A BEHAVIORAL SIMULATOR FOR SIGMA-DELTA ANALOG TO DIGITAL CONVERTERS

By

ANURAG K. TYAGI

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1994

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 1996

A BEHAVIORAL SIMULATOR FOR

SIGMA-DELTA ANALOG TO

DIGITAL CONVERTERS

Thesis Approved:

Thesis Advisor

Thomas C. Collins

Dean of the Graduate College

ACKNOWLEDGMENTS

The completion of this thesis marks the culmination of my Master's Degree in Electrical Engineering, along with the foreseeable completion of my formal education. Throughout my life, there are definitely many people whom I would like to have shown my gratitude, but have passed by or have not taken the opportunity to do so. Since it is widely accepted that a thesis acknowledgment is a conventional site to thank those who have aided one in achieving this pinnacle, I will try to use this opportunity to its fullest.

To begin, I am dedicating this thesis to my sister, Ami, my mother, Abha, and my father, Avdhesh. Throughout my tumultuous life, they have sustained an essential connection with me which has eventually been the motivation for all of my accomplishments. This is especially true of Ami whose abounding love has lifted my spirits in my darkest times.

I also deeply appreciate the love and guidance given by the rest of my family, here in America and in India. My cousins, Akhilesh and Dinesh, have always been my brotherly role models. I have always striven towards the strength, intelligence, and love by which they have characterized their lives. My cousin, Yesh, who has always been encouraging in every aspect of my life, is a faithful brother and friend. My cousin-in-laws, Soma and Geeta, who have been loving, tender, and affectionate elder sisters. My uncles and aunts,

Mukul and Savita, Subhash and Madhu, Anil and Amita, Satyavati, and the rest of my extended family have all emphatically and fondly supported me throughout my life. Most of all, my grandparents, Ram Autar, Ramesh, Daya Wati, and Prem, have played a crucial part of my life. Their loving guidance has molded my every spiritual and material manner.

I also greatly treasure the camaraderie of all my close friends and coworkers. Over the past couple of years, my lab-mates, Soon 'Jeff' Lim, Prashanth Adiraju, and Jie Zheng, have given me encouragement, friendship, and assistance, all being very valuable in the completion of my work at Oklahoma State University. I am very proud to have worked with them in the Advanced Analog VLSI Design Center (AAVDC). I will never forget the time we have spent together. Furthermore, my experiences with my friends from Austin have been very important in the shaping of my life. Ajay, Babin, Meenu, Mukund, Nabeel, Noel, Vikram and the rest of the Austin crew left me with substantial, meaningful memories reaching into every aspect of my life. My experiences with all the people I have mentioned above have made me the person I am today. It is impossible to express my love and gratitude for each and every one of them.

I would like to thank the faculty of the ECEN Department at Oklahoma State University, especially, Dr. Keith Teague and Dr. John Cartinhour for being on my Graduate Committee. I am also indebted to Ms. Rea Maltsberger whose referral gave me my first step towards the completion of my research work. In addition, I am grateful to Naval Research and Development (NRaD), San Diego and its staff for giving me the chance to complete this work. Last and foremost, I would like to thank my advisor and friend, Dr. Chris Hutchens. I am deeply appreciative of all the opportunities he has presented me over the past two years. His tutelage and guidance have enhanced my chances for success at Oklahoma State University and my subsequent career. He is truly a 'guru' in every sense of the word.

TABLE OF CONTENTS

1.

LIST OF FIGURES

LIST OF SYMBOLS

'a'	Slope of transition voltage from on to off state of MOSFET
AAF	Anti-aliasing filter
A_i	Interpolative architecture's i^{th} feedback gain
A_V	Integrator voltage gain
A_{vol}	OTA open-loop gain
B	Number of bits
B_0	Interpolative architecture's feedback gain
C'	OTA input capacitance
$C_{eq,block}$	Block equivalent capacitance
C_{fb}	Feedback capacitor in DAC
C_{g1}	Effective sampling parasitic capacitance
C_{gd}	MOSFET gate to drain parasitic capacitance
C_{gs}	MOSFET gate to source parasitic capacitance
C_{int}	Integration capacitor in switched-capacitor integrator
C_{L1}	Integrator load capacitance
C_{ox}	MOS oxide capacitance
C_{si}	Sampling capacitor in switched-capacitor integrator
C_x	OTA gate to output parasitic capacitance
DAC	Digital-to-analog converter
DR	Dynamic range
e_i	Quantization error introduced by i^{th} $\Sigma\Delta$ loop stage
e_Q	Quantization error
f_B	Frequency band-of-interest or frequency pass-band in hertz

FFT	Fast Fourier Transform
f_N	Nyquist frequency in hertz
freq_samp	Sampling frequency constant in $\Sigma\Delta$ toolbox
freq_des	Desired input frequency constant in $\Sigma\Delta$ toolbox
f_S	Sampling frequency in hertz
f_{SB}	Stop-band frequency in hertz
g_i	Residual Scaling $\Sigma\Delta$ architecture's i^{th} digital gain
g_{ds}	MOSFET drain to substrate admittance
g_{m1}	OTA's effective transconductance
g_o	OTA output admittance
GSPS	Gigasamples per second
jitter_err	Maximum percentage time deviation around ideal time-sampled sample
k	Integer sample
K_i	Interpolative architecture's i^{th} integrator gain
kSPS	Kilosamples per second
L	$\Sigma\Delta$ modulator order
M	Oversampling ratio
max_in	Maximum input voltage constant in $\Sigma\Delta$ toolbox
mis_err	Maximum percentage of process component mismatch error
MOSFET	Metal on Substrate Field Effect Transistor
NSTerm	Noise-shaping term of $\Sigma\Delta$ modulator
NTF	Noise transfer function
num_bits	Number of quantizer bits constant in $\Sigma\Delta$ toolbox
num_cycles	Number of desired input cycles constant in $\Sigma\Delta$ toolbox
OSR	Oversampling ratio
OTA	Operational Transconductance Amplifier
over_ratio	Oversampling ratio constant in $\Sigma\Delta$ toolbox
ph_time	Clocking phase time in seconds
P_{NS}	Average noise power

q	Unit sample delay
R_{amp}	Amplifier amplification resistor
R_{g1}	Effective switch resistance
$R_{eq,block}$	Block equivalent resistance
R_{si}	Amplifier input resistor
s	Laplacian Operator
S_{ee}	Output noise power
SNR	Signal-to-noise ratio or signal-to-quantization-noise ratio
$SNR_{Residual}$	SNR for Residual Scaling ΣΔ modulator
SPS	Samples per second
STF	Signal transfer function
S_{xx}	Input signal power
$S_{xx}\|_{FS}$	Largest input signal which does not permit clipping
$S_{xx}\|_{SNR=1}$	Input signal power at which SNR is unity
t_{fall}	MOSFET fall time from on to off state in seconds
T_S	Sampling period
V_{DAC}	DAC output voltage
V_{ds}	MOSFET Drain to Source Voltage
V_{FS}	Full-scale voltage
V_g	MOSFET gate voltage
$V_{g,ON}$	MOSFET gate voltage for on state
V'	OTA input voltage
V_{gs}	MOSFET gate to source voltage
V_{in}	Integrator input signal
V_{in}	Input signal of ΣΔ modulator
V_{int}	Integrator input voltage
V_{int_out}	Integrator output voltage without settling error
v_o	OTA output voltage
$V_{out,block}$	Block output voltage with settling error

V_{out}	Block output voltage without settling error
V_{qout}	Quantizer output voltage
V_{ref_DAC}	DAC reference voltage
w	Output signal of integrator
x_i	Input to ith $\Sigma\Delta$ loop stage
Y_{f1}	MOS Integration Capacitor's Admittance
Y_{g1}	MOS Switch's Admittance
Z_i	Block impedance without mismatch error
Z'_i	Block impedance with mismatch error
β	Kaiser window beta term
Δ	Quantization step size
γ_i	Residual Scaling $\Sigma\Delta$ architecture's ith analog gain
ϕ_1	First phase of clocking cycle
ϕ_2	Second phase of clocking cycle
ω	Frequency in radians
ρ_e	Probability distribution of quantization error
$\Sigma\Delta$	Sigma-Delta
σ_e^2	Variance of quantization error
σ_x^2	Variance of input
τ_{block}	Block time delay

Chapter 1 *Introduction*

In recent years, there has been a radical shift from analog systems to digital systems paralleling the advent of digital computing. Signal processing in these digital systems has impacted virtually all aspects of life even remotely concerned with electronics. Common applications include *clear sounding* compact discs, multimedia computer applications, pocket cellular telephones, digital networking, innovative high definition television, and more. Moreover, developing process technologies have provided an inexpensive, high-speed medium for the proliferation of digital systems at the integrated chip (IC) level. Since the 'real world' interacts in an analog environment, there exists formidable tasks in converting everyday analog signals into accurate digital signals for a digital system to process and vice versa.

These transformations, termed as Analog-to-Digital Conversion (A/D) and Digital-to-Analog Conversion (D/A), are vital for human related interaction with electronic communication systems. But, in order to satisfy consumer application demands, these conversion tasks are compounded by the electronic industry's ever growing need for faster, more precise, and higher bandwidth A/D and D/A conversion. In addition, there is a demand for single chip solutions that provide an increase in overall reliability, an easier integration of mixed-mode systems, and the use of lower tolerance components. Unfortunately, the precision of scaled IC components limits the attempted accuracy of Nyquist-rate A/D converters (ADC) and D/A converters (DAC) to about 10 to 12 bits of digital resolution using traditional approaches, like successive approximation and flash converters[Van De Plassche, 1994]. This hindrance is one reason for the recent increase in research of innovative higher resolution ADC techniques.

Today, in systems where feasible operating clock rates are well above signal bandwidth requirements, oversampled A/D conversion techniques produce an overall reduction of noise power. In addition, noise-shaping methods have been employed to further reduce noise power by attenuating the noise floor in the frequency band of interest while augmenting the floor outside the band. During the past 15 years, an oversampled, noise-shaping technique known as Sigma-Delta ($\Sigma\Delta$) Modulation has become popular due to its resilience to limited device matching accompanying its increase in high resolution conversion performance over conventional ADCs [Boser, 1988; Candy, 1985; Nadeem, 1994].

Implementation of these $\Sigma\Delta$ ADCs from theory to IC has been hindered by inefficient or insufficiently accurate simulation programs. Currently, after the initial design of a $\Sigma\Delta$ architecture, verification through simulation has been very time-consuming. Current commercially available simulation packages are either too slow, like SPICE, or do not include sufficient behavioral analysis, like SwitCAP, to deduce accurate ADC performance measures in a timely manner. To overcome this obstacle, two proprietary simulators are known to have been developed at other universities to aid their design centers. The first is MIDAS, a mixed-mode, sampled-data simulator developed at the Center for Integrated Systems, Stanford University, California [Boser, 1988]. The second was developed at the Massachusetts Institute of Technology to study nonlinear circuit effects in ΣΔ architectures [Chao, 1990].

Thus, the motivation for this work was the development of a simulation package which behaviorally models ΣΔ ADC architectures for use by the Advanced Analog VLSI Design Center, Oklahoma State University, Oklahoma (AAVDC) in its ΣΔ design efforts. This package, also referred to as the ΣΔ toolbox, allows for the limited inclusion of component non-idealities which are known to significantly hamper A/D conversion [Van De Plassche, 1994]. The use of this toolbox will aid the designer in rapid prototyping and behavioral insight of a variety of ΣΔ architectures. The toolbox has been used in the design and verification of novel ΣΔ architectures being developed by the AAVDC for the Naval Research and Development Division (NRaD) of the Naval Command, Control, and Ocean Surveillance Center (NCCOSC) in San Diego, California.

1.1 Organization

Following this introduction, various aspects of ΣΔ A/D conversion are covered. Chapter 2 develops a basis for understanding of ΣΔ ADCs by investigating the fundamental types of A/D conversion: Nyquist rate conversion and Oversampled conversion. This analysis allows for an introduction to the concepts involved in A/D conversion and as a foundation for evaluating the worth of ΣΔ A/D conversion. Theoretical performances of the fundamental A/D conversion types will be given for comparison to ΣΔ A/D conversion.

In Chapter 3, modeling and simulation of ΣΔ A/D conversion is thoroughly examined. Behavioral aspects and theoretical performance formulas for the conversion process are discussed in detail. Throughout the discussion, simulations from the ΣΔ toolbox verify the purported theoretical behavior and performance. Crucial non-idealities are considered and included in simulations to gain understanding of their role as limiting factors in ΣΔ A/D conversion. Lastly, two higher-order architectures being developed by the AAVDC are developed and analyzed to exhibit the use of the ΣΔ toolbox. They are a 3^{rd} order Residual Scaling ΣΔ architecture which employs a quantization error cancellation technique, and a 3^{rd} order Interpolative ΣΔ architecture based on an architecture developed by S. Nadeem [Nadeem, 1994].

A summary of the results of this research and the possibilities for future research are summarized in Chapter 4.

Chapter 2 Nyquist-Rate and Oversampled Analog-to-Digital Converters

Analog-to-Digital (A/D) conversion transforms a continuous-time, continuous-amplitude signal (analog signal) into a discrete-time, discrete-amplitude signal (digital signal) by two fundamental operations, sampling and quantization. The conversion's ability to produce accurate digital information is primarily limited by the Analog-to-Digital converter's (ADC) sampling speed and by the preciseness of quantization. Sampling defines the extent of the signal bandwidth; while quantization introduces noise. The ADC's A/D conversion resolution must be at least equal to the required resolution of subsequent digital signal processes. Meeting this requirement maintains the prescribed digital resolution essential to those digital systems irrespective the amount of noise added by the ADC. This chapter focuses on the above limitations and how they affect the resolution of two basic A/D conversion processes: Nyquist-rate converters and Oversampled converters.

The first section describes the effects of quantization noise in Nyquist-rate converters. After a theoretical analysis of the resolution of these converters, the next section illustrates oversampling and the resolution improvement it provides. Throughout this chapter, ideal behavior is considered for performance evaluations. That is, performance measurements are only limited by quantization noise. Further description of other significant non-idealities is considered in Chapter 3. Overall, this chapter provides a good foundation for the following chapter discussing Sigma-Delta ADCs.

2.1 Nyquist-Rate ADCs

A block diagram of a typical Nyquist-rate ADC is shown in Figure 2.1. The input to the system is a *real world* analog signal, x(t), which is continuous in time and in amplitude. This signal is pre-filtered by an analog, low-pass anti-aliasing filter (AAF). The filter prevents aliasing of sampled frequencies by limiting the input frequency range to the maximum frequency of the ADC's band-of-interest, f_B [Oppenhiem, 1989]. This band-limited signal, x'(t), is then sampled at the ADC's sampling frequency, f_S, in uniform time intervals. Thus, it transforms the signal into a discrete-time signal, $x(kT_S)$ where k is an integer. Nyquist's sampling theorem states that the minimum f_S (also known as the Nyquist rate) must be at least twice the Nyquist frequency, f_N, in order to prevent loss of information during sampling [Oppenhiem, 1989]. Nyquist-rate converters use a f_S that is slightly greater than $2f_b$ (by letting $f_B = f_N$) to exploit the slowest f_S required to satisfy the theorem. After sampling, the quantizer processes the discrete-time signal, $x(kT_S)$, into a fully digital signal with discrete levels, $y(kT_S)$. A simple description of the quantizer is an ideal transfer function with additive quantization error or noise, $e_Q(kT_S)$, as illustrated in Figure 2.1. Lastly, this digital information may be digitally encoded into binary for proper usage by subsequent digital signal processing. Specifying the number of bits desired in this binary mapping is a convenient and often used practice when defining the resolution of an ADC.

