ where A_1 and A_3 are in dB, and the input amplitude V_x in volts, we can find the amplitudes of the first- and third-order terms at the output using (4.19) and (4.20) as follows:

$$\begin{aligned} \text{First order:} \quad V_1 &= V_x 10^{\frac{A_1}{20}}\angle\theta_1 \\ \text{Third order:} \quad V_3 &= \tfrac{V_x^3}{8} 10^{\frac{A_3}{20}}\angle\theta_3 \end{aligned} \tag{6.4}$$

Now: when the network analyzer emits a −28dBm tone, we know that it will lose about 13dBm from the signal divider and splitter. A −41dBm tone has an amplitude $V_x = 2.82\text{mV}$ from (6.3); let us use this to calculate the output component amplitudes using (6.4) assuming $A_1 = 0.5\text{dB}$ and $A_3 = 115\text{dB}$.

$$\begin{aligned} |V_1| &= V_x 10^{\frac{0.5}{20}} &= 2.99\text{mV} \\ |V_3| &= \tfrac{V_x^3}{8} 10^{\frac{115}{20}} &= 1.57\text{mV} \\ \text{Ratio } \tfrac{|V_3|}{|V_1|} &= 53\% \end{aligned}$$

So the third-order term is half as large as the first-order term. A good rule of thumb to find the fifth-order term amplitude $|V_5|$ is to assume that $\frac{|V_5|}{|V_3|}$ and $\frac{|V_3|}{|V_1|}$ are approximately equal; this means $\frac{|V_5|}{|V_1|} = 27\%$, which is very large indeed. Performing the same series of calculations with a −34dBm tone from the network analyzer reveals $\frac{|V_3|}{|V_1|} = 13\%$ and $\frac{|V_5|}{|V_1|} = 1.7\%$. Therefore, a −34dBm tone is large enough to give fairly consistent M_1 and M_3 values while still keeping M_5 small.

The results for $(-40, -34)$ and $(-43, -33)$ agree fairly well: both find that $M_1 \approx 0.6\text{dB}\angle{-20^o}$ and $M_3 \approx 114\text{dB}\angle{-148^o}$. As long as we choose our input amplitudes in the right range, the results seem reasonable.

6.3.3 Compression Over a Range of Frequencies

It would be interesting to measure $M_3(f,f,-f)$ over a whole range of frequencies. This is not something we looked at in §4.6 but our test setup can be easily adapted to perform such measurements. Therefore let us briefly examine how $M_3(f,f,-f)$ depends on $M_1(f)$.

Consider the denominator of M_3, equation (4.17), when $(f_1,f_2,f_3)=(f,f,-f)$, $f=2\pi\omega$:

$$C_1\sum_3 j\omega_i+\frac{1}{R_1}+\frac{g_{m1}g_{m2}}{C_2\sum_3 j\omega_i} \quad = \quad C_1 j\omega+\frac{1}{R_1}+\frac{g_{m1}g_{m2}}{C_2 j\omega} \tag{6.5}$$

Comparing this to M_1, equation (4.13), we note that (6.5) is equal to $\frac{g_{mi}}{M_1(f)}$. Substituting this in (4.17) gives

$$M_3(f,f,-f)=\frac{M_1(f)}{g_{mi}}\left\{6\epsilon_i-6M_1(f)M_1(f)M_1(-f)\left[\frac{\epsilon_2 g_{m1}}{C_2 j\omega}+\frac{\epsilon_1 g_{m2}^3}{C_2^3 j\omega^3}\right]\right\} \tag{6.6}$$

We see that $M_3(f,f,-f)$ is proportional to the fourth power of $M_1(f)$. Is this borne out in measurements?

To facilitate taking measurements at many frequencies, a program was written in QBASIC on a PC to control the network analyzer over the Hewlett-Packard Instrument Bus (HPIB). It operates as follows:

1. Choose a low pair of input amplitudes.
2. Measure M_1 and M_3.
3. Using equations (6.3) and (6.4), estimate the input amplitude that will make $\frac{|V_3|}{|V_1|}=15\%$. This will ensure an accurate measurement while keeping M_5 small.
4. If the input amplitude is too far above or below the estimate, set it to the estimate and go to step two.
5. Tabulate the results, increase the frequency, and go back to step one.

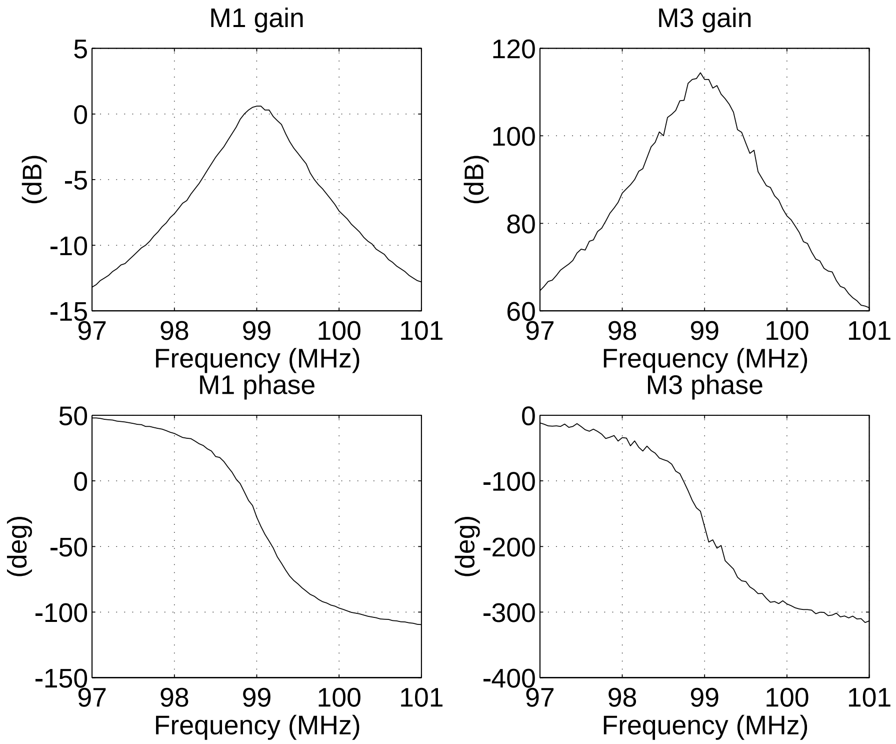


Figure 6.5: Graphs of measured linear and compression terms.

QBASIC has the ability to save textual data to disk, and is generally easier to use than the 4195A built-in BASIC, so it is ideal for the task. Figure 6.5 displays the results.