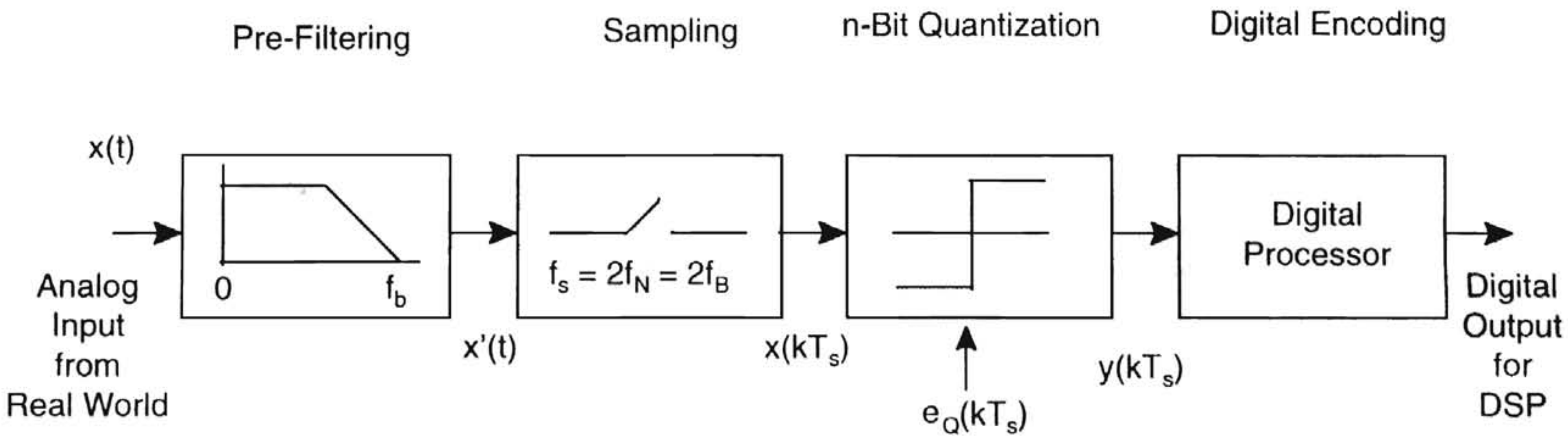


Figure 2.1 Block Diagram of a Nyquist-Rate ADC

Often, the signal-to-quantization-noise-ratio (SNR) of an ADC is given in terms of the number of digital bits resolution to determine an ADC's performance. So, an analysis of the quantization error introduced by the quantizer will aid in examining the performance of an ADC. The transfer function of a typical uniform mid-riser quantizer is given in Figure 2.2. The output is seen to be granular in that it is confined in discrete levels. In a uniform quantizer, there exists only two parameters: the number of decision levels and the quantization step size, Δ [Rabiner, 1978]. The number of levels is usually of the form 2^B; where B is the number of bits in the quantizer and also relates to the desired B-bit binary code words. Between each of these levels, the quantization step size is determined by

$$\Delta = \frac{V_{FS}}{2^B} \tag{2.1}$$

where V_{FS} is the peak-to-peak amplitude of the quantizer. With these two parameters in place, the quantizer transfer function of Figure 2.2 can be simply described as

$$y(kT_s) = x(kT_s) + e_Q(kT_s) \tag{2.2}$$

The noise, $e_Q(kT_S)$, is dependent on the amplitude of $x(kT_S)$ contrary to the usual signal-plus-noise models in communication theory [Gersho, 1977]. As seen in Figure 2.3, $e_Q(kT_S)$ is bounded in the following fashion:

$$-\frac{\Delta}{2} \le e_Q(kT_s) \le \frac{\Delta}{2} \tag{2.3}$$

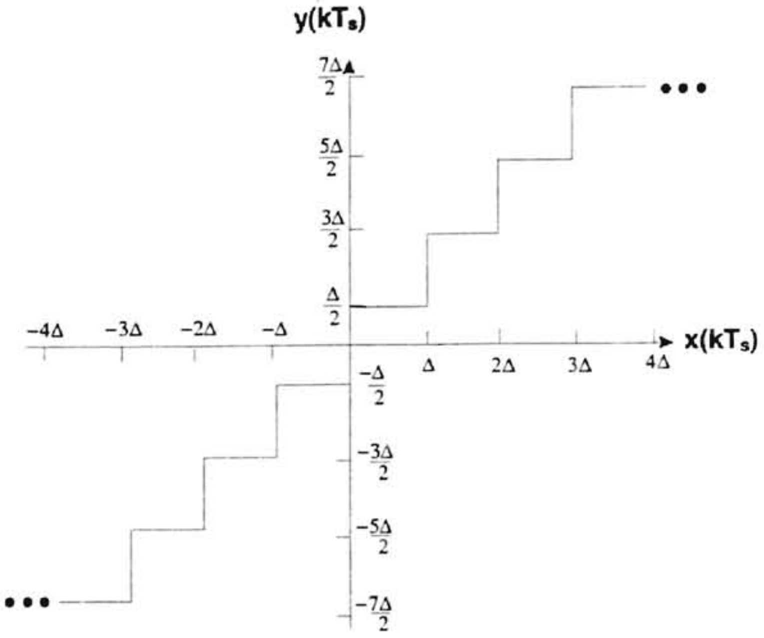


Figure 2.2 Transfer Function of a Mid-Riser Quantizer.

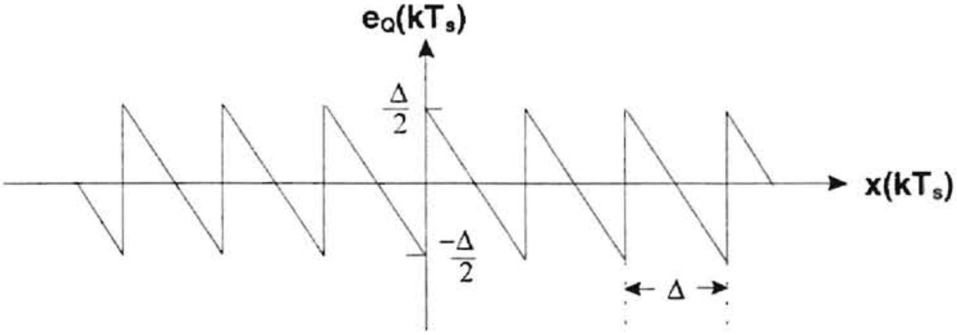


Figure 2.3 Quantizer Error Introduced by a Mid-Riser Quantizer.

Due to the non-linearity of the quantization noise, clearly revealed in Figure 2.3, a statistical approach is commonly used to investigate quantization effects. A statistical model, known as the Bennett noise model [Williams, 1992], treats $e_Q(kT_S)$ as a stationary white noise process. It has been developed assuming the following:

1. the quantizer input does not exceed the signal range of the quantizer, i.e. no quantizer clipping or overloading occurred,

2. the quantizer has a large number of quantization levels, Δ is small relative to the input signal level, and

3. the joint probability between two quantizer input signals is smooth.

These conditions are necessary to affirm that the quantization noise and quantizer input signal are uncorrelated, and that the quantizer is being used to its fullest potential without overloading. As a result, the statistical model has an approximately uniform probability distribution for the quantization error, i.e.,

$$\rho_e(e_Q) \approx \begin{cases} \dfrac{1}{\Delta}, & \dfrac{-\Delta}{2} \le e_Q \le \dfrac{\Delta}{2} \\ 0, & \text{otherwise} \end{cases} \qquad (2.4)$$

Quantizers using 5 or more bits, meaning 32 or more quantization levels, have been shown to satisfy the above conditions and to fulfill this probability density function [Rabiner, 1978; Van De Plassche, 1994]. With this statistical description, the quantizer can be modeled as a unity linear gain with additive white noise having a variance of

$$\sigma_e{}^2 = \int_{-\infty}^{\infty} e_Q{}^2 \cdot \rho_e(e_Q)\, de_Q = \frac{\Delta^2}{12}. \qquad (2.5)$$

Although many ADC systems may not achieve all the above required conditions, a white noise approximation of Bennett noise model furnishes the means for derivation of important ADC performance measurements.

2.1.1 Nyquist-Rate ADC Performance Measures

Two important ADC performance measures are the ADC's SNR and the ADC's useful signal range or dynamic range (DR). SNR is defined as the ratio

$$\mathrm{SNR} = \frac{S_{xx}}{S_{ee}} \qquad (2.6)$$

where, S_{xx} is the input signal power, and S_{ee} is the output noise power. The DR is defined as the ratio

$$\mathrm{DR} = \frac{S_{xx}|_{FS}}{S_{xx}|_{SNR=1}} \qquad (2.7)$$

where, $S_{xx}|_{FS}$ is the largest input signal power which does not permit quantizer clipping, and $S_{xx}|_{SNR=1}$ is the input signal power at which the SNR is unity.

For a Nyquist-rate ADC, the average output noise power is equal to the quantization noise variance. The input signal power can be assumed to be the input signal variance. For most performance calculations, a sine wave input to the ADC is assumed. If the input is $(V_{FS}/2)\cdot\sin(\omega t)$, then the output signal power excluding noise is $(V_{FS})^2/8$. Note that the maximum input voltage is limited to the maximum quantizer voltage, $V_{FS}/2$, to fulfill the first assumption of Bennett's Noise Model. Using these observations along with (2.5) and (2.6), the signal-to-quantization-noise-ratio of a Nyquist-rate ADC is

$$\mathrm{SNR}_{\mathrm{Nyquist}} = \frac{S_{xx}}{S_{ee}} = \frac{\sigma_x^{\,2}}{\sigma_e^{\,2}} = \frac{12\cdot\left(\frac{V_{FS}^{\,2}}{8}\right)}{\Delta^2} \tag{2.8}$$

Substituting (2.1) into (2.8), the SNR in decibels becomes

$$\mathrm{SNR}_{\mathrm{Nyquist}} = \frac{3}{2}\cdot 2^{2B} = (\,6.02\cdot B + 1.76\,)\ \mathrm{dB} \tag{2.9}$$

where, again, B is the number of bits in the quantizer. This equation implies that each additional bit added to the quantizer yields approximately 6 dB of SNR improvement in Nyquist-rate ADCs. The number of bits required for Nyquist-rate A/D conversion is a common benchmark for comparison of other A/D conversion methods. Hereafter, reference to the number of bits resolution for a particular system will be synonymous to the equivalent number of bits resolution for a Nyquist-rate A/D conversion.

The DR performance equation for a Nyquist-rate converter can be developed by noting that the largest input amplitude is basically $(V_{FS}/2)$ meaning the largest input signal power is $(V_{FS})^2/8$. The input signal power at which the SNR is unity is found by observing (2.8) at very small input amplitudes. Although this violates the Bennett model's rule that Δ should be small relative to the input signal level, it provides a useful performance measurement for further comparisons later in this thesis. So,

$$DR_{Nyquist} = \frac{S_{xx}|_{FS}}{S_{xx}|_{SNR=1}} = \frac{{V_{FS}}^2 / 8}{S_{ee}} = \frac{3}{2} \cdot \frac{{V_{FS}}^2}{\Delta^2} \tag{2.10}$$

Once again, using (2.1) by substitution into (2.10), the DR in decibels for a Nyquist-rate ADC is

$$DR_{Nyquist} = \frac{3}{2} \cdot 2^{2B} = (\ 6.02 \cdot B + 1.76\ )\ dB \tag{2.11}$$

This equation also implies that each additional bit added to the quantizer enhances the DR of a Nyquist-rate ADC by approximately 6 dB.

(2.9) and (2.11) are important measurements for Nyquist-rate ADCs. They show that the theoretical performance estimation for the SNR and DR of an ADC is only proportional to the number of bits or decision levels used in the quantizer. Due to quantizer limitations in the number of achievable quantization levels and in the consistency of decision making, there are definite SNR and DR restrictions in the possible amount of resolution a Nyquist-rate ADC may accomplish.

2.1.2 Nyquist-Rate ADC Performance Limitations

There are two primary drawbacks to most Nyquist-rate ADCs: resolution limitations and AAF implementation. Resolution limitations arise from the fact that A/D conversion resolution of a Nyquist-rate converter is directly proportional to the number of bits in the quantizer. Considering an ADC with a desired SNR of 95 dB, (2.9) states that the quantizer must have approximately 16 bits. This implies that the quantizer must have 2^{16} or 65536 quantization levels with a level separation of $\Delta \approx 31\ \mu V$, from (2.1) with $V_{FS} = \pm 1$ V. The quantizer is commonly comprised of (2^B-1) comparators which develop the digital signal by comparing the quantizer input signal to the (2^B-1) reference levels. Matching between any two of these comparators must be to the same resolution of the ADC, i.e., one part in 2^{16} or approximately 0.002%. Current MOS comparator technology permits a minimum comparison of roughly 10 mV due to the comparator's inherent offset voltage[Van De Plassche, 1994]. This is obviously greater than the Δ required for a 16-bit quantizer. Implementation of Nyquist-rate ADCs beyond approximately 10-bits of resolution is virtually unattainable in current process technologies without using some sort of calibration techniques, like laser trimming. In addition, as the number of bits in the quantizer grows to the 10-bit maximum, it is difficult to maintain an accurate step-size, Δ. This introduces greater integral non-linearity into the conversion process.

Another deficiency in these ADCs is the implementation of the AAF. To remove extraneous signals outside the ADC's band-of-interest, hereafter known as the ADC's pass-band, the AAF's stop-band for a Nyquist-rate ADC must begin at the Nyquist frequency, $f_S/2$. But, for all of the frequencies in the pass-band to be processed, the pass-band for the AAF must end at the same Nyquist frequency. Therefore, the AAF's transition-band must be very narrow and have a very steep response. This requires a complex filter with several precisely placed poles which is difficult to implement as an analog circuit.

Sensitivity to these ADC limitations can be partially overcome by a technique called oversampling. Today, sampling frequencies in very large scale integrated circuit (VLSI) technology have risen far above what is required for popular signal processing applications, such as for audio signals. The oversampling method uses this excess process bandwidth to enhance the DR and SNR of an ADC.

2.2 Oversampled ADCs

Oversampled ADCs sample an input signal in excess of the minimum required Nyquist rate. A block diagram of a typical oversampled ADC is presented in Figure 2.4. The input to the system is again a *real world* analog signal, x(t), which is continuous in time and in amplitude. This signal is pre-filtered by the analog, low-pass AAF. For the oversampled ADC, the filter uses a pass-band for frequencies within the band of interest, i.e., less than f_B. The transition band extends from the edge of the pass-band to the beginning of the stop-band at a frequency f_{SB}. After filtering, this band-limited signal, x'(t), is uniformly sampled at the ADC's f_S, which is greater than the Nyquist rate. Thus, the sampling transforms the signal into a discrete-time signal, $x(kT_S)$ where k is an integer. For a conventional oversampled ADC, the sampled signal is simply quantized into a fully digital signal, $y(kT_S)$. Chapter 3 discusses the replacement of the quantizer

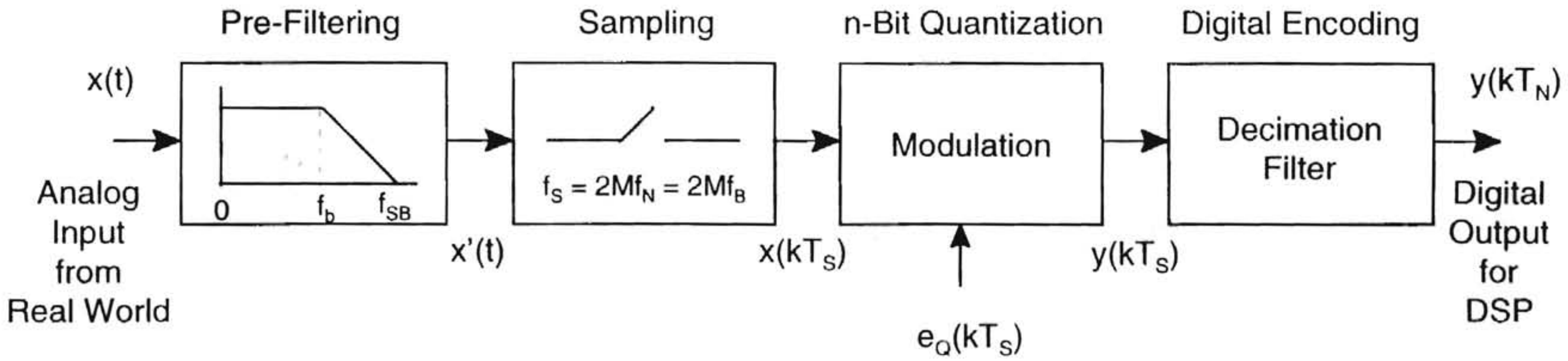


Figure 2.4 Block Diagram of an Oversampling ADC

with a modulator. Lastly, the signal is post-filtered, decimated back to the Nyquist rate. and encoded into binary code words to produce digital information, $y(kT_N)$, for further digital processing. Since the sampling rate of an oversampled ADC is higher than the Nyquist rate, a new term is introduced as the oversampling ratio, M, which is defined as follows:

$$M = \frac{f_S}{2 \cdot f_b} \tag{2.12}$$

where $2 \cdot f_B$ can be equated to the Nyquist frequency of the *input* signal (not to be confused with the Nyquist frequency of the *sampled* signal).

2.2.1 Oversampling ADC Enhancements

There are two primary advantages of oversampled ADCs over conventional Nyquist-rate ADCs. The first is a relaxation of the narrow transition band restriction for the AAF, as seen in Figure 2.5. Since the converter samples the input signal at M times the signal's Nyquist frequency, the transition band of the AAF can utilize a larger frequency range from f_B to $f_{SB} = (f_S - f_B) = f_B \cdot (2 \cdot M - 1)$. Although setting this f_{SB} allows for aliasing of transition band frequencies, pass-band frequencies are not affected. This loosening of the AAF restriction sanctions a lower order AAF with less need for accurate pole placement. However, during post-filtering of the digital signal, a digital filter with a narrow transition band is required during decimation. Implementing such a digital filter is definitely easier than an analog filter at the VLSI level. Digital design of such a filter is

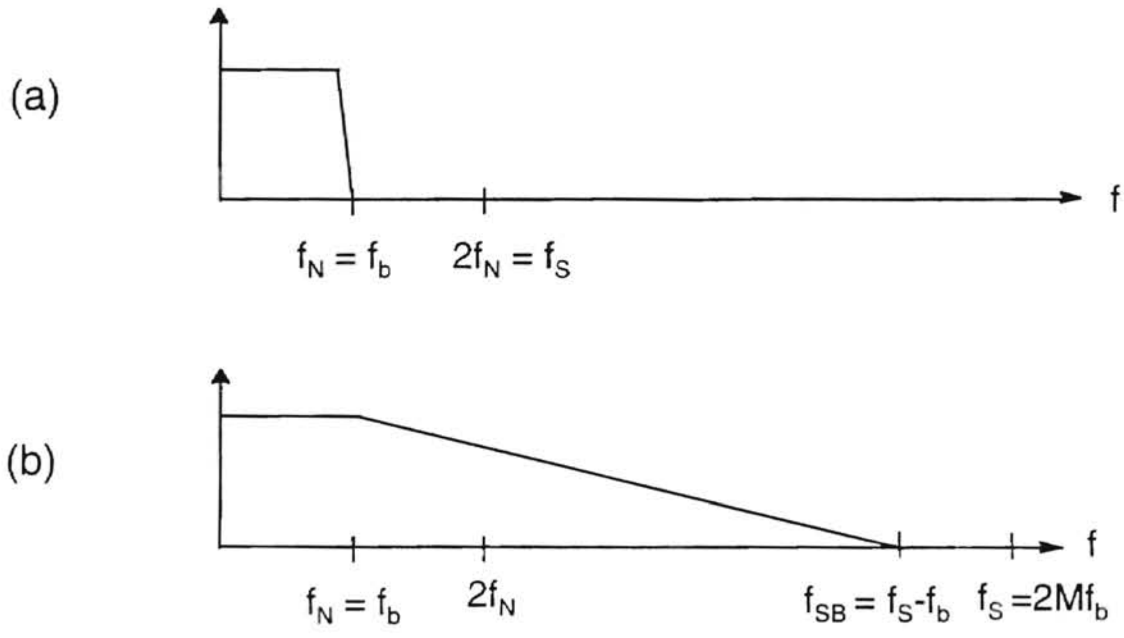


Figure 2.5 Frequency Responses of AAFs
(a) Nyquist-Rate, (b) Oversampled.

easily accomplished using a hardware development language, such as VHDL. In addition, by easing the AAF's complexity, lower fabrication costs and overall increased system reliability are attained.

A second enhancement of oversampled ADCs over Nyquist ADCs is an overall reduction of quantization noise throughout the frequency range of the pass-band. If the white noise approximation for the quantization noise is used, the quantization noise power at the quantizer output, $y(kT_S)$, will be evenly distributed throughout the sampling frequency bandwidth. The low-pass post-filtering attenuates the noise present outside the pass-band such that the noise power at the output, $y(kT_N)$, becomes

$$S_{ee} = \frac{{\sigma_e}^2}{f_S} \cdot \int_{-f_b}^{f_b} df = \frac{{\sigma_e}^2}{M} \qquad (2.12)$$

Again, assuming a system input of $(V_{FS}/2)\cdot\sin(\omega t)$, the signal-to-quantization-noise-ratio for the oversampling ADC improves to

$$SNR_{Oversampled} = \frac{S_{yy}}{S_{ee}} = \frac{{\sigma_x}^2}{{\sigma_e}^2/M} = M \cdot \frac{12 \cdot \left(\frac{{V_{FS}}^2}{8} \right)}{\Delta^2} \qquad (2.13)$$

Substituting (2.1) into (2.13) and realizing that the maximum input voltage is $V_{FS}/2$ for operation without quantizer clipping, the SNR becomes

$$SNR_{Oversampled} = M \cdot \frac{3}{2} \cdot 2^{2 \cdot B} = [\ 6.02 \cdot B + 1.76 + 10 \cdot \log_{10}(M)\ ]\ dB \qquad (2.14)$$

Thus, oversampling improves the Nyquist-rate ADC's SNR by M times. In addition, the DR is effected in the same manner yielding

$$DR_{Oversampled} = \frac{S_{xx}\big|_{FS}}{S_{xx}\big|_{SNR=1}} = \frac{{V_{FS}}^2 \big/ 8}{S_{ee}} = 12 \cdot M \cdot \frac{\left(\frac{{V_{FS}}^2}{8}\right)}{\Delta^2} \qquad (2.15)$$

Lastly, using (2.1) by substitution into (2.15), the DR for an oversampled ADC is

$$DR_{Oversampled} = M \cdot \frac{3}{2} \cdot 2^{2 \cdot B} = [\ 6.02 \cdot B + 1.76 + 10 \cdot \log_{10}(M)\ ]\ dB \qquad (2.16)$$

Thus, an oversampling ADC's DR is also greater than the Nyquist-rate ADC's DR by M times.

Over the past 20 years, oversampling ADCs have been preferred over Nyquist-rate ADCs precisely for these enhancements. Unfortunately, quantizer limitations in the number of achievable quantization levels and in the consistency of decision making still arise. Better performance can be accomplished using refined oversampling, noise-shaping techniques. One of these recently investigated methods is Sigma-Delta (ΣΔ) Modulation. This technique is used in ΣΔ ADCs to produce even greater conversion resolution than conventional oversampling ADCs with a reduced number of bits required in the quantizer.

Chapter 3 Sigma-Delta Modulators: Modeling and Simulation

During the last 15 years, there has been a growing interest in ΣΔ modulation and its implementation in ΣΔ ADCs. The theoretical noise-shaping properties of these converters sanction a higher resolution in A/D conversion over conventional Nyquist-rate and oversampled ADCs. Currently, inefficient commercial simulation packages have been available for theoretical analysis of ΣΔ designs. This has been an obstacle in speedy design and verification of a particular theoretical ΣΔ architecture. These simulators undertake investigation of ΣΔ architectures in either too much detail or too little detail. For example, SPICE is a well-known transistor-level circuit simulator. But due to its transient analysis, meaningful simulation of a simple ΣΔ modulator would take several hours or more. On the other hand, SwitCAP is a simulation package dealing with switched-capacitor circuits that are commonly found in ADCs. This simulator is known to take less time for simulation runs. But, it does not allow for the incorporation of some crucial ΣΔ circuit non-idealities, such as Op-Amp harmonic distortion effects and component mismatch errors. Two other simulators are known to have been developed at Stanford University, California [Boser, 1988] and at the Massachusetts Institute of Technology, Massachusetts [Chao, 1990] for the use of their design centers in the investigation of sampled-data systems. Therefore, the basis of the research behind this work was the development of a $\Sigma\Delta$ simulation package for the AAVDC that would be fairly efficient in simulation and could include the effects of important non-ideal behaviors. This simulator was developed by describing typical $\Sigma\Delta$ modulator components as modular block functions written in Matlab code. Matlab was chosen due to its advanced matrix manipulation properties. These properties are ideal for operating on intermediate node in sampled-data systems. Overall, the simulator, hereafter known as the $\Sigma\Delta$ toolbox, allows for rapid prototyping and useful insight into a variety of $\Sigma\Delta$ architectures.

Thorough investigation of the behavioral aspects of $\Sigma\Delta$ modulators was performed during the development of the toolbox and the subsequent incorporation of significant non-idealities. This chapter details these ideal and non-ideal behavioral aspects along with simulated performance measures for a few $\Sigma\Delta$ architectures. Initially, $\Sigma\Delta$ modulation, which is the fundamental approach for A/D conversion in $\Sigma\Delta$ ADCs, is described in detail. As in Chapter 2, ideal aspects are considered in the first section, i.e., only quantization errors are considered. With this resulting background, ideal theoretical performance measures are developed for typical cascaded $\Sigma\Delta$ architectures and contrasted with Nyquist-rate and oversampled ADC measures found in Chapter 2. An explanation of the primary components in typical $\Sigma\Delta$ modulator designs and their development as modular block functions in the $\Sigma\Delta$ toolbox ensues. Simulation results for a 1st-order $\Sigma\Delta$

modulator are then given to verify theoretical behavior and theoretical performance measures.

The second section discusses notable non-idealities that hamper performance of ΣΔ modulation. Particularly, these non-ideal behaviors are identified as integrator harmonic distortion errors, block component settling errors, switch charge injection errors, circuit common-mode errors, clock *jitter* errors, and circuit component mismatch errors. Once defined, the effects of these non-idealities are examined by use of the ΣΔ toolbox on the 1st-order modulator example. With the 1st-order ΣΔ modulator example complete, the remaining portion of the chapter investigates a couple of higher order modulators.

The third section analyzes a 3rd-order residual scaling ΣΔ architecture being developed by the AAVDC for NRaD. This architecture is stated to give 16 bits of A/D conversion resolution in a 62.5 MSPS band-of-interest performing at 8 times oversampling by utilizing digital error correction functions. By use of simulations from the ΣΔ toolbox, the architecture is enhanced and verified.

The last section considers a 3rd-order interpolative architecture based on a design developed by S. Nadeem [Nadeem, 1994]. This architecture refines the ΣΔ modulation process to shape the quantization noise power into a Chebyshev Type II form. By using this interesting concept, 18-bits of A/D conversion resolution is proposed for a band-of-interest of 10 kSPS performing at 64 times oversampling. Again, the ΣΔ toolbox is used to aid the initial design of the architecture and to verify the expected A/D conversion performance.

The ΣΔ toolbox simulations throughout the chapter briefly show the potential for the ΣΔ toolbox. As a whole, the simulator proved to be an excellent tool in the rapid design and prototyping of ΣΔ architectures being designed and developed by the AAVDC.

3.1 Sigma-Delta Modulation

ΣΔ modulation is an oversampling, noise-shaping A/D conversion method that uses feedback to enhance the effective conversion resolution. The increase in A/D conversion resolution is achieved by replacing the quantizer in an oversampled ADC with a ΣΔ feedback modulation network. This method, which induces noise-shaping, alters the uniform behavior of the quantization noise throughout the oversampled frequency spectrum into a non-linear manner.

Noise-shaping is a technique used where quantization noise is attenuated in the modulator's pass-band and increased outside the pass-band. Figure 3.1 displays the noise-shaping characteristic in relation to Nyquist-rate and oversampled quantization noise attributes. A qualitative view of quantization noise power is shown for each of the A/D conversion methods.

As discussed in chapter 2, the quantization noise power for a Nyquist-rate ADC displayed from dc to the end of the pass-band, which is also the Nyquist frequency, occurs from simple quantization. Oversampling is achieved by sampling at a higher frequency than the Nyquist rate. This allows for the same quantization noise power to be spread over a larger frequency range. In-band quantization noise is reduced in oversampled ADCs by decimation and low pass, post-filtering of frequencies outside the pass-band. Thus, this digital filtering effectively removes much of the quantization noise power. Oversampling causes a distinct reduction by M in quantization noise power within the pass-band in contrast to the Nyquist-rate ADC. The resolution of an oversampled, noise-shaping ADC is increased over conventional oversampling by effectively shifting quantization noise power to higher frequencies in a non-linear fashion. Again, the digital decimation filtering that follows this refined oversampled modulation process greatly attenuates the higher frequencies. Consequently, this removes a larger amount of quantization noise power from the pass-band in contrast to both the Nyquist-rate and conventional oversampled ADCs. This fact alone obliges the investigation of ΣΔ modulators. The remainder of this chapter distinguishes the development of ΣΔ modulators, their interesting improvements to standard A/D conversion, and their implementation within the ΣΔ toolbox.

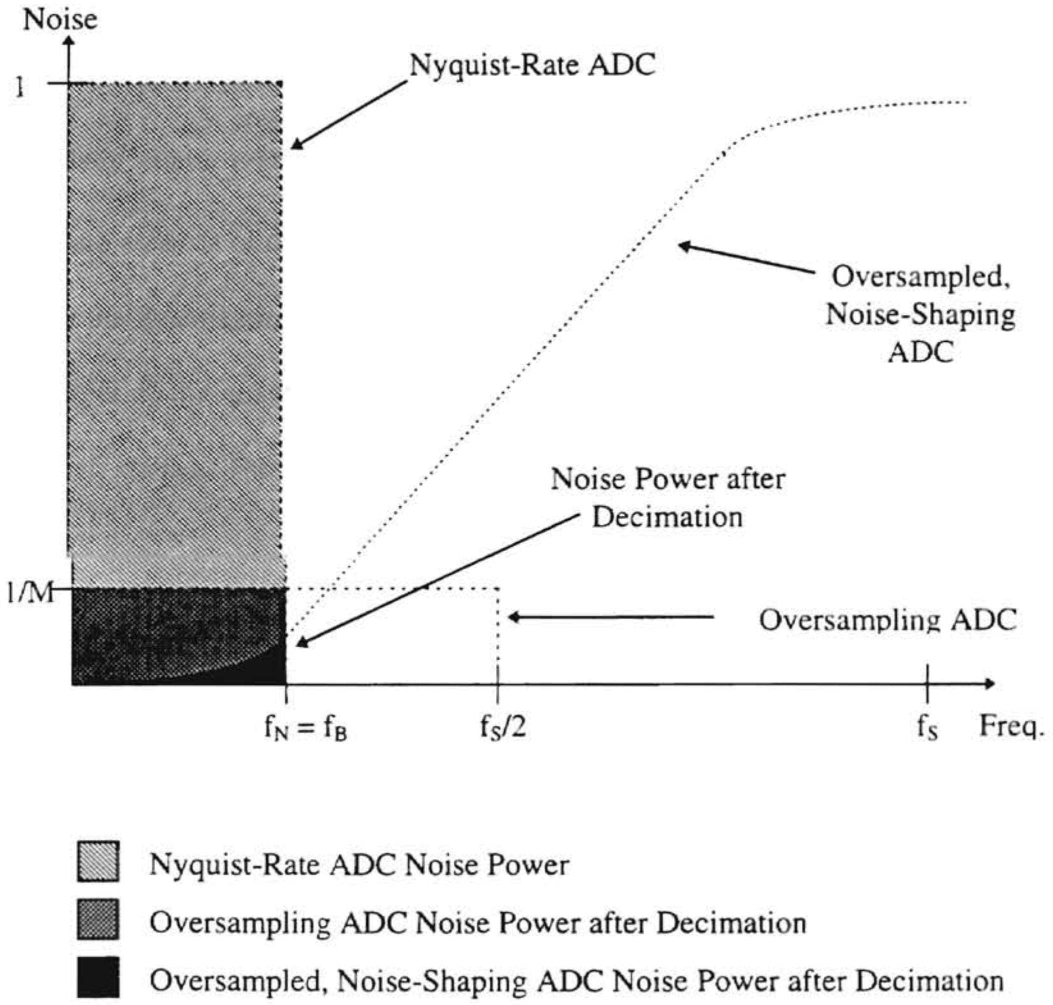


Figure 3.1 Quantization Noise Power Comparisons between Nyquist-Rate, Oversampling, and Oversampled, Noise-Shaping ADCs.