The linear gain characteristic, M_1, looks quite similar to Figure 6.3. Any differences can be attributed to the fact that the filter drifted a bit between Figure 6.3 and Figure 6.5. As well, the compression characteristic, M_3, has roughly the same shape as M_1 and from (6.6) we are expecting it to follow M_1 in the ratio 4:1. Does it? We see that M_1 rises from about -13dB to $+0.5$dB from 97MHz to 99MHz and that over this same range M_3 rises from about $+64$dB to $+115$dB. The ratio is then 51:13.5=3.78:1, quite close to the expected value. The ratio is even better for 99MHz to 101MHz, about 53:13=4.07:1.

6.3.4 Gain Compression vs. Gain Expansion

The phase graphs in Figure 6.5 have interesting features too. Returning to Figure 6.4 for a moment, we can see that for an input amplitude of −26dBm, gain *expansion* takes place between about 97MHz and 98.4MHz, while gain *compression* occurs between 98.4MHz and 101MHz. The relative phases of M_1 and M_3 tell us which should take place. If the M_1 and M_3 vectors are in the *same* direction (i.e., if $\theta_1 - \theta_3 = 0^o$), then both terms in the vector sum $V_x M_1 + \frac{V_x^3}{8} M_3$ are in the same direction, and so gain *expansion* will occur for suitably large input signals. Conversely, M_1 and M_3 being in *opposite* directions, $\theta_1 - \theta_3 = 180^o$, will effect gain *compression.*

More generally, if the phase difference is anywhere in the range $-90^o < \theta_1 - \theta_3 < 90^o$, adding the long M_1 vector to the short M_3 vector will produce a longer vector, which means gain expands. A phase difference in the other half plane means gain compresses. Glancing quickly at the phase graphs, we see that at 97MHz, $\theta_1 - \theta_3 \approx 50^o - (-15^o) = 65^o$, while at 101MHz, $\theta_1 - \theta_3 \approx -110^o - (-310^o) = 200^o$. These tell us we *expect* the expansion at 97MHz and the compression at 101MHz we observed in Figure 6.4.

We can do better than this, however: we should be able to *predict* the 1dB-compression point for any point on the gain plot if such a point exists. The radio engineer will no doubt be used to the concept of a 1dB-compression point for an *amplifier*, but what about for a filter? Assuming $M_1 = A_1 \angle \theta_1$ and $M_3 = A_3 \angle \theta_3$ to be the only two significant output components, we should be able to calculate the input amplitude V_{c1} at which the total gain $M_1 + M_3$ deviates from the linear gain M_1 by 1dB.

The calculation has been left for Appendix B where the result is derived and stated in equation (B.6), reproduced here for convenience.

$$V_{c1} = \sqrt{8 \times 10^{\frac{A_1 - A_3}{20}} \left[-\cos(\theta_1 - \theta_3) - \sqrt{10^{-0.1} - \sin^2(\theta_1 - \theta_3)} \right]} \qquad (6.7)$$

The Appendix also says that $+117^o < \theta_1 - \theta_3 < -117^o$ is required for a 1dB-

Table 6.3: Calculated vs. measured 1dB-compression points.

Freq.	A_1	A_3	$\theta_1 - \theta_3$	V_{c1} (dBm)		
(MHz)	(dB)	(dB)	(^{o})	predicted	measured	error
99.0	+0.6	+113.5	+135.4	−45.3	−44.5	−0.8
99.5	−2.9	+97.7	+175.8	−40.9	−41.5	+0.6
100.0	−7.2	+81.1	−169.9	−34.7	−34.5	−0.2
100.5	−10.3	+69.0	−159.6	−29.9	−30.0	+0.1
101.0	−12.7	+60.7	−156.8	−26.9	−26.5	−0.4

compression point to exist; the graphs show that frequencies 99MHz and beyond satisfy this condition.

Using the data from the graphs in equation (6.7), the predicted 1dB-compression point at various frequencies is compared with the measured value in Table 6.3. The measured values assume 13.5dB loss between the network analyzer and the filter input. The predictions and measurements match one another closely.

6.4 General Trends

Let us turn to the distortion term trends from §4.6. As stated in §6.1.2, it is logical to suppose that this filter, being biquadratic and having cubic nonlinearities, would exhibit similar behavior to the filter in Chapter 4.

Of the six variables from §4.6, center frequency f_0, filter Q, peak gain A_0, nonlinearity strength NL%, signal amplitude, and channel separation CS%, we can control all of them but NL%. Additionally, as stated in §4.6.5, signal amplitude is not a particularly interesting variable in that the dependence of distortion on signal amplitude is obvious. Furthermore, our extractions in §6.3 are based on finding M_1 and

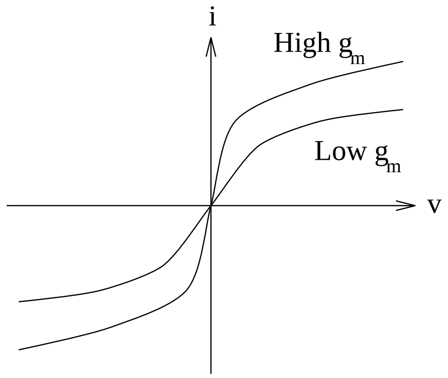


Figure 6.6: Transconductor i-v characteristic tuning.

M_3 alone rather than VM_1 or V^3M_3, so in this sense they are independent of signal amplitude.

Our task, then, will involve f_0, Q, A_0, and CS%: keeping three variables fixed, we will extract M_3 as a function of the fourth variable.

A Note on Nonlinearity Strength

In order to change f_0, Q, or A_0, we must alter the transconductance of G_{m21}, G_{m22}, or G_{mi}, respectively. This is accomplished by tuning a control voltage; a higher control voltage means a higher g_m, a lower control voltage means a lower g_m. The i-v characteristic becomes scaled *vertically* as shown in Figure 6.6 [Shov93]. The curve is S-shaped because the output current saturates for large input voltage swings,

Unfortunately there is no easy way to determine what effect g_m tuning will have on the nonlinearity strength. That is, suppose that for small inputs we may assume an i-v characteristic of the form $i = g_m v - \epsilon v^3$. Then there is no way to predict how tuning g_m will alter ϵ without writing a defining equation for ϵ in terms of transconductances, capacitances, etc. Such an effort is beyond the scope of this thesis, but we can get an idea of how the ratio $\text{NL\%} = \dfrac{0.01\epsilon}{g_m}$ from equation (4.46) varies. This is done in the following section.

6.4.1 Center Frequency

In §4.6 we stated that our results ought to be independent of f_0: if we keep all five other variables fixed, M_3 shouldn't change. This does *not* hold true for this circuit. The compression measurements for two circuits with $Q = 80$ and $A_0 \approx -8.0$dB are shown in Table 6.4.