3.1.1 Behavioral Modeling of Sigma-Delta Modulators

There are a variety of high gain functions that may be implemented as ΣΔ modulators. The classical transfer function for a cascaded ΣΔ modulator is comprised of a linear combination of integration stages that differentiate the quantization noise. This particular implementation is well suited for VLSI circuits. A block diagram of a typical ΣΔ modulator circuit that satisfies such a transfer function is shown in Figure 3.2. The forward path consists of delaying integrators followed by a B-bit quantizer. For this ideal case, only the quantization error, $e(kT_S)$ introduced by the quantizer is considered. The digital output of the quantizer, $y(kT_S)$, is a thermometer encoded estimate of the analog input signal, $x(kT_S)$. The digital output signal is fed back through a digital-to-analog converter (DAC) to be subtracted from the input signal as well as the outputs of each subsequent integrator. The number of integrators used in the design defines the

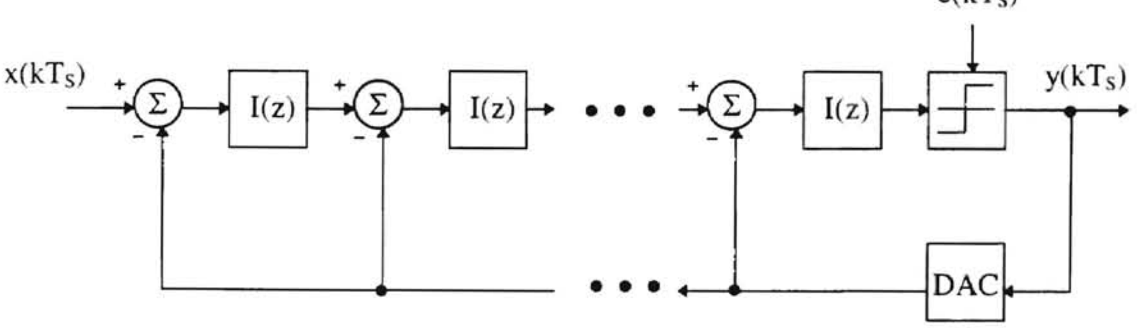


Figure 3.2 A Typical ΣΔ Modulator Block Diagram.

modulator order, L, of a ΣΔ modulator. Since the z-domain transfer function of a delaying integrator is

$$I(z) = \frac{z^{-1}}{1 - z^{-1}} \tag{3.1}$$

and assuming an ideal DAC, the baseband output of a cascaded L^{th}-order ΣΔ modulator in the z-domain is

$$Y(z) = z^{-L} \cdot X(z) + (1 - z^{-1})^{L} \cdot E_{Q}(z) \tag{3.2}$$

The output is simply a L^{th} sample delay of the input with a L^{th} order quantization noise difference term. This differencing of the noise causes the high-pass, noise-shaping characteristic of ΣΔ modulation. Systems utilizing a higher modulator order perform a higher ordered differencing operation on the quantization noise, thus providing a stronger attenuation of quantization noise at lower frequencies. Further analysis of this equation provides quantitative theoretical performance measures for cascaded ΣΔ modulation.

3.1.2 Sigma-Delta Performance Measures

Performance measures for ΣΔ modulation are usually in terms of the system's output SNR. For the cascaded ΣΔ modulator, the quantization noise power within the pass-band is found from the quantization noise contribution to (3.2), i.e., the noise transfer function (NTF). The NTF for an Lth-order ΣΔ modulator is

$$H_E(z) = (1 - z^{-1})^L \tag{3.3}$$

The spectral distribution of the quantization noise after noise-shaping is the product of the NTF spectral density and the quantization error spectral density introduced by the quantizer. So, the quantization noise power spectral density is found using (3.3) and converting to the Fourier domain:

$$S_{ee}(f) = \left[\left|H_E(z)\right|^2_{z=\exp\left(\frac{j \cdot 2 \cdot \pi \cdot f}{f_s}\right)} \right] \bullet \left[\frac{\sigma_e^2}{f_S} \right] \tag{3.4}$$

Substitution of (3.3) into (3.4) and using the white noise approximation of Bennett's noise model gives the following equation for the power spectral density of the quantization noise at the output of a cascaded ΣΔ modulator:

$$S_{ee}(f) \approx \left(2 \cdot \sin\left(\frac{\pi \cdot f}{f_S} \right) \right)^{2 \cdot L} \bullet \left[\frac{\sigma_e^2}{f_S} \right] \tag{3.5}$$

Integration over the entire pass-band gives the quantization noise power over the band-of-interest. This is also approximately the quantization noise power found at the output of the ADC after decimation. Thus, the inband quantization noise power is

$$P_Q = \int_{-f_B}^{f_B} S_{ee}(f)\, df \approx \sigma_e^2 \cdot \frac{\pi^{2L}}{(2 \cdot L + 1)} \cdot \frac{1}{M^{2 \cdot L + 1}} \tag{3.6}$$

This average noise power floor is an essential value to determine the modulator's SNR.

As in chapter 2, the cascaded ΣΔ modulator's SNR can be found assuming a modulator input of (V_{FS} /2)·sin(ωt). Since the output signal power excluding noise is $(V_{FS})^2/8$, the SNR for an oversampled, Lth-order cascaded ΣΔ modulator is

$$SNR_{\Sigma\Delta} = \frac{V_{FS}^{2}}{8} \Big/ P_Q = \frac{3}{16} \cdot \frac{(2 \cdot L + 1)}{\pi^{2 \cdot L}} \cdot M^{2 \cdot L + 1} \cdot 2^{2 \cdot B} \tag{3.7}$$

where again M is the oversampling ratio and B is the number of bits in the quantizer. An interesting item to note is that the SNR is proportional to the 2L+1 power of the oversampling ratio, M, which is a much greater improvement over conventional oversampling. Due to this strong relationship to M, the number of bits required to attain a prescribed SNR is much less for ΣΔ modulators than Nyquist-rate or conventional oversampled ADCs. Figure 3.3 shows these benefits by plotting the SNR versus M for ΣΔ modulators of orders L = 1, 2, and 3 and quantization bits, B = 1. The number of bits required for a conventional Nyquist-rate ADC to achieve the same SNR is shown on the right of the graph. Even by using a ΣΔ modulator containing a 1-bit quantizer, comparatively high resolution is attained when compared to a Nyquist-rate converter. In addition, an increase in the number of bits in the quantizer does have the same effect in ΣΔ modulation as in both Nyquist-rate and conventional oversampled converters. Equation (3.4) showed that the quantization error spectral density plays a distinct role in the quantization noise power spectral density. Increasing the number of bits in the quantizer reduces the quantization noise power by 6 dB per additional bit. Thus, each curve in Figure 3.3 is effectively raised by an additional 6 dB for each additional bit

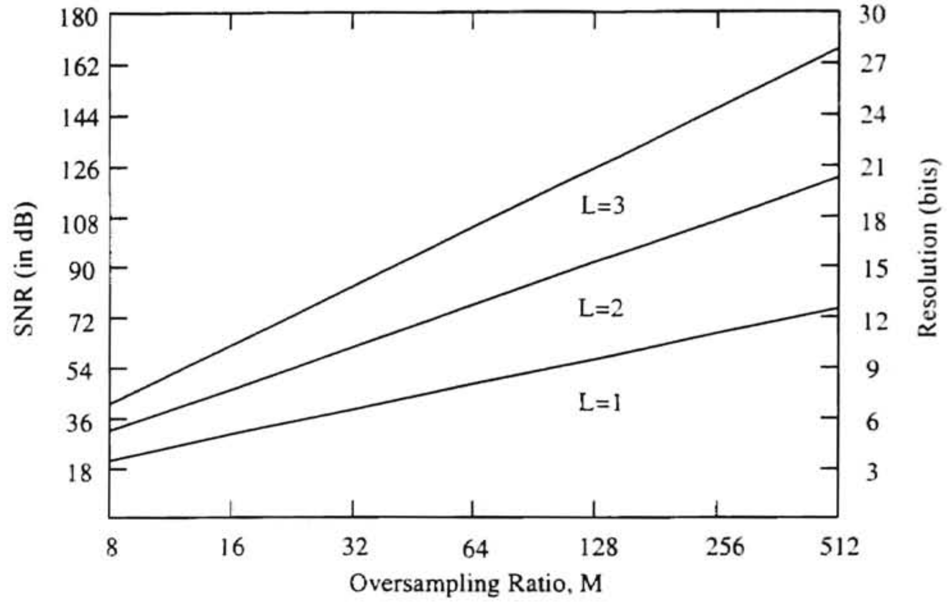


Figure 3.3 Cascaded ΣΔ Modulator SNR vs. M for a 1-Bit Quantizer.

added to the quantizer. Figure 3.4 shows this by plotting the SNR versus M for ΣΔ modulators of orders L = 1, 2, and 3 and quantizer bits, B = 4. Therefore, by increasing the number of bits in the quantizer, a lower oversampling ratio may be used to attain an equivalent 1-bit modulator A/D conversion resolution. This property gives a simple method for the reduction of the digital power consumption in low-power ΣΔ ADC designs. Thus, simple cascaded ΣΔ architectures apparently perform A/D conversion more effectively than Nyquist-rate or conventional, oversampled ADCs.

It is difficult to gain an intuitive feel for how ΣΔ modulation can actually attain such high resolution from such a small number of quantizer bits. In the next section, the

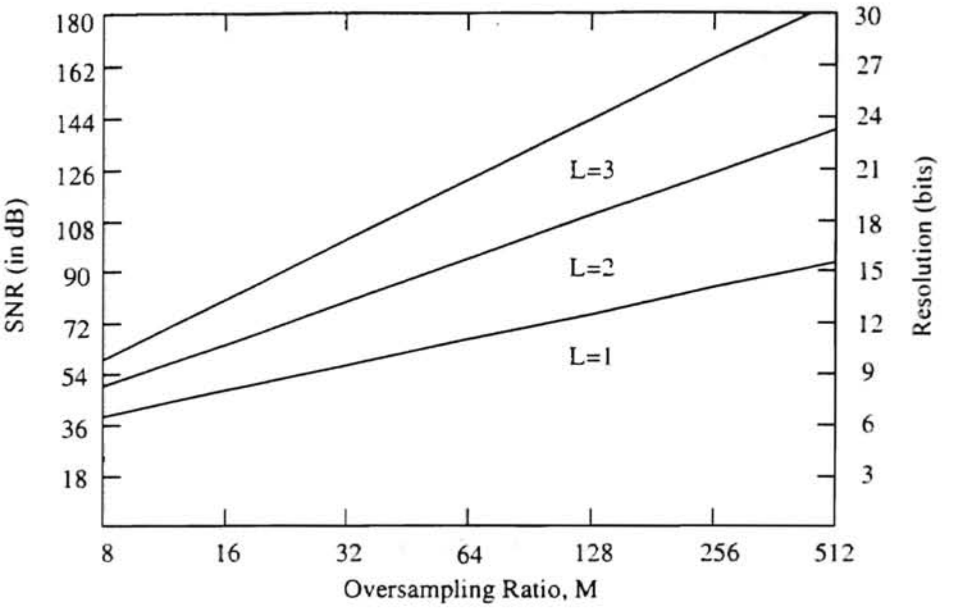


Figure 3.4 Cascaded ΣΔ Modulator SNR vs. M for a 4-Bit Quantizer.

above theoretical observations are investigated and verified by use of the ΣΔ toolbox. After a discussion of the basic building blocks in the ΣΔ toolbox, a 1st-order ΣΔ modulator is developed to gain some insight in the ΣΔ modulation process.

3.1.3 *Implementation of ΣΔ Modulators using the ΣΔ Toolbox*

As seen in Figure 3.2, the basic building blocks for a ΣΔ modulator are delaying integrators, summation nodes, single or multi-bit quantizers, and DACs. In addition, higher order modulators also use amplifiers or gain stages to modify the modulator's noise-shaping characteristics. In this section, it is helpful to consider a 1^{st}-order modulator example when discussing these basic components. A block diagram of a 1^{st}-order ΣΔ modulator is shown in Figure 3.5. The ΣΔ toolbox implements each of these basic blocks in an individual, modular fashion. Before the toolbox can be used, an architecture must be theoretically developed in a block diagram form using the above mentioned building blocks. Once a ΣΔ architecture is developed, an architecture file is written in Matlab which specifies the vital components for each block and the basic clocking sequence of blocks. The ΣΔ toolbox iterates a simple loop for each input sample by calling each block function in the succession specified by the architecture file. In typical architectures and in the 1^{st}-order example of Figure 3.5, the beginning of the iterated loop starts with the input voltage to the system and its transition into the integrator.

In a typical circuit implementation, the summation node and the integrator are treated as a single entity. The ΣΔ toolbox models them in somewhat the same manner. A clocked, differential switched-capacitor, MOSFET operational transconductance amplifier (OTA) continuous-time integrator implementation is used for integration. A basic diagram for a single-sided switched-capacitor continuous-time MOSFET integrator, which the ΣΔ toolbox uses as a model, is shown in Figure 3.6. This circuit has three basic parts which need to be defined in the ΣΔ toolbox. These are the OTA's open-loop gain, A_{vol}, the sampling capacitor, C_{si}, and the OTA's integrating capacitor, C_{int}. The switched-capacitor circuit uses a two-phase non-overlapping clock. During the sampling,

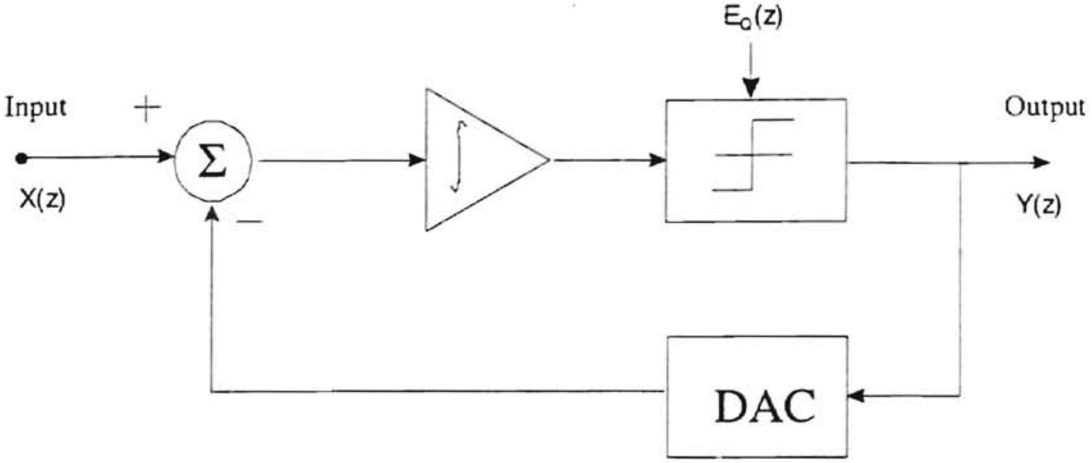


Figure 3.5 Block Diagram of a 1st-Order ΣΔ Modulator.