Table 6.4: M_3 compression values for two different f_0.

f_0 (MHz)	A_0 (dB)	Q	$M_1(f_0)$ (dB)	$M_3(f_0, f_0, -f_0)$ (dB)
100.08	-7.96	80	$-7.9 \angle -18.4^o$	$+94.0 \angle -149.6^o$
150.03	-8.14	79	$-8.1 \angle -33.5^o$	$+74.5 \angle -154.8^o$

The center-frequency compression term $M_3(f_0, f_0, -f_0)$ is quite a bit higher for the $f_0 = 100$MHz case than for the $f_0 = 150$MHz case, even though Q and $M_1(f_0)$ are almost identical. The most reasonable explanation is that NL% is changing. In §4.6.3 we noted that a larger NL% resulted in more distortion. It would seem that *lowering* f_0, which is accomplished by *lowering* G_{m21}, results in a *higher* relative ϵ value.

In fact, to generate the results in the Table, changing G_{m21} to alter f_0 also alters the filter Q and A_0 due to finite output conductances of the TAs [Shov93] and other parasitic effects. Consequently the control voltages for all three of G_{m21}, G_{m22}, and G_{mi} had to be tweaked individually to arrive at two filter configurations with the same Q and A_0 but different f_0. The net result is that NL% gets altered by different amounts in each TA; overall, NL% seems to increase for the same filter with a lower f_0.

The important lesson to be gleaned is that we cannot make any of our results truly independent of NL%. The best we can do is try to alter control voltages as little as possible, which means taking measurements over fairly small ranges.

Table 6.5: Varying Q and A_0 simultaneously with $f_0 = 150\text{MHz}$.

Q	$20\log_{10} Q$	$A_0 = \lvert M_1(f_0)\rvert$ (dB)	$\lvert M_3(f_0, f_0, -f_0)\rvert$ (dB)
54	34.6	-11.2	62.4
69	36.8	-9.3	70.0
90	39.1	-7.0	78.5
129	42.2	-3.5	92.3
145	43.2	-3.0	94.0

A Brief Note Before Continuing

In the next couple of sections we will be extracting the compression term at f_0, $M_3(f_0, f_0, -f_0)$, first holding A_0 constant and then holding Q constant. Before we do let us observe what happens when both Q and A_0 are altered simultaneously. This is accomplished by tuning G_{m22} alone, and the results are shown in Table 6.5 for a filter with $f_0 = 150\text{MHz}$.

Comparing A_0 to $|M_3|$ we see a 4:1 ratio in their slopes: A_0 rises by 8.2dB while $|M_3|$ rises by 31.6dB. This can be explained by comparing the rise in $|M_3|$ to the rise in *both* Q and A_0: §4.6.1 says the compression $|M_3|$ should rise with Q in the ratio 1:1 (the compression line in Figure 4.28) while §4.6.2 says compression $|M_3|$ should rise with A_0 in the ratio 3:1 (the compression line in Figure 4.29). Is this what happens in the Table? Yes — Q rises by 8.6dB, A_0 rises by 8.2dB, and $|M_3|$ rises by 31.6dB, which is quite close to $8.6 + 3 \times 8.2$.

This implies that "normalization" in the following sense is valid: if, when we are trying to vary Q and hold A_0 constant, there are slight variations in A_0, they can be subtracted out of the measured M_3 value in the ratio 3:1. Likewise, if, when we are trying to vary A_0 and hold Q constant, there are slight variations in Q, they can be subtracted out of M_3 in the ratio 1:1. The measured M_3 values in the following two

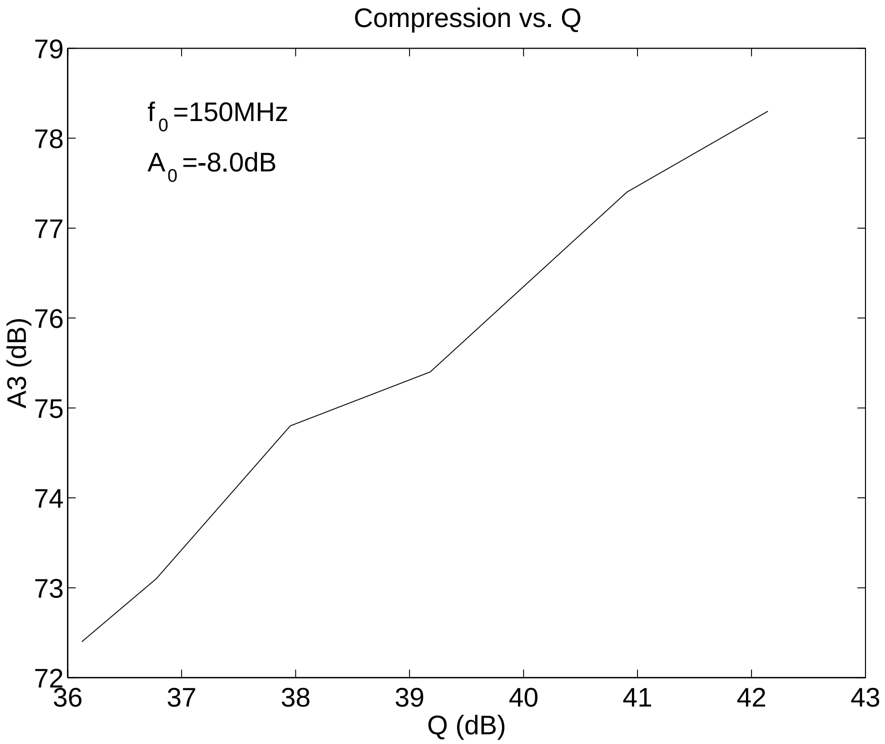


Figure 6.7: $|M_3(f_0, f_0, -f_0)|$ as a function of filter Q.

sections have been normalized in this manner.

6.4.2 Filter Q

A filter at $f_0 = 150$MHz with normalized $A_0 = -8.0$dB was characterized over a range of Q values from about 65 to 130. The measured $|M_3(f_0, f_0, -f_0)|$ values in dB are plotted versus Q in dB in Figure 6.7. (The author was able to keep the actual A_0 between -8.2 and -7.8, so the largest amount of normalization required was $(8.0 - 7.8) \times 3 = 0.6$dB.)

With both scales in dB we expect a graph slope of one, and we find that as Q rises from 36.1dB to 42.1dB, the magnitude of $|M_3(f_0, f_0, -f_0)|$ rises from 72.4dB to 78.3dB — which confirms the expected result. Only six data points are plotted, but it is difficult to do take measurements over a wider range for several reasons:

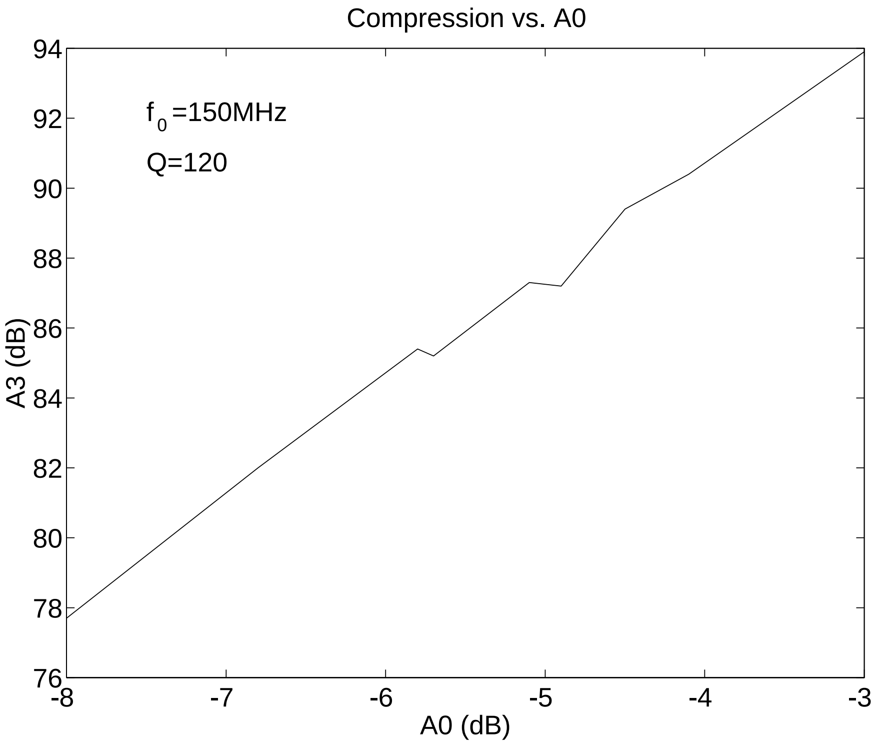


Figure 6.8: $|M_3(f_0, f_0, -f_0)|$ as a function of filter A_0.

- The amount of tweaking required to get the same A_0 and f_0 is quite tedious by hand.
- Filter drift is particularly problematic; measurements must be taken rapidly.
- NL% starts to become significant if Q is varied too much more.

The first two points could be addressed by automating the tuning, but it was not felt that the effort was worth it. The results quoted here are satisfactory.

6.4.3 Peak Filter Gain

The same filter with $f_0 = 150$MHz was characterized for normalized $Q = 120$ and A_0 varying from -8.0dB to -3.0dB. The measured data is plotted in Figure 6.8.

The expected results are closely followed in this case as well: for a 5.0dB increase in A_0, $|M_3|$ increases by 16.2dB. Ideally the increase should be 15dB, so the observed ratio 3.2:1 is not exact, but it is close.

6.4.4 Channel Separation

We have only been measuring the compression at the f_0 so far; let us try to measure desensitization $M_3(f_0, f_b, -f_b)$ for various f_b around f_0. This will allow us to plot a graph like Figure 4.31 and so see how desensitization changes as a function of CS%.

The QBASIC program to measure the data shown in Figure 6.5 was modified to control both the network analyzer (the desired tone, V_a at f_0) and the SMHU (the interfering tone V_b at f_b) over the HPIB. Again, care must be taken to ensure V_b is not too large to render M_5 significant, so the program automatically measures M_3 at the point where the third-order component is 15% of the first-order component, exactly as was done in §6.3.3. Both the linear transfer function M_1 for a filter with $(f_0, Q, A_0) = (100\text{MHz}, 83, -11.1\text{dB})$ and the desensitization at the center frequency $M_3(f_0, f_b, -f_b)$ are plotted in Figure 6.9.

As predicted in Figure 4.31 desensitization gets more severe for frequencies close to f_0, and it peaks at f_0. One thing not shown in §4.6.4 is how the *phase* of $M_3(f_0, f_b, -f_b)$ varies with f_b. A quick check of the Volterra transfer function reveals that phase remains constant — certainly a nice thing to see, given that the phase of M_3 in Figure 4.31 is mostly constant. Its variation over the range CS% $= -5$ to CS% $= 5$ is confined to $\pm 2.5^o$, and it can be attributed to measurement noise.

6.4.5 Volterra Surfaces

To close out our measurements, we will automate the measurement of the Volterra transfer function $M_3(f_a, f_b, -f_b)$ over a range of f_a and f_b values around f_0 and display some pretty three-dimensional plots of the results. Each of the forthcoming plots was generated completely automatically: the program steps both frequencies

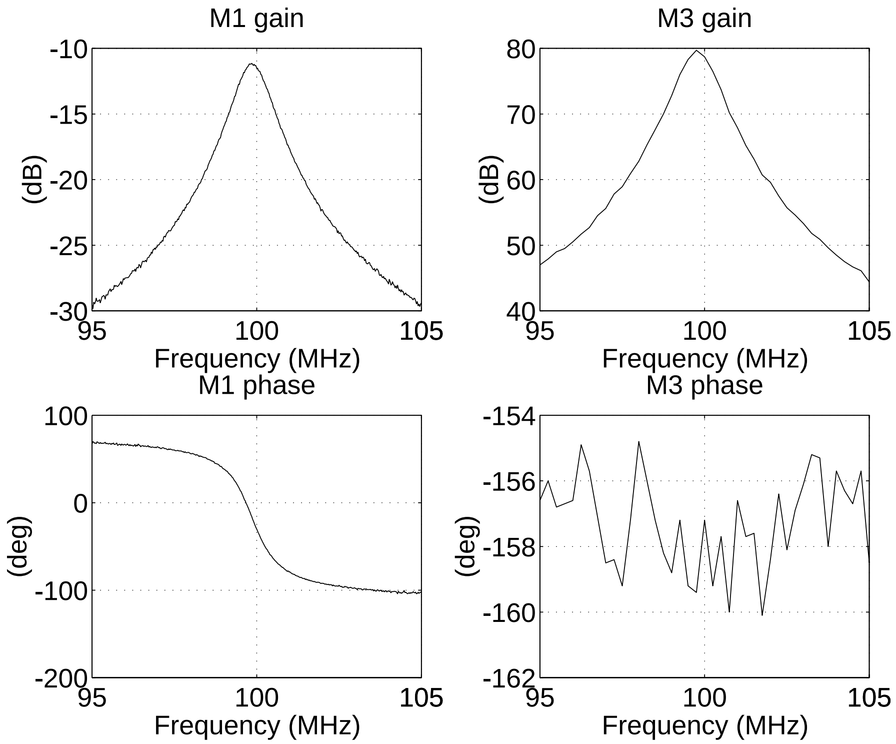


Figure 6.9: Linear gain (left); desensitization at f_0 as a function of f_b (right).

and determines the optimum input signal levels at which to measure distortion and summarizes the results to disk. Measurements at one hundred (f_a, f_b) points can be taken in about ten minutes.