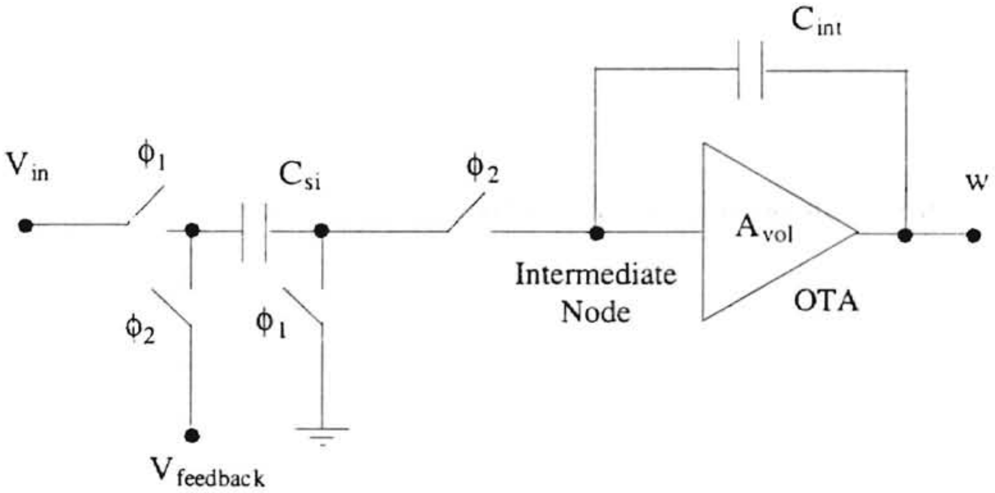


Figure 3.6 Switched-Capacitor Continuous-Time Integrator.

phase 1, ϕ_1, of a clock cycle the charge from the input, V_{in}, is transferred to C_{si}. During the integration, phase 2, ϕ_2, a charge proportional to the difference of the voltage being fed-back from the DAC, V_{DAC}, and V_{in} is mirrored onto C_{int} for integration. The integrator output, $w(kT_S)$, is modeled as:

$$w(k \cdot T_S) = \frac{A_{vol}}{1 + A_{vol}} \cdot w((k-1) \cdot T_S) + \frac{C_{si}}{C_{int}} \cdot \left[v_{in}(k-1) \cdot T_S) - v_{feedback}(k-1) \cdot T_S)\right] \quad (3.8)$$

C_{si}/C_{int} specifies the closed-loop gain of the integrator. An important design consideration is that in most cases this closed-loop gain should ensure that the integrator output is within the full-scale voltage of the quantizer, V_{FS}, to prevent quantizer clipping. The leakage of this integrator is identified in equation (3.8) to be approximately $1/A_{vol}$ of the integrated value which is lost in each clock cycle. This leakage is on the same order as that in a continuous-time integrator. Overall, equation (3.8) suffices as a model to implement the integration operation in the ΣΔ toolbox.

The ΣΔ toolbox separates (3.8) into an integrator block function and a summation node block function. This allows for the intermediate node, in addition to the input, feedback, and output nodes to be viewed after the simulation. The differential behavior of the integrator is simulated by simply copying the single-sided circuit using a negative version of the input signal.

During ϕ_2, the integrator output, w, is fed to the quantizer for conversion from an analog signal to a digital signal. A uniform mid-riser quantizer was implemented in the ΣΔ toolbox. The A/D conversion is accomplished in hardware by using a string of comparators, as shown in Figure 3.7. The number of decision or quantization levels define the number of comparators. For a B-bit quantizer, there are (2^B-1) comparators which develop the digital signal by comparing the quantizer's analog input signal to the (2^B-1) reference levels. The toolbox accomplishes this conversion by a simple 'if-then' comparison between the input to the quantizer and the appropriate quantization level which are uniformly positioned between $\pm V_{FS}/2$. This produces a thermometer encoded digital output of (2^B-1) lines which is subsequently fed back into the DAC. This digital data is also the output of the ΣΔ modulator. It may be further encoded and decimated by a digital post-processor as mentioned in Chapter 2.

The thermometer encoded, digital data is fed back through a DAC to the analog summation node in order to complete the ΣΔ modulation loop. A basic schematic diagram of the DAC is presented in Figure 3.8. The (2^B-1) lines of output from the quantizer is used to create an equivalent analog signal. This is accomplished by charging the appropriate number of feedback capacitors, C_{fb}, and combining their voltages into a 1-line, analog DAC output. During the first portion of sampling ϕ_1, all the capacitors are discharged to ground. During the second portion of phase ϕ_1, all the capacitors except for the initial C_{fb} are pre-charged to the appropriate DAC reference voltages, $\pm V_{ref\_DAC}$. Finally during the conversion phase, ϕ_2, certain capacitors depending on the quantizer output information, V_{qout}, are connected to ground while the others remain at $\pm V_{ref\_DAC}$. While converting, the charge is redistributed over the capacitors and an equivalent analog voltage is fed through the voltage follower as the output of the DAC, V_{DAC}. The DAC

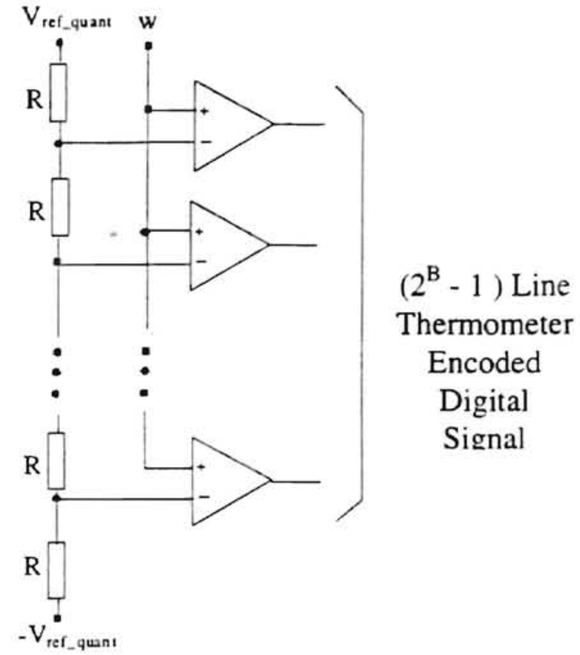


Figure 3.7 Basic B-Bit Quantizer Schematic.

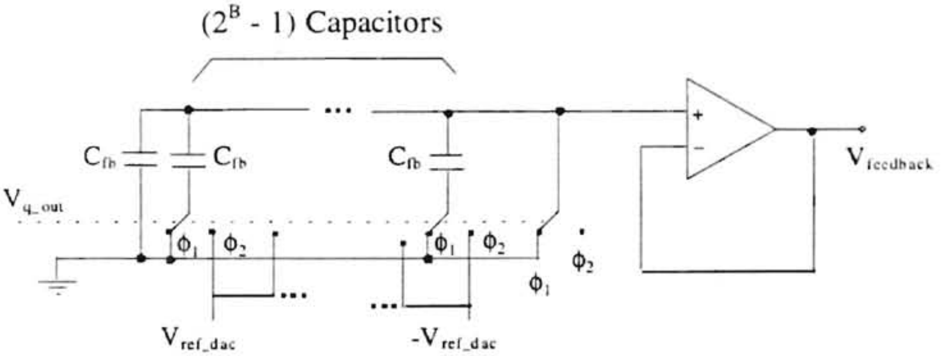


Figure 3.8 Basic B-Bit DAC Schematic.

conversion is implemented in the ΣΔ toolbox by scaling the output of the quantizer by the following formula:

$$V_{DAC} = \frac{V_{FS}}{2} \cdot \frac{C_{si}}{C_{fb}} \cdot \frac{2}{2^B - 1} \cdot V_{qout} \tag{3.9}$$

This accounts for the redistributed charge in all the C_{fb} terms and scales V_{DAC} appropriately to match V_{FS}.

The last of the building blocks considered in the ΣΔ toolbox is an amplification block. This block is used at various nodes in many higher order ΣΔ architectures which try to optimize the noise-shaping process. The amplification block is similar to the integrator block toolbox function as seen in Figure 3.9. The architecture file specifies the essential values for the amplifier: the amplifier's open-loop gain, A_{vol}, the input resistor, R_{si}, and the amplification resistor, R_{amp}. There are essentially two differences between the amplifier and the integrator. One is that a delay is not required since a switched-

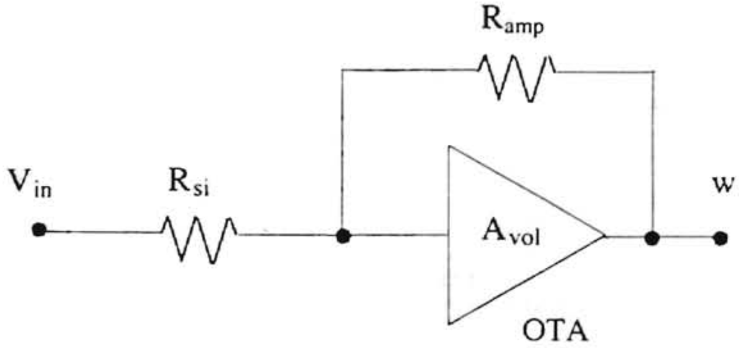


Figure 3.9 Basic Amplifier Schematic.

capacitor arrangement is not present. The second being that the closed loop gain is attained by a resistor ratio, R_{si}/R_{amp}, specified in the architecture file rather than a capacitor ratio. In typical analog VLSI amplifier implementaions the amplifier arrangement of Figure 3.9 is not commonly used. Amongst other forms of amplification the following methods are used, single-stage class A, two-stage class A, and single-stage class AB amplifiers [Williams, 1993]. In addition, simple integer fraction switched-capacitor ratios are commonly used as gain stages, as will be seen in Section 3.4. Nevertheless, the model in Figure 3.9 provides a more intuitive method including the basic features required to accurately implement amplification at a block level within the ΣΔ toolbox.

With each of these blocks defined, a simulation is performed by using the desired architecture file. The beginning of the ΣΔ toolbox architecture file states a variety of required values for the overall system. These include the following constants: the sampling frequency, freq_samp, the desired input frequency, freq_des, the oversampling ratio, over_ratio, the number of cycles desired, num_cycles, the number of quantizer bits, num_bits, and the maximum input voltage, max_in. The following formula is used to determine the desired number of samples:

$$\text{Nuber of Samples} = \frac{\text{freq\_ samp}}{\text{freq\_ des}} \cdot \text{num\_ cycles} \qquad (3.10)$$

These values are passed onto an input generation function which generates a specified input voltage waveform for the number of desired samples. These input samples are subsequently operated on by each of the sequentially specified block functions in the architecture file's iteration loop. The loop is iterated for the specified number of samples while saving the variety of node values for each iteration.

Once a simulation of an architecture is completed, the toolbox allows for a performance evaluation of the resulting, undecimated, digital data. A variety of intermediate nodes are available for plotting to gain insight in the behavior of the desired architecture. In addition, the SNR for the system may be calculated by using a post-simulation function. This function computes the *total* noise power in the same manner as described in equations (3.3) through (3.7) above. The difference is that the simulated combination, $H_E(z)*E_Q(z)$ is used to calculate the total noise power by noting that the *noise-shaped* term, NSTerm, is:

$$\mathrm{NSTerm}(z) = H_E(z) \cdot E_Q(z) = Y(z) - H_X(z) \cdot X(z) \qquad (3.11)$$

This noise-shaped term actually accounts for other noise contributions beyond quantization noise, considered in the next section. Before simulation, the right-hand side of (3.11) must be converted to its sampled time-domain representation in the architecture file. Then, the resulting NSTerm can be used to find the shaped noise contribution in the output of the desired modulator. The noise power spectral density is calculated by the ΣΔ toolbox using the Fast-Fourier Transform, FFT, on the simulated NSTerm:

$$S_{ee}(f) = \left[\left. |\mathrm{NSTerm}(z)|^2 \right|_{z=\exp\left(\frac{j \cdot 2 \cdot \pi \cdot f}{f_s}\right)} \right] \tag{3.12}$$

The calculation continues by finding the total noise power by:

$$P_{NS} = 2 \cdot \int_0^{f_B} S_{ee}(f)\, df \tag{3.13}$$

Lastly, the toolbox function gives the simulated SNR by taking the ratio of the simulated output signal power to the simulated P_{NS}. An initial view of the ΣΔ toolbox's performance measurement capabilities is accomplished by simulating the 1st-order ΣΔ modulator described in Figure 3.5.

The following simulations use a sampling rate of f_S = 1 GSPS, C_{si} = 30 pF, C_{int} = 32 pF, and a unit C_{fb} = 0.4 pF. First, consider the modulator utilizing a 1-bit quantizer with a V_{FS} = ± 1 Volt. If a ramp voltage is input into the system at 32 times oversampling, M = 32, the simulator predicts the output as in Figure 3.10. Notice the local averaging characteristic of ΣΔ modulation. That is, while the input is at ±1 V., the modulator output is also ±1 V. As the input rises to 0 V., the output begins to oscillate around 0 V. In fact, the local average of the output waveform is zero when the input voltage crosses 0 V. This behavior is common in ΣΔ modulators utilizing a 1-bit quantizer[Candy, 1992].

If a sine wave is input into the same modulator at M=8 and B=1, the theoretical SNR from (3.7) predicts a ratio equal to 20.67 dB. After simulating the modulator with

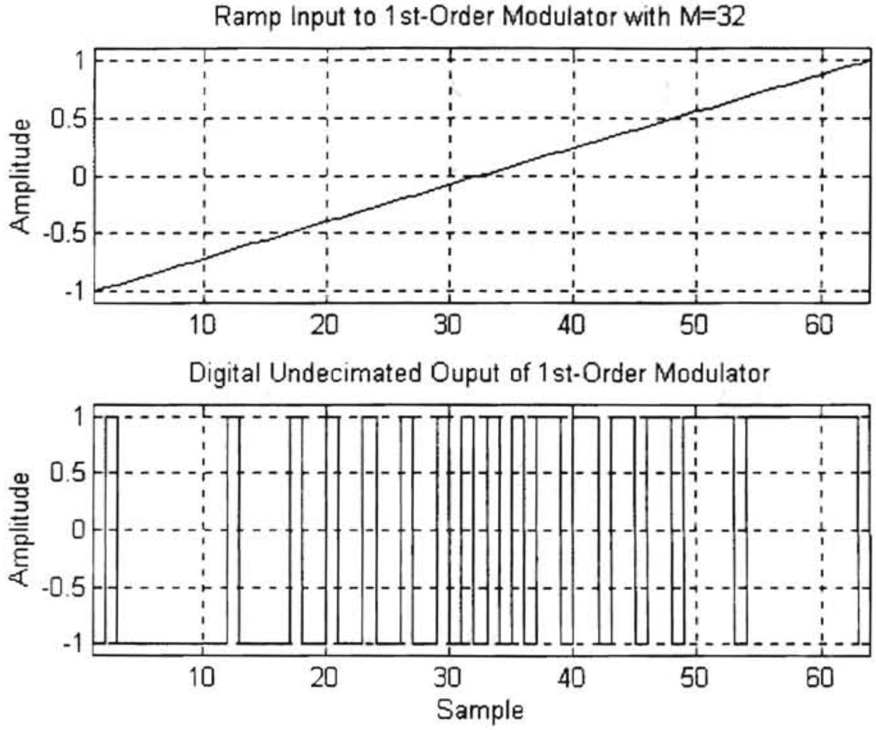


Figure 3.10 Output of a 1st-Order ΣΔ Modulator with a 1-Bit Quantizer at M=32 with a Ramp Input Voltage.

these new specifications using 32,000 samples, the $\Sigma\Delta$ toolbox gave the simulated SNR of 20.95 dB, shown in Figure 3.11. The upper graph gives the simulated inband quantization noise power spectral density. Notice the noise-shaping characteristic for this 1st-order $\Sigma\Delta$ modulator. The quantization noise power has definitely been shaped to attenuate the noise power in the passband. The lower graph gives the simulated output spectrum for the modulator. A major spike occurs at f_B, which was the frequency of the input sine wave. The amplitude of the spike has been slightly distorted by the use of a Hamming window for better a FFT. Not windowing the time-domain data before using the FFT has the same effect as using a rectangular window. A rectangular window does not have sufficient sidelobe attenuation for calculation of the noise power. In later examples, the $\Sigma\Delta$ toolbox uses a Kaiser window since it provides a much lower sidelobe attenuation than the default rectangular window[Oppenhiem, 1989]. This is important for viewing the spectral content of $\Sigma\Delta$ modulators having a resolution nearing 20 bits. Therefore, the graphs are meant more as qualitative views for $\Sigma\Delta$ modulator behavior. Quantitative results are shown to the right of the bottom graph. These figures are fairly consistent with the theory.

Next, consider the same modulator operating at a higher oversampling rate of $M = 32$. Theory from (3.7) dictates that the SNR should be 38.73 dB. Figure 3.12 gives the simulated performance measurements again using 32,000 samples. Once again, noise-shaping is seen in the upper plot of the noise power spectral density. The output of the modulator and the SNR is shown in the lower plot. An interesting observation is that the general shape of the output spectrum is similar to Figure 3.11. The higher SNR comes

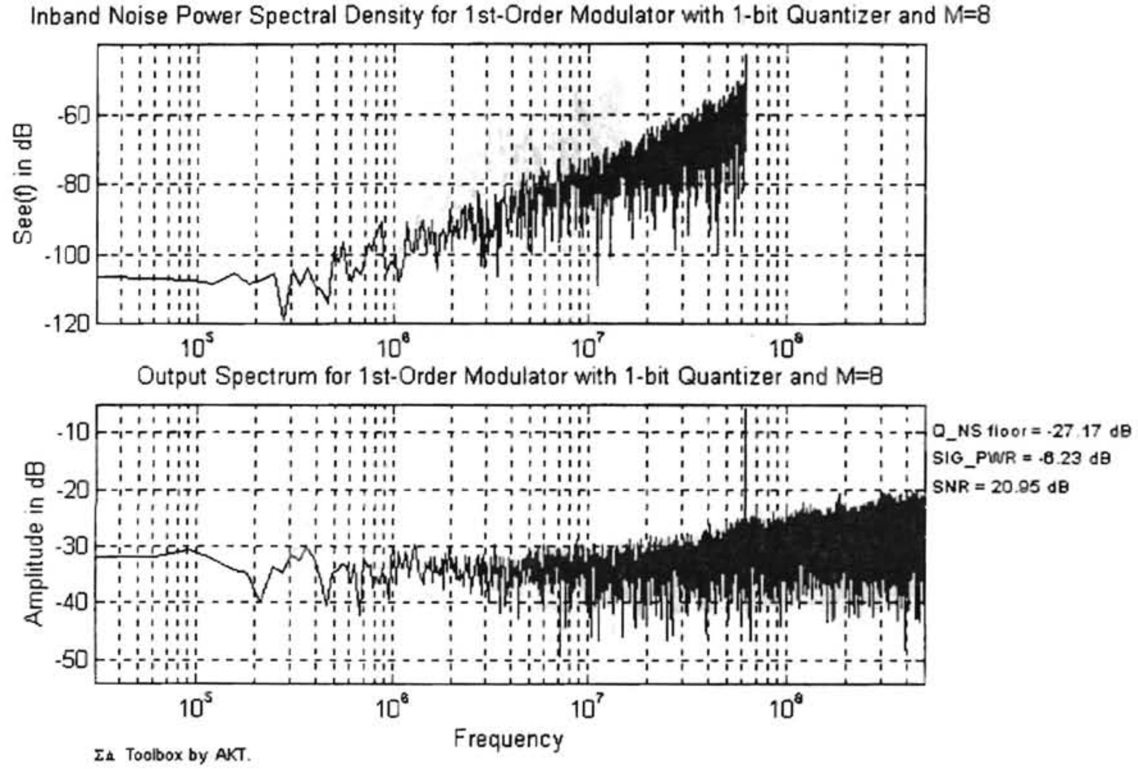


Figure 3.11 Simulated Performance Specifications of a 1^{st}-Order ΣΔ Modulator with a 1-Bit Quantizer at M=8 with a Sine Input Voltage.

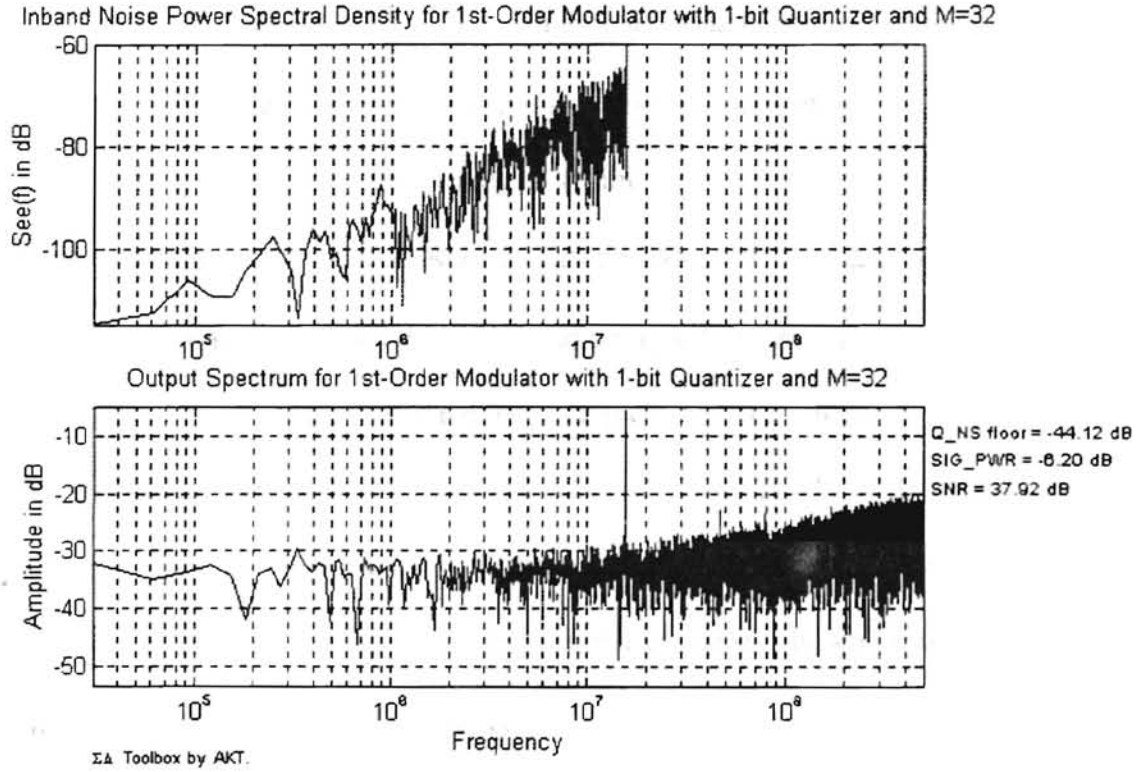


Figure 3.12 Performance Specifications of a 1st-Order ΣΔ Modulator with a 1-Bit Quantizer at M=32 with a Sine Input Voltage.

solely from a higher oversampling ratio. The simulated SNR of 37.92 dB is again fairly close to the theoretical behavior. The ΣΔ toolbox seems to give respectable simulations for the ΣΔ modulator using a 1-bit quantizer at different oversampling rates. Before continuing, a quantitative comparison between the oversampled ΣΔ modulation technique and conventional oversampling can be made.

The SNR for a 1st-order ΣΔ modulator using a 1-bit quantizer was seen to be about 21 dB with 8 times oversampling. Using equation (2.14), a conventional, oversampled technique also performing at 8 times oversampling would require a 2-bit quantizer to achieve the same SNR as the ΣΔ modulator. If the ΣΔ modulator is clocked at a higher 32 times oversampling and still utilizing a 1-bit quantizer, a SNR of about 39 dB was seen. But for a conventional, oversampled ADC to achieve a comparable SNR, its oversampling ratio would have to be increased to about M = 330 using the 2-bit quantizer or the number of bits in its quantizer would have to be increased to about 4-bits performing at M = 32. So again, these quantitative values show that the ΣΔ modulator definitely has distinct advantages over conventional, oversampled ADCs.

Continuing with the simulator example of the 1st-order ΣΔ modulator in Figure 3.5, consider the performance effects of using a multi-bit quantizer. Figure 3.13 demonstrates the use of a 4-bit mid-riser quantizer. The lower plot is the quantized, digital output for the ΣΔ modulator with a 1 V. sine wave input at 8 times oversampling. The uneven behavior at the modulator output is due to the mid-riser quantizer's property of having no zero level and also due to the low oversampling ratio. Nevertheless, the simulator produces an output that has the one sample delay that was expected from the

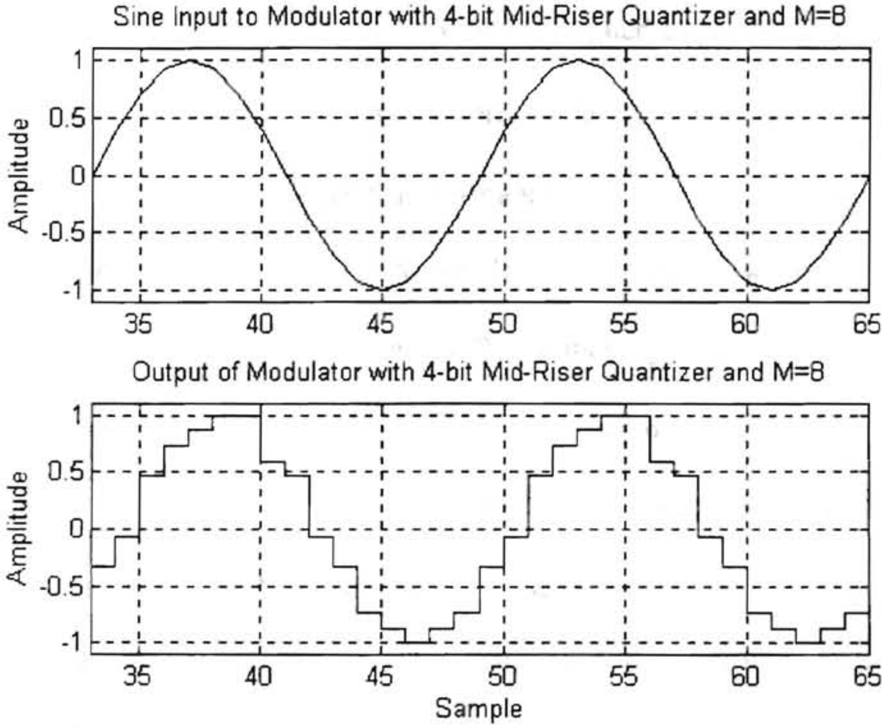


Figure 3.13 Output of a 1^{st}-Order ΣΔ Modulator with a 4-Bit Quantizer at M=8 with a Sine Input Voltage.

signal transfer function, STF, in (3.2).

Figure 3.14 gives the simulated performance characteristics for this architecture using a multi-bit quantizer using 32,000 samples. The reason for the relatively flat noise power spectral density in the upper graph is due to the small quantization step size, Δ, which accompanies the greater number of bits in the quantizer. The flattening arises when the small Δ contributes to the $E_Q(z)$ when multiplied by the NTF, $H_E(z)$, in (3.11). Since the $\Sigma\Delta$ toolbox uses the resulting NSTerm rather than just the NTF when plotting performance measures, the prominent noise-shaping characteristic produced by $H_E(z)$ is not easily seen, although it still exists. Again, the lower graph shows the output spectrum of the $\Sigma\Delta$ modulator. Theoretically from (3.7), the modulator should produce a SNR of 38.73 dB. The simulator predicts a relatively equal SNR of 39.12 dB. Note that this is approximately the same SNR predicted in Figure 3.11, i.e., the SNR for the $\Sigma\Delta$ modulator using a 1-bit quantizer and M = 32. An increase in the number of quantizer bits has allowed for the reduction of the sampling rate by 4 times, which corresponds to a reduction in overall modulator power dissipation.

Overall, the $\Sigma\Delta$ toolbox seems to provide accurate performance measures for an ideal 1^{st}-order $\Sigma\Delta$ modulator with varying oversampling ratios and number of quantizer bits. Simulation times for each of these modulators were fairly fast. Initial simulations were used to verify system parameters and to check for quantizer or integrator clipping. The performance measures presented are for 32,000 sample simulations and 32,000 point FFTs. The simulations were performed on an IBM-compatible 486DX2-66 computer

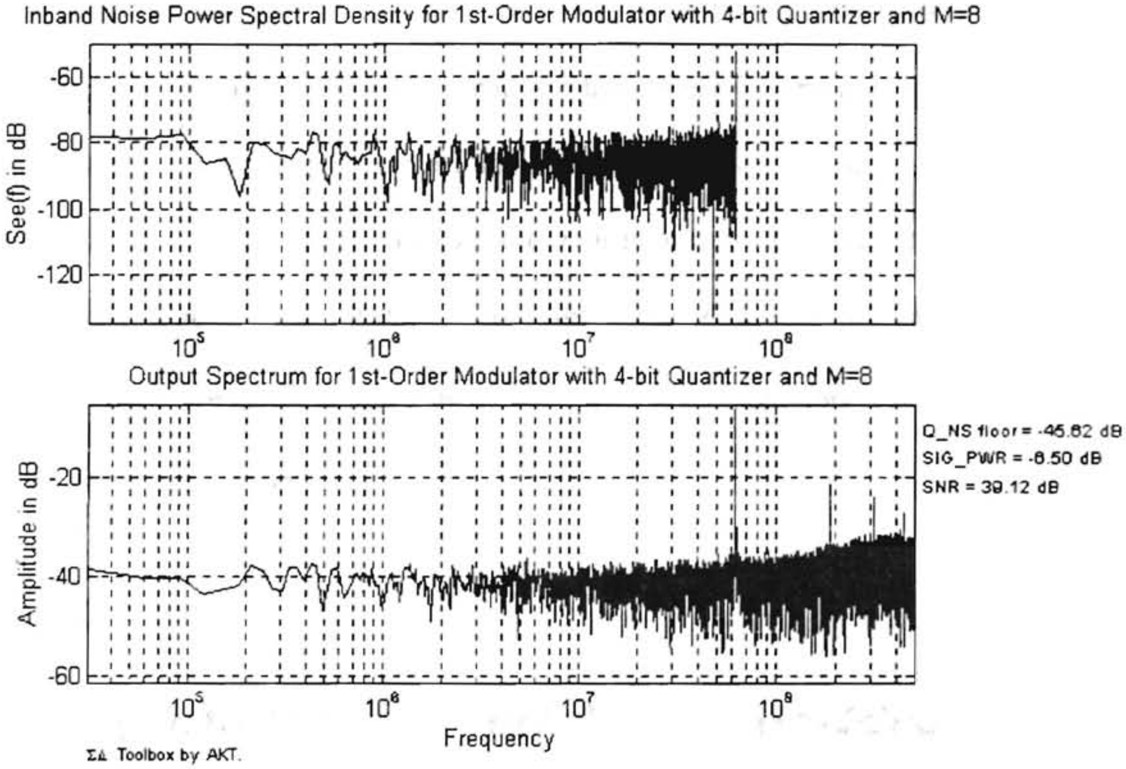


Figure 3.14 Performance Specifications of a 1st-Order ΣΔ Modulator using a 4-Bit Quantizer at M=8 with a Sine Input Voltage.

with 16 Mbytes of memory. On this relatively slow computer, each 32,000 sample simulation was completed in about 2.5 hours. Since each simulation stored each intermediate nodes waveform, the simulation time could be drastically reduced by specifying desired node waveforms to save. Thus, for simple ΣΔ architectures, rapid prototyping is easily achieved on a basic computer system.