Figure 6.10 illustrates the Volterra magnitude and phase surfaces for a filter with $(f_0, Q, A_0) = (98.8\text{MHz}, 125, 1.0\text{dB})$. The $|M_3|$ surface reaches a maximum of 109.4dB for f_a and f_b both equal f_0. The M_3 graphs in Figure 6.9 are simply the cross-sections of Figure 6.10 with $f_a = f_0$. The phase graph is constant for fixed f_a just as it was in Figure 6.9.

Figure 6.11 features a filter with $(f_0, Q, A_0) = (98.8\text{MHz}, 45, -7.8\text{dB})$. The maximum $|M_3|$ value still occurs at $f_a = f_b = f_0$ but is has become 76.8dB. It is easy to explain why: Q and A_0 have fallen by 8.9dB and 8.8dB, respectively, so the peak compression should fall by $3 \times 8.8 + 8.9 = 35.3$dB. It fell from 109.4dB to 76.8dB,

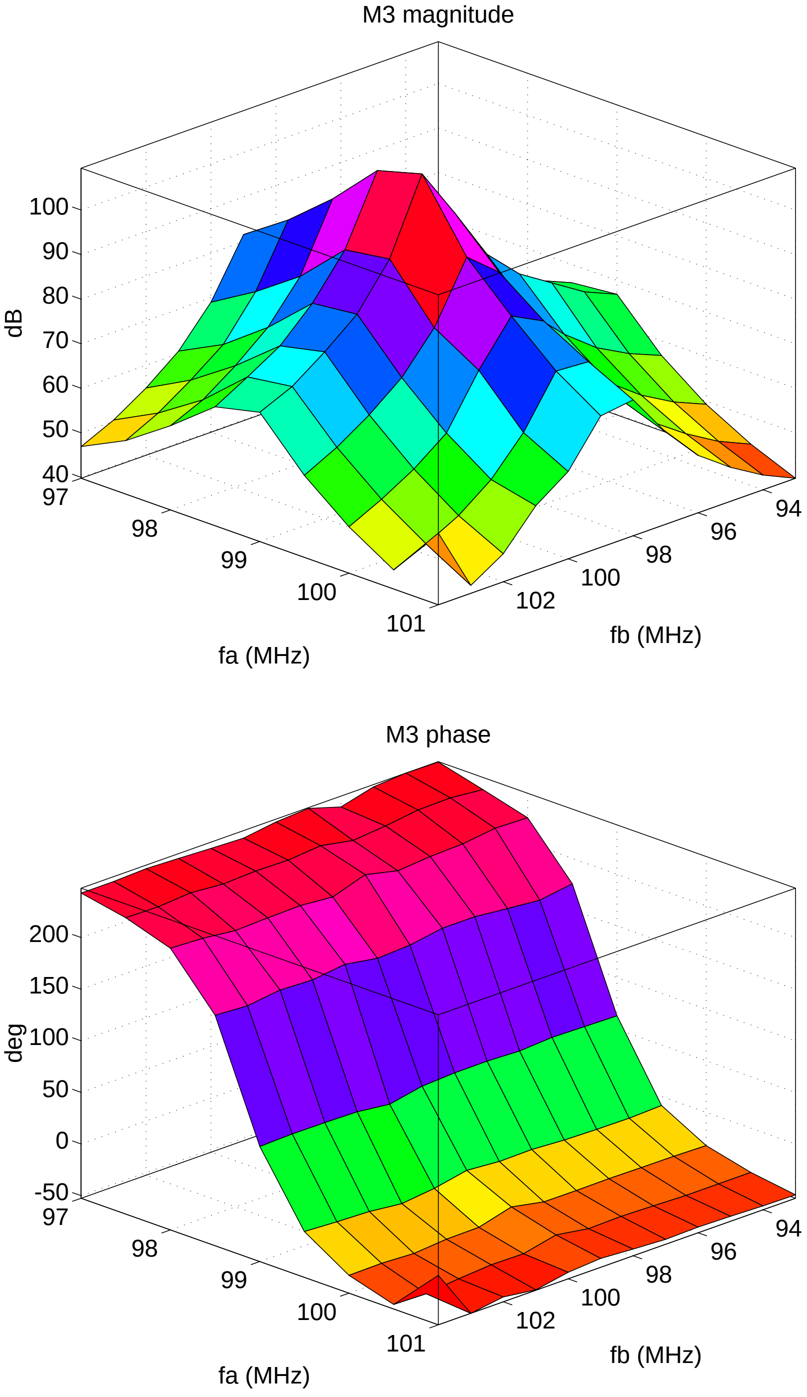


Figure 6.10: Plots of $M_3(f_a, f_b, -f_b)$ for $f_0 = 98.8$MHz, $Q = 125$, $A_0 = 1.0$dB.