Overall, the above simulations prove the validity of the ΣΔ toolbox and its ideal performance measurements. But, to gain further insight into more realistic performance measurements, significant non-idealities of A/D conversion must be addressed by the toolbox. The next section discusses 6 different non-idealities which are included in the ΣΔ toolbox. The inclusion of these non-idealities provides more meaningful performance measurements when considering higher ordered ΣΔ modulators which have inherent instability considerations.

3.2 Significant ΣΔ Modulator Non-Idealities

In the development of the theoretical models discussed in this work thus far, an assumptions were made to the circuit *ideality* of each component. The resulting theoretical performance measurements based on these assumptions are therefore optimal and sometimes unrealistic. The only error considered in the aforementioned development of ΣΔ modulator performance measures was the error introduced by the quantization process of the quantizer. Although this is the most formidable error introduced in ΣΔ A/D conversion, there are a variety of other circuit non-idealities which limit the *ideal* ΣΔ A/D conversion process, particularly in high-fidelity ADCs. In order to extend the usefulness of performance measurements, the ΣΔ toolbox has accounted for 6 significant non-idealities. This section describes the models used in the ΣΔ toolbox for each of the following non-idealities: **integrator harmonic distortion errors, block component settling errors, MOSFET switch charge injection errors, clock *jitter* errors, circuit component mismatch errors, and circuit common-mode errors**. These non-idealities are common in the physical implementation of the most crucial component in ΣΔ modulator, the integrator. Their inclusion will provide mcre realistic simulations which may aid in the rapid prototyping of a desired architecture. The toolbox defines each of these errors individually such that their effects can be viewed independently or cumulatively. After the description of each error model, their effects on the 1st-order example of Figure 3.5 will be seen.

The first non-ideality that is considered is integrator harmonic distortion effects. The most important component in a general ΣΔ modulator is the continuous-time MOSFET integrator. The accuracy of ΣΔ modulation is largely dependent on how precise the modulator input signal can be replicated at the output of the integrator. Therefore any errors introduced by the integrator will have significant consequences in the overall A/D conversion process. Harmonic distortions are primarily due to the inherent non-linearities of the MOS components comprising a MOSFET OTA continuous-time or switched-capacitor integrator. To account for these harmonic distortion effects, the ΣΔ toolbox models the integrator of Figure 3.6 as a distortion-free integrator with a distorted signal at its input after the switched-capacitor stage. If the input to the integrator is $V_{int}(kT_S)$, the distorted input signal $V_{int}(kT_S)'$ is modeled as

$$\begin{aligned} V_{int}(kT_S)' = V_{int}(kT_S) + a_0\left[V_{int}(kT_S)\right]^2 + a_1\left[V_{int}(kT_S)\right]^3 \cdots \\ + a_2\left[V_{int}(kT_S)\right]^4 + a_3\left[V_{int}(kT_S)\right]^5 \end{aligned} \tag{3.14}$$

where a_0, a_1, a_2, and a_3 are the 2nd, 3rd, 4th, and 5th distortion coefficients. This equation is used in the ΣΔ toolbox integrator block function. The distortion coefficients are determined from prescribed harmonic distortion measurements of the MOSFET integrator being considered for a particular ΣΔ architecture. The coefficients are calculated using an elementary trigonometric form of (3.14). That is, if a sine wave of amplitude V_A is assumed as the input to the integrator, the distortion coefficients become

$$a_0 = \frac{2 \cdot HD_2}{V_A} \tag{3.15}$$

$$a_1 = \frac{4 \cdot HD_3}{V_A{}^2} \tag{3.16}$$

$$a_2 = \frac{8 \cdot HD_2}{V_A{}^3} \tag{3.17}$$

$$a_3 = \frac{16 \cdot HD_2}{V_A{}^4} \tag{3.18}$$

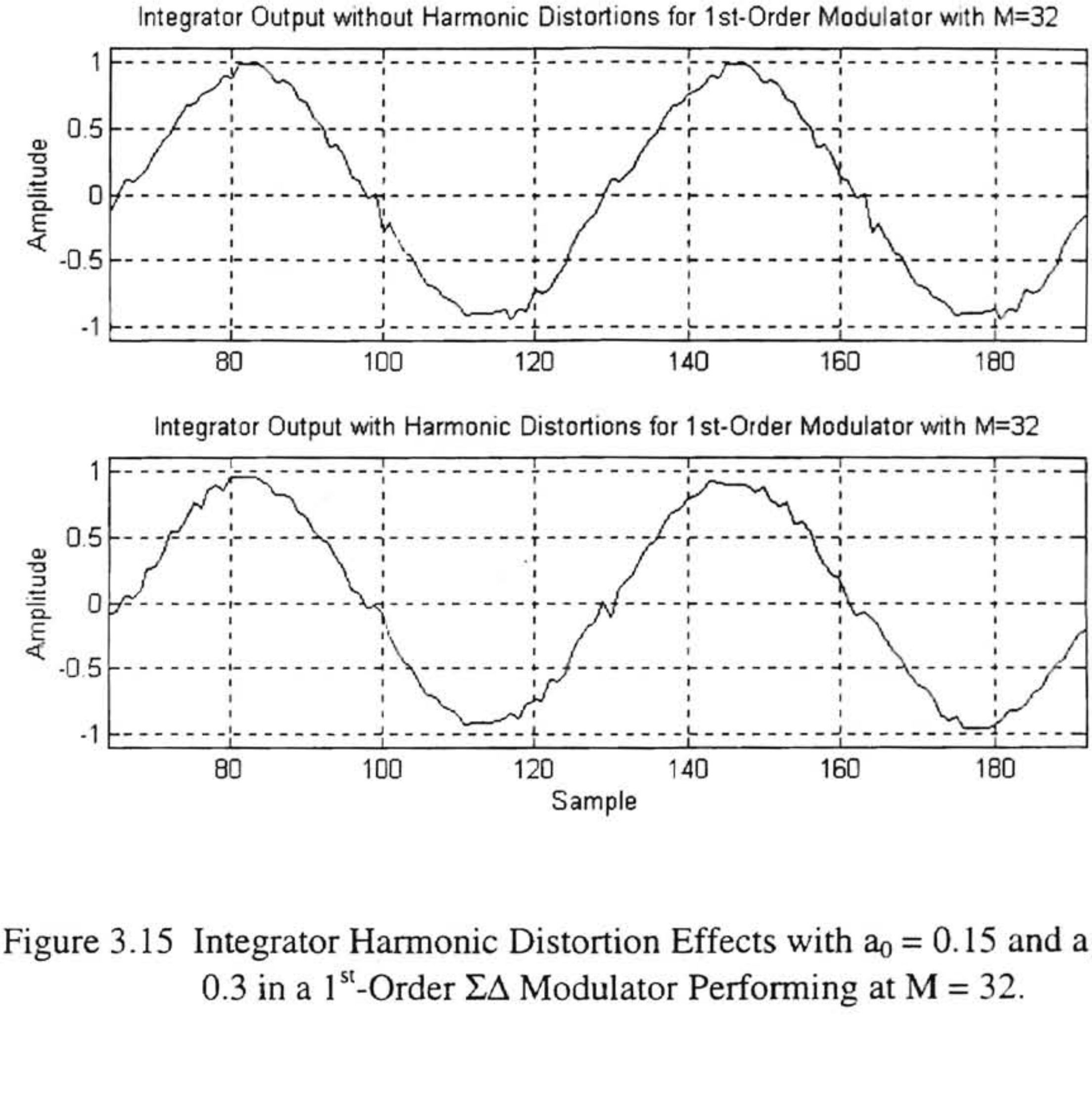


Figure 3.15 Integrator Harmonic Distortion Effects with $a_0 = 0.15$ and $a_1 = 0.3$ in a 1st-Order ΣΔ Modulator Performing at M = 32.

Although it is widely excepted the 2nd and 3rd harmonic distortion terms are the most prominent, the ΣΔ toolbox allows up to the 5th harmonic distortion term to be included in a simulation. Figure 3.15 shows the effect of the 1st and 2nd harmonic distortions in the 1st-order ΣΔ modulator example with a 4-bit quantizer. $a_0 = 0.15$ and $a_1 = 0.3$ while $V_A =$ 1V., f_S =1 GSPS, and M=32 for this simulation. The top graph displays the output of the integrator without harmonic distortion effects. The lower plot gives the effects of harmonic distortion. A close look shows distortion of the higher frequency components in the integrator output. These harmonic distortions are primarily due to spurious harmonic frequencies.

Figure 3.16 gives the Fourier spectrum for a 1st-order ΣΔ modulator example with a 4-bit quantizer. This figure can be compared with Figure 3.17 which displays the Fourier spectrum of the same system with the aforementioned harmonic distortion contributions. Figure 3.18 and 3.19 enlarge pertinent areas of Figures 3.16 and 3.17, respectively. Spurious harmonic frequencies are easily seen in Figure 3.19 when contrasted to Figure 3.18. In addition, the lobes adjacent to the center frequency are reduced by a couple dB when harmonic distortion is considered in Figure 3.19. Although these distortions should not have a very drastic effect in a 1-bit quantizer implementation, they could have an effect on the performance of ΣΔ modulators using multi-bit quantizers. This idea should be noted when considering higher order ΣΔ modulators, such as the one discussed in Section 3.4, which depend on higher frequency components for a more precise representation of the modulator input. If the harmonic distortion terms

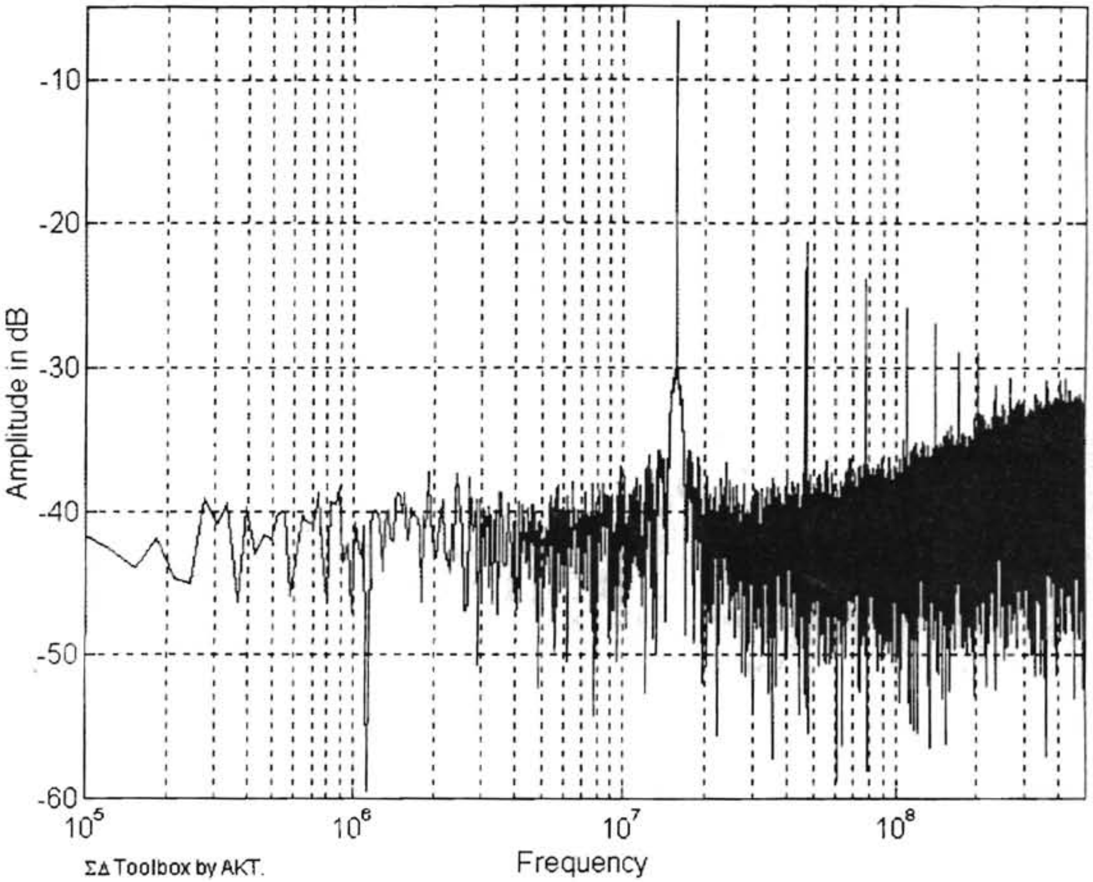


Figure 3.16 Performance Specifications of a 1st-Order ΣΔ Modulator using a 4-Bit Quantizer at M=32 **without** Harmonic Distortion.

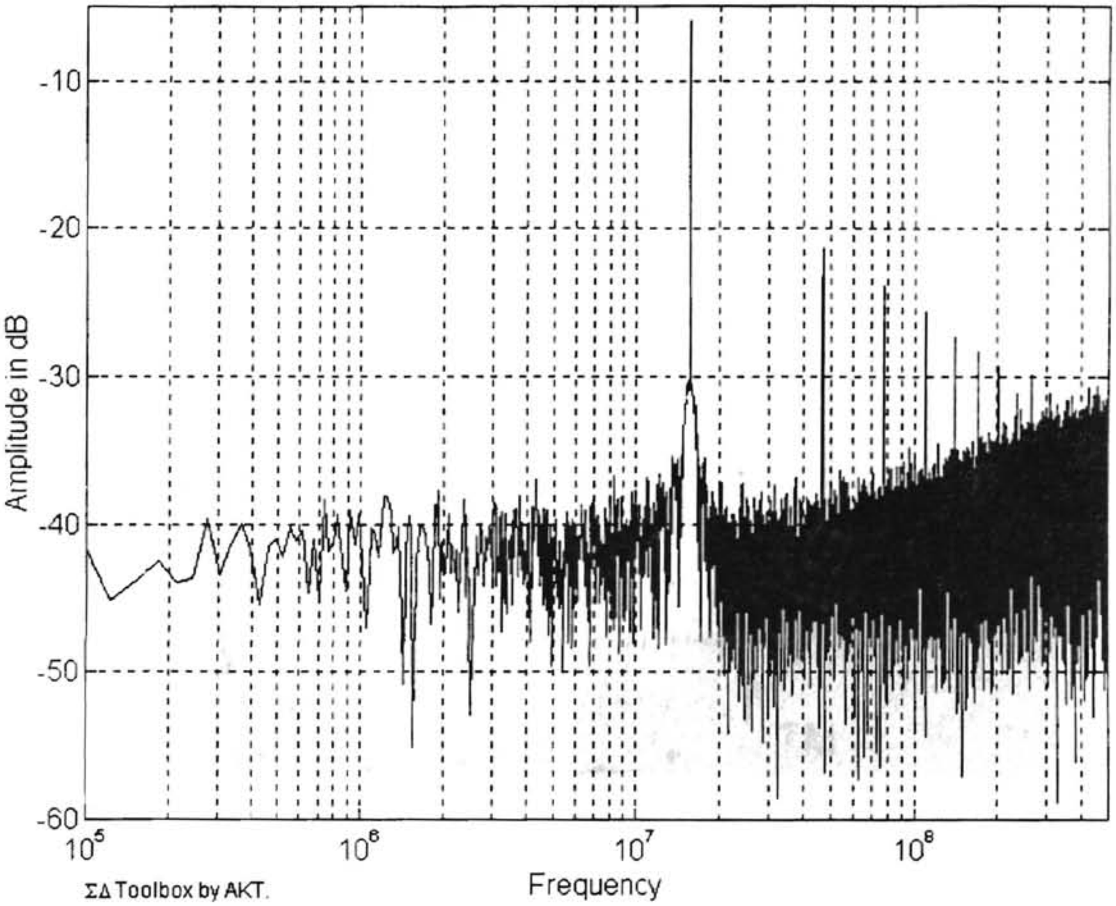


Figure 3.17 Performance Specifications of a 1st-Order ΣΔ Modulator using a 4-Bit Quantizer at M=32 **with** Harmonic Distortion.

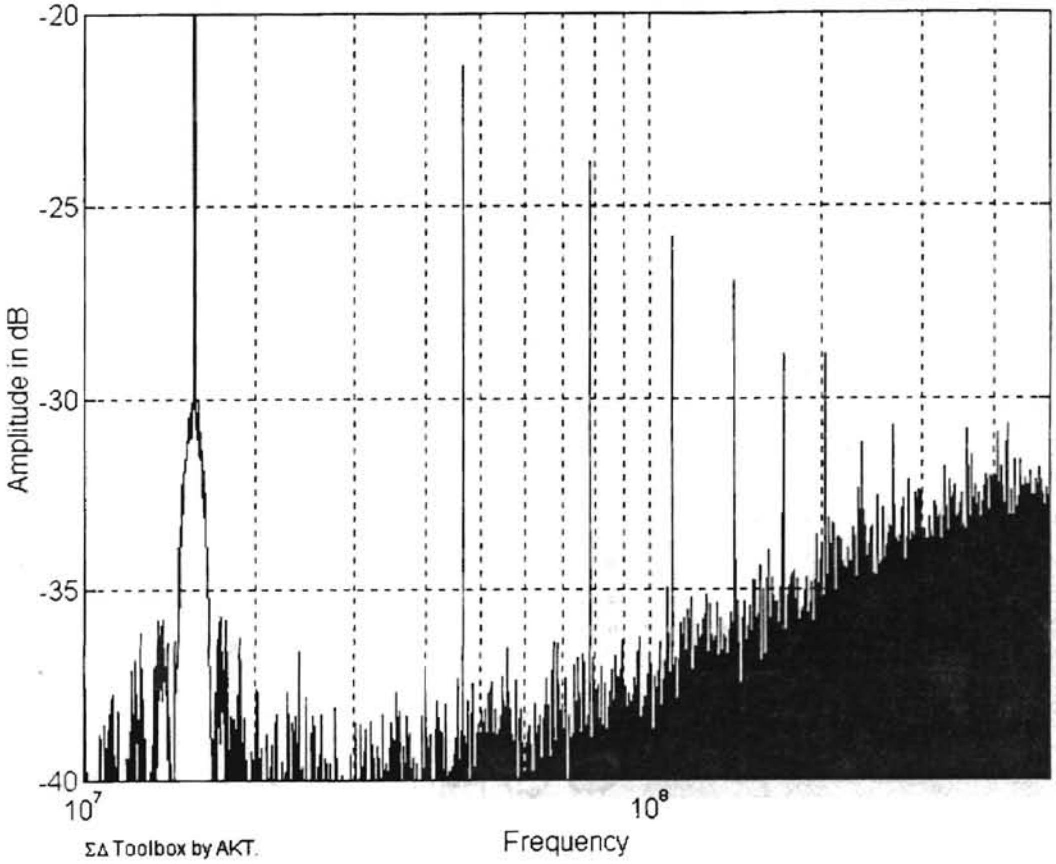


Figure 3.18 Enlarged View of Performance Specifications Figure 3.15 **without** Harmonic Distortion.

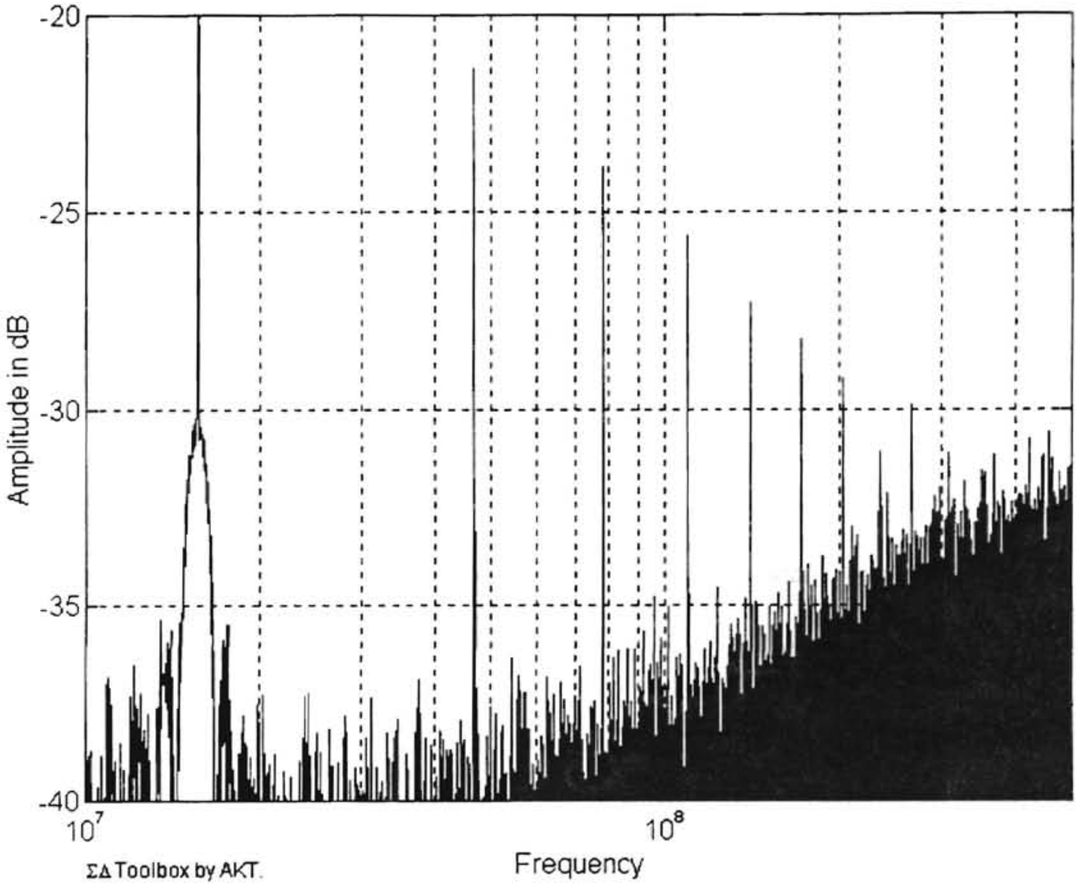


Figure 3.19 Enlarged View of Performance Specifications Figure 3.16 **with** Harmonic Distortion.

are large enough, they may cause improper comparator decisions in the quantizer which may have an adverse effect in overall ΣΔ modulation performance. Another non-ideality which may cause the same problem is incomplete settling of ΣΔ modulator component blocks.

Each component block in a ΣΔ modulator has an associated settling time. If each component does not completely settle within its allotted clock phase time, incorrect block outputs are fed through the ΣΔ modulation loop. This may also be detrimental to the A/D conversion process. Thus, component settling time issues are included in the ΣΔ toolbox by calculating the settling time constant for each individual ΣΔ modulator block. The toolbox can incorporate the settling time error in the output of each block using the calculated time constants. The most difficult time constant calculation is for the integrator. Appendix A details the mathematical development of the integrator's effective settling time error by employing a two-pole, small-signal circuit MOSFET model in the Laplace domain. Figure 3.20 describes the schematic diagram for this integrator model.

The 2 time constants are obtained from the mathematical representation of the integrator's voltage gain, A_V. Applying Kirchhoff's Current Law at the gate and drain nodes gives the following two equations:

$$(v' - v_{in}) \cdot Y_{g1} + v' \cdot C' \cdot s + (v' - v_o) \cdot (s \cdot C_x + Y_{f1}) = 0 \qquad \text{(A.1)}$$

$$v_o \cdot (g_o + s \cdot C_{L1}) + g_{m1} \cdot v' + (v_o - v') \cdot (s \cdot C_x + Y_{f1}) = 0 \qquad \text{(A.2)}$$

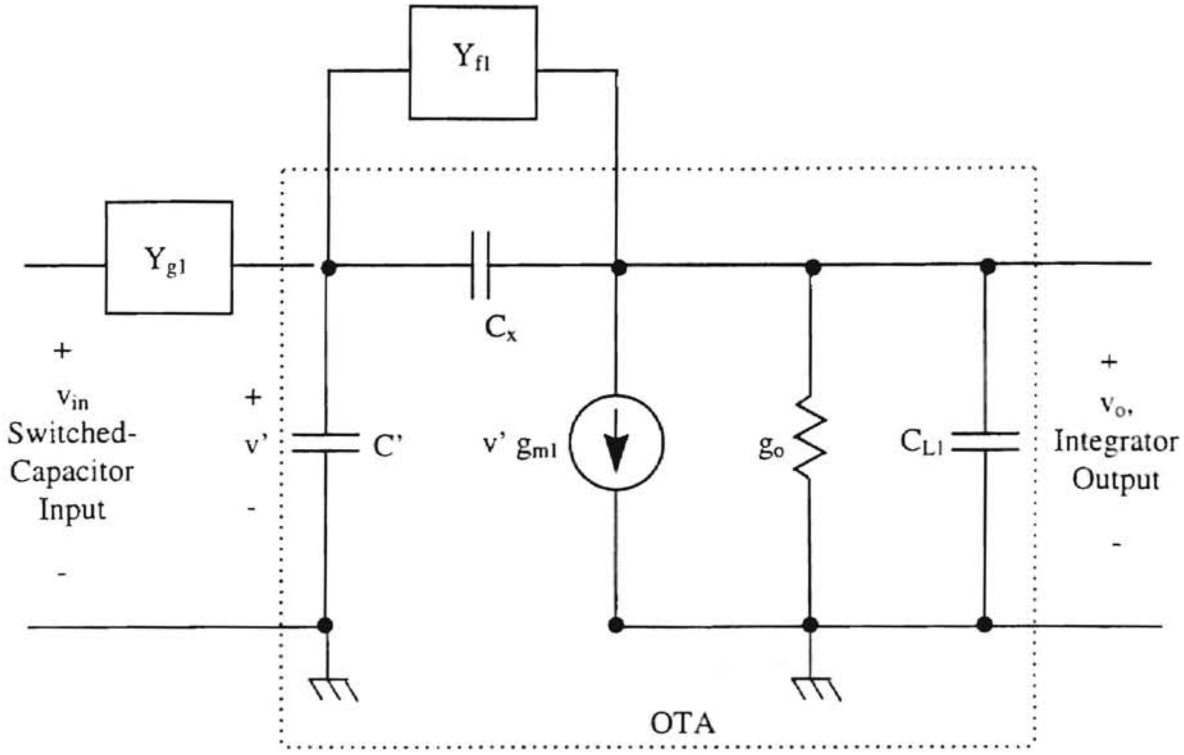


Figure 3.20 Schematic of Two-Pole, Small-Signal Integrator Model.

where, v_{in} is the switched-capacitor input voltage; v' is the gate to source voltage; v_o is the output voltage; C' is the OTA input parasitic capacitance; C_x is the gate to output parasitic capacitance; C_{L1} is the load capacitance; g_{m1} is the OTA's effective transconductance; Y_{g1} is the switch's admittance; Y_{f1} is the integration capacitor's admittance; and s is the Laplacian operator. Equations (A.1) and (A.2) are used to find A_V. Solving these equations, substituting for the full form of the Y_{g1} and Y_{f1}:

$$Y_{gl} = \frac{1}{\left(R_{gl} + \frac{1}{s \cdot C_{gl}}\right)} = \frac{C_{gl}}{\left(R_{gl} \cdot s \cdot C_{gl} + 1\right)} \tag{A.6}$$

$$Y_{gl} = s \cdot C_{int} \tag{A.7}$$

where, R_{gl} is the effective switch resistance; C_{gl} is the effective sampling switch capacitance; and C_{int} is the integration capacitor. Simplifying by removing insignificant terms, A_V is found to be

$$A_v = \frac{\dfrac{C_{gl} \cdot (-g_{ml} + C_{int} \cdot s)}{(C_{int} \cdot g_{ml})}}{\dfrac{\left[\begin{array}{l}(C_{L1} \cdot C_{int} \cdot R_{gl} \cdot C_{gl} + C_{int} \cdot C' \cdot R_{gl} \cdot C_{gl} + C_{L1} \cdot C' \cdot R_{gl} \cdot C_{gl}) \cdot s^2 \ldots \\ \left(\begin{array}{l} C_{L1} \cdot C' + C_{L1} \cdot C_{gl} + C_{int} \cdot g_{ml} \cdot R_{gl} \cdot C_{gl} \ldots \\ + C_{int} \cdot C' + C_{L1} \cdot C_{int} + C_{int} \cdot C_{gl} \end{array}\right) \cdot s + C_{int} \cdot g_{ml}\end{array}\right]}{C_{int} \cdot g_{ml}}} \tag{A.8}$$

The denominator for this equation is used to determine the 2 time constants for the integrator. The ΣΔ toolbox substitutes the prescribed values for each of the variables in the denominator of (A.8). It then finds the roots of this numerical form of the denominator. The roots are of the structure:

$$(s \quad p_1) \cdot (s - p_2) \tag{A.10}$$

where p_1 and p_2 are the calculated poles of the denominator. The final result for the settling time constants τ_1 and τ_2 becomes:

$$\tau_1 = \frac{1}{p_1} \tag{A.11}$$

$$\tau_2 = \frac{1}{p_2} \tag{A.12}$$

These time constants are used to determine if incomplete settling occurs for the integrator during the integration phase of the clock cycle. The ΣΔ toolbox accounts for a possible integrator settling error by including the time constant contributions at the output of the integrator for a specified phase time. The toolbox uses (3.19)

$$w = V_{int\_out} \cdot \left(1 - \exp\left(-\frac{ph\_time}{\tau_1}\right) - \exp\left(-\frac{ph\_time}{\tau_2}\right)\right) \tag{3.19}$$

where w is the output of the integrator including settling error, V_{int_out} is the output of the integrator without settling error, and ph_time is the integration phase time in seconds. The settling error of the quantizer and DAC blocks is included in a less tedious manner.

The toolbox requires that the values for the *equivalent capacitances and resistances* must be included in the desired ΣΔ modulator architecture file. The respective time constant for each block is calculated as follows:

$$\tau_{block} = R_{eq,block} \cdot C_{eq,block} \tag{3.20}$$

where τ_{block} is the block's time constant, $R_{eq,block}$ is the block's equivalent resistance, and $C_{eq,block}$ is the block's equivalent capacitance. Using (3.20), the output for each block becomes:

$$V_{out,block} = V_{out} \cdot \left(1 - \exp\left(-\frac{ph\_time}{\tau_{block}}\right)\right) \tag{3.21}$$

where $V_{out,block}$ is the output of the block including settling error, V_{out} is the output of the block without settling error, and ph_time is the integration phase time in seconds. A simulation of the 1^{st}-order ΣΔ modulator can show the effects of incomplete settling of its block components. Figure 3.21 displays the integrator outputs for a simulation performing at f_S = 1 GSPS and M=32. The upper plot gives the ideal integrator output, while the lower plot is for the integrator output with incomplete block component settling. In the simulation, the ΣΔ toolbox calculated the following time constants from the specified data for each block component: $8.6\text{x}10^{-11}$ sec^{-1} and $3.8\text{x}10^{-12}$ sec^{-1} for the integrator, $1\text{x}10^{-11}$ sec^{-1} for the quantizer, and $1\text{x}10^{-11}$ sec^{-1} for the DAC. A close look at the graph shows how settling errors cumulatively distort the integrator output.

Figure 3.22 gives the enlarged output frequency spectrum for the same system. This figure can be compared to the undistorted output frequency spectrum of Figure 3.18. The 3^{rd} through 7^{th} harmonics are similar in amplitude on both Figures 3.16 and 3.22. But, higher harmonic frequency amplitudes of Figure 3.22 are inconsistent with their corresponding undistorted harmonic frequencies of Figure 3.16. Once again, higher frequency contributions have been adversely effected by the non-ideality. In turn, settling errors may be detrimental to the important local averaging process of ΣΔ modulation. As seen in the last section, local averaging aids in a more accurate digital representation of the analog