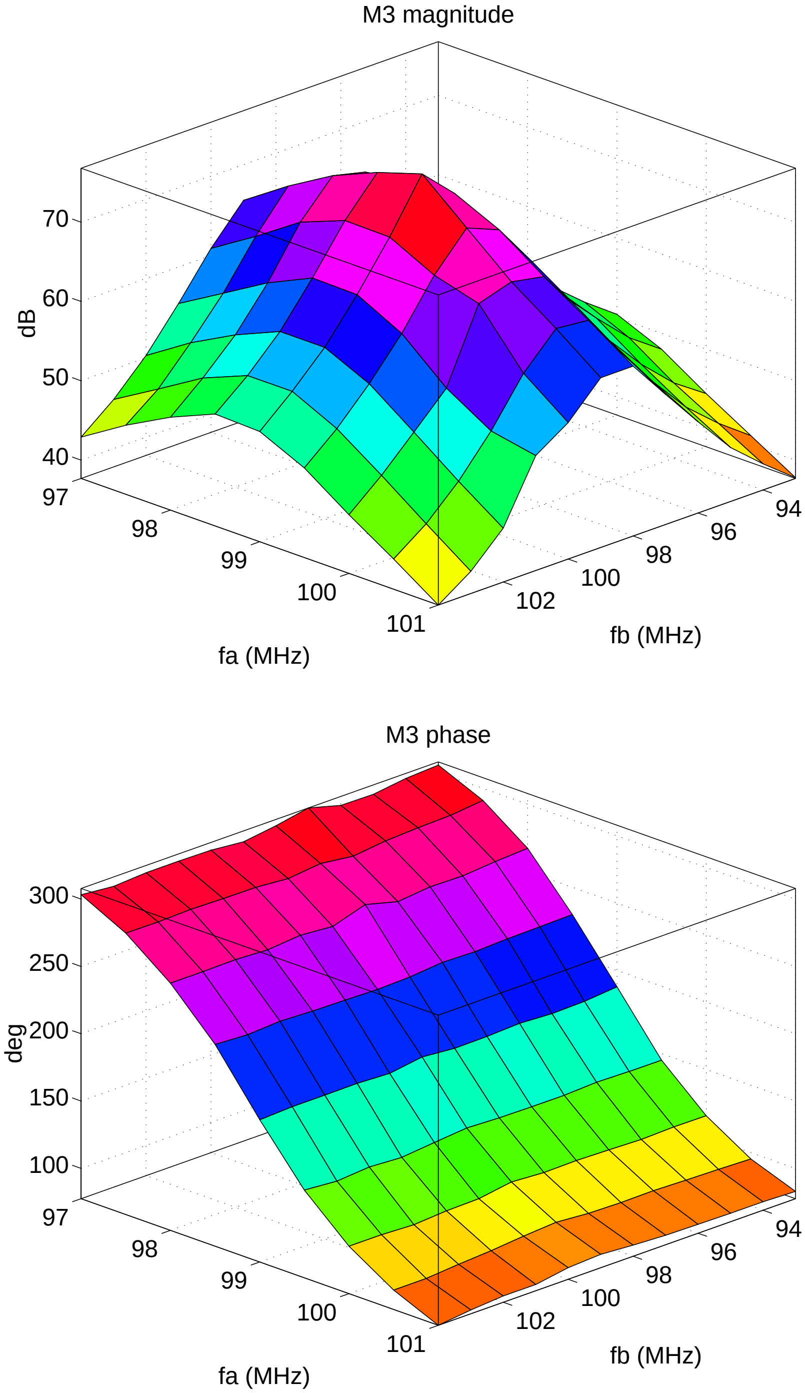


Figure 6.11: Plots of $M_3(f_a, f_b, -f_b)$ for $f_0 = 98.8\text{MHz}$, $Q = 45$, $A_0 = -7.8\text{dB}$.

a drop of 32.6dB, quite close to the expected value. Again, the phase is constant to within measurement noise for fixed f_a.

Chapter 7

Conclusions

7.1 Summary

Overall, this thesis has shown that for analyzing distortion in weakly-nonlinear band pass filters, Volterra series are a powerful tool.

In Chapter 4, the Volterra transfer functions for a second-order biquadratic band pass filter were derived and the results checked against numerical simulation. Strong nonlinearity was studied with the forced Duffing equation. For weak nonlinearities, Volterra series were shown to agree almost perfectly with simulated one- and two-tone inputs. General distortion trends as a function of filter variables were derived and explained.

In Chapter 5, an architecture aimed at reducing distortion with three parallel filters and feedback was analyzed. Its linear and nonlinear transfer functions were derived and calculations showed that under some conditions distortion is reduced. A method for extracting distortion terms from numerical simulations was derived and demonstrated. General distortion trends as a function of filter variables were examined.

In Chapter 6, the extraction method proposed in Chapter 5 was applied to a real integrated circuit with good success. Compression and desensitization values for various filter configurations were measured. Gain compression and expansion were explained in terms of Volterra transfer functions and 1dB-compression point inputs were predicted. The general distortion trends in Chapter 4 were shown to hold, which implies that the results in this thesis apply to any architecture so long as it is a second-order biquadratic BPF.

7.2 Additional Research

The following issues that arose in this thesis should be studied in greater detail in future work.

1. Generating Volterra transfer functions in §4.2 needs to be automated in a symbolic algebra program. The Laplace transform command in Maple, for example, is implemented with a simple set of base rules and a list of how to apply them. Such an approach could be adopted for Volterra transfer function calculation.

2. The suitability of 3filt for a specific application needs to be investigated. What sort of architecture should be used? How will it be tuned? What frequency will it operate at? Should it be at RF or IF? What should it notch out? How close can the notches be? These are a sampling of the questions that need to be answered.

3. Off-tuning 3filt so that the gain peak in Figure 5.11 coincides with the desired tone should be investigated to determine its distortion performance.

4. General trend graphs such as those in Figure 5.15 and Figure 5.17 would be interesting to draw for the realistic 3filt simulated in §5.5.2. It would be interesting to see under what conditions using that filter in a 3filt significantly outperforms a 1filt.

5. The filter tuning in §6.4.2 and §6.4.3 could be automated. The potentiometer resistive dividers could be replaced with programmable voltage sources, and in

this way the distortion for many different filter configurations could be measured easily.

6. Laboratory measurements on a 3filt should be performed. They were omitted in this thesis because of the absence of a suitable circuit. An attempt was made to measure the distortion in two separate filters connected with splitters and cables but phase shifts rendered the attempt futile. A single-board 2filt or 3filt is required.

7. This thesis considered only sinusoidal signals. To be realistic, signals with modulation need to be considered, although this is a much more difficult task. [Buss74] looks at the problem somewhat.

Bibliography

[Ada94] T. Adachi, A. Ishikawa, K. Tomioka, S. Hara, K. Takasuka, H. Hisajama, and A. Barlow, "A Low Noise Integrated AMPS IF Filter," *IEEE 1994 Custom Integrated Circuits Conference* proceedings, pp. 159–162.

[Bed71] E. Bedrosian and S. O. Rice, "The Output Properties of Volterra Systems (Nonlinear Systems with Memory) Driven by Harmonic and Gaussian Inputs," *Proc. IEEE*, vol. 59, Dec. 1971, pp. 1688–1707.

[Boyd83] S. Boyd and L. O. Chua, "Measuring Volterra Kernels," *IEEE Trans. Circuits and Systems*, vol. 30, March 1983, pp. 571–577.

[Boyd85] S. Boyd and L. O. Chua, "Fading Memory and the Problem of Approximating Nonlinear Operators with Volterra Series," *IEEE Trans. Circuits and Systems*, vol. 32, Nov. 1985, pp. 1150–1161.

[Buss74] J. J. Bussgang, L. Ehrman, and J. W. Graham, "Analysis of Nonlinear Systems with Multiple Inputs," *Proc. IEEE*, Aug. 1974, pp. 1088–1119.

[Chua69] L. O. Chua, *Introduction to Nonlinear Network Theory.* New York: McGraw-Hill, 1969.

[Chua82] L. O. Chua and Y. Tang, "Nonlinear Oscillation via Volterra Series," *IEEE Trans. Circuits and Systems*, vol. 29, March 1982, pp. 150–169.

[Cook68] A. B. Cook and A. A. Liff, *Frequency Modulation Receivers.* Englewood Cliffs: Prentice-Hall, 1968.

[Culb84] J. J. Culbert, *Distortion Analysis in Filters Containing Weakly Nonlinear Elements.* Bachelor's Thesis, University of Toronto, Toronto, Canada, 1984.

[Fish79] R. E. Fisher, "A Subscriber Set for the Equipment Test," *Bell System Technical Journal*, vol. 58, January 1979, p. 133.

[IEEE87] *Proc. IEEE*, vol. 75, Aug. 1987.

[Lath74] B. P. Lathi, *Signals, Systems, and Controls.* New York: Intext, 1974.

[Lee64] Y. M. Lee et al., *Selected Papers of Norbert Wiener.* Cambridge, Mass.: M.I.T. Press, 1964.

[Mey94] R. G. Meyer and W. D. Mack, "A 1GHz BiCMOS RF Front-End IC," *IEEE Journal of Solid-State Circuits*, vol. 29, March 1994, pp. 350-355.

[Nar70] S. Narayanan, "Application of Volterra Series to Intermodulation Distortion Analysis of Transistor Feedback Amplifier," *IEEE Trans. Circuit Theory*, vol. 17, Nov. 1970, pp. 518–527.

[Nayf79] A. H. Nayfeh and D. T. Mook, *Nonlinear Oscillations.* New York: John Wiley and Sons, 1979.

[Nguy90] N. M. Nguyen and R. G. Meyer, "Si IC-Compatible Inductors and *LC* Passive Filters," *IEEE Journal of Solid-State Circuits*, vol. 25, Aug. 1990, pp. 1028–1031.

[Nguy92] N. M. Nguyen and R. G. Meyer, "A 1.8-GHz Monolithic *LC* Voltage- Controlled Oscillator," *IEEE Journal of Solid-State Circuits*, vol. 27, March 1992, pp. 444–450.

[Pad91] M. Padmanabhan and K. Martin, "Resonator-Based Filter-Banks for Frequency-Domain Applications," *IEEE Trans. Circuits and Systems*, vol. 38, Oct. 1991, pp. 1145–1159.

[Pap80] A. Papoulis, *Circuits and Systems.* New York: Holt, Rinehart, and Winston, 1980.

[Pre92] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge: Cambridge University Press, 1992.

[Rap94] J. Rapeli, "IC Solutions for Mobile Telephones", from *Design of Analog-Digital VLSI Circuits for Telecommunications and Signal Processing*, 2nd ed., edited by J. E. Franca and Y. Tsividis. Englewood Cliffs: Prentice-Hall, 1994.

[Rugh81] W. J. Rugh, *Nonlinear System Theory: The Volterra/Weiner Approach.* Baltimore: John Hopkins University Press, 1981.

[Sand83a] I. W. Sandberg, "On Volterra Expansions for Time-Varying Nonlinear Systems," *IEEE Trans. Circuits and Systems*, vol. 30, Feb. 1983, pp. 61–67.

[Sand83b] I. W. Sandberg, "Volterra-like Expansions for Solutions of Nonlinear Integral Equations and Nonlinear Differential Equations," *IEEE Trans. Circuits and Systems*, vol. 30, Feb. 1983, pp. 68–77.

[Sand83c] I. W. Sandberg, "The Mathematical Foundations of Associated Expansions for Mildly Nonlinear Systems," *IEEE Trans. Circuits and Systems*, vol. 30, July 1983, pp. 441–455.

[Sch90] R. Schaumann, M. S. Ghausi, and K. R. Laker, *Design of Analog Filters.* Englewood Cliffs: Prentice-Hall, 1990.

[Shov93] A. Shoval, D. A. Johns, and W. M. Snelgrove, "A Wide-Range Tunable BiCMOS Transconductor," *Microelectronics Journal*, vol. 24, Aug. 1993, pp. 555–564.

[Vdp34] B. Van der pol, "The Nonlinear Theory of Electric Oscillations," *Proc. IRE*, vol. 22, Sept. 1934, pp. 1051–1086.

[Vol59] V. Volterra, *Theory of Functionals and of Integro and Integro-Differential Equations*. New York: Dover, 1959. Reprint of 1930 edition.

[Wein80] D. D. Weiner and J. F. Spina, *Sinusoidal Analysis and Modeling of Weakly Nonlinear Circuits*. New York: Van Nostrand Reinhold, 1980.

[Wien58] N. Wiener, *Nonlinear Problems in Random Theory*. Cambridge, Mass.: M.I.T. Press, 1958.

Appendix A

Harmonic Input Method

We shall prove the assertion from §3.2.1: when the input to a system is

$$x(t) = \exp(j\omega_1 t) + \ldots + \exp(j\omega_n t) \tag{A.1}$$

where $\omega_i = 2\pi f_i, i = 1, \ldots, n$, and the ω_i are incommensurable, then the nth Volterra transfer function can be found from

$$G_n(f_1, \ldots, f_n) = \{\text{coefficient of } \exp[j(\omega_1 + \ldots + \omega_n)t] \text{ in (3.3)}\} \tag{A.2}$$

To begin, substitute (A.1) in the general equation for $y(t)$, equation (3.3)

$$y(t) \quad = \quad \sum_{n=1}^{\infty} \frac{1}{n!} \int_{-\infty}^{\infty} du_1 \cdots \int_{-\infty}^{\infty} du_n g_n(u_1, \ldots, u_n) \prod_{r=1}^{n} x(t - u_r)$$

and consider its nth term

$$y_n(t) = \frac{1}{n!} \int_{-\infty}^{\infty} du_1 \cdots \int_{-\infty}^{\infty} du_n g_n(u_1, \ldots, u_n) \prod_{r=1}^{n} \left[\sum_{s=1}^{n} \exp(j\omega_s(t - u_r)) \right]$$

Expanding the product of sums on the right, and keeping only the $\exp[j(\omega_1 + \ldots + \omega_n)t]$ term, yields

$$y_n(t) = \frac{1}{n!} \exp(j \sum_{s=1}^{n} \omega_s t) \int_{-\infty}^{\infty} du_1 \cdots \int_{-\infty}^{\infty} du_n g_n(u_1, \ldots, u_n) \sum_{n!} \exp(-j \sum_{s=1}^{n} \omega_s u_{(s)})$$

where the $\sum_{n!}$ and $u_{(s)}$ denote summation over the $n!$ permutations of the subscripts of the u_i. The summation $\sum_{n!}$ may be moved outside the integrals

$$y_n(t) = \frac{1}{n!}\exp(j\sum_{s=1}^{n}\omega_s t)\sum_{n!}\int_{-\infty}^{\infty}du_1\cdots\int_{-\infty}^{\infty}du_n g_n(u_1,\ldots,u_n)\exp(-j\sum_{s=1}^{n}\omega_s u_{(s)})$$

The integral in this expression is simply the n-fold Fourier transform of g_n, equation (3.8), leading to

$$y_n(t) = \frac{1}{n!}\exp(j\sum_{s=1}^{n}\omega_s t)\sum_{n!}G_n(f_{(1)},\ldots,f_{(n)}) \tag{A.3}$$

where, again, the subscripts on the $f_{(i)}$ indicate they are to be permuted over all $n!$ possibilities. But now recall that the G_n are symmetric. That is, the arguments f_i may be permuted without altering the value of $G_n(f_1,\ldots,f_n)$. Thus

$$\sum_{n!}G_n(f_{(1)},\ldots,f_{(n)}) = n!\,G_n(f_1,\ldots,f_n) \tag{A.4}$$

and (A.4) in (A.3) gives

$$y_n(t) = \exp(j\sum_{s=1}^{n}\omega_s t)G_n(f_1,\ldots,f_n)$$

proving (A.2).

Appendix B

1dB-Compression Point

We will derive the 1dB-compression point for a system with Volterra kernels $M_1(f) = A_1\angle\theta_1\mathrm{dB}$ and $M_3(f,f,-f) = A_3\angle\theta_3\mathrm{dB}$. For an input V_x, the linear output component can be found from (6.4) to be

$$\begin{aligned} V_x M_1 &= V_x 10^{\frac{A_1}{20}}(\cos\theta_1 + j\sin\theta_1) \\ &= V_x B_1\cos\theta_1 + jV_x B_1\sin\theta_1 \end{aligned} \tag{B.1}$$

where $B_1 = 10^{\frac{A_1}{20}}$. The first- and third-order output components together add to

$$\begin{aligned} V_x M_1 + \frac{V_x^3}{8}M_3 &= V_x 10^{\frac{A_1}{20}}(\cos\theta_1 + j\sin\theta_1) + \frac{V_x^3}{8}10^{\frac{A_3}{2}}(\cos\theta_3 + j\sin\theta_3) \\ &= [V_x B_1\cos\theta_1 + \frac{V_x^3}{8}B_3\cos\theta_3] + j[V_x B_1\sin\theta_1 + \frac{V_x^3}{8}B_3\sin\theta_3] \end{aligned} \tag{B.2}$$

The 1dB-compression point occurs when $V_x = V_{c1}$ and the magnitudes of (B.2) and (B.1) differ by -1dB:

$$\left|\frac{[V_{c1}B_1\cos\theta_1 + \frac{V_{c1}^3}{8}B_3\cos\theta_3] + j[V_{c1}B_1\sin\theta_1 + \frac{V_{c1}^3}{8}B_3\sin\theta_3]}{V_{c1}B_1\cos\theta_1 + jV_{c1}B_1\sin\theta_1}\right| = -1\mathrm{dB} \tag{B.3}$$

Dividing top and bottom of the LHS by $V_{c1}B_1$ and rewriting the RHS as $10^{\frac{-1}{20}}$ gives

$$\left|\frac{[\cos\theta_1 + \frac{V_{c1}^2}{8}\frac{B_3}{B_1}\cos\theta_3] + j[\sin\theta_1 + \frac{V_{c1}^2}{8}\frac{B_3}{B_1}\sin\theta_3]}{\cos\theta_1 + j\sin\theta_1}\right| = 10^{-0.05}$$

Evaluating the magnitude on the LHS and squaring both sides gives

$$\left[\cos\theta_1 + \frac{V_{c1}^2}{8}\frac{B_3}{B_1}\cos\theta_3\right]^2 + j\left[\sin\theta_1 + \frac{V_{c1}^2}{8}\frac{B_3}{B_1}\sin\theta_3\right]^2 \quad = \quad 10^{-0.1}$$

Expanding the terms on the LHS and collecting yields

$$\left(\frac{V_{c1}^2}{8}\frac{B_3}{B_1}\right)^2 + 2\cos(\theta_1-\theta_3)\left(\frac{V_{c1}^2}{8}\frac{B_3}{B_1}\right) + \left(1-10^{-0.1}\right) \quad = \quad 0 \tag{B.4}$$

Using the quadratic formula on (B.4) and simplifying:

$$\begin{aligned}\frac{V_{c1}^2}{8}\frac{B_3}{B_1} &= \frac{-2\cos(\theta_1-\theta_3) \pm \sqrt{4\cos^2(\theta_1-\theta_3) - 4(1-10^{-0.1})}}{2} \\ &= -\cos(\theta_1-\theta_3) \pm \sqrt{10^{-0.1} - \sin^2(\theta_1-\theta_3)} \end{aligned} \tag{B.5}$$

The two roots of (B.5) correspond to the two different 1dB-compression points as follows: for very small inputs, the third-order term contributes almost nothing, so there is 0dB of compression. As the input becomes larger, the third-order term starts to contribute more until the compression is 1dB; this corresponds to the negative root of (B.5). The compression continues to get larger until the third-order term starts to dominate the first-order term. The ratio in (B.3) then starts to become more positive, making the compression drop. Eventually the compression will pass through 1dB again, and this corresponds to the positive root of (B.5). Increasing the input still further leads to negative compression, i.e., gain expansion.

We are interested in the negative root only, and so the input level, in volts, which causes -1dB of compression at the output is obtained by solving (B.5):

$$\begin{aligned} V_{c1} &= \sqrt{\frac{8B_1}{B_3}\left[-\cos(\theta_1-\theta_3) - \sqrt{10^{-0.1} - \sin^2(\theta_1-\theta_3)}\right]} \\ &= \sqrt{8\times 10^{\frac{A_1-A_3}{20}}\left[-\cos(\theta_1-\theta_3) - \sqrt{10^{-0.1} - \sin^2(\theta_1-\theta_3)}\right]} \end{aligned} \tag{B.6}$$

for A_1 and A_3 in dB. Incidentally, the quantity under the outer square-root sign is non-negative for $+117^o < \theta_1 - \theta_3 < -117^o$. For $\theta_1 - \theta_3$ outside this range, one of two things might occur:

1. Compression occurs, but it is never as much as -1dB.

2. Gain *expansion* rather than gain compression occurs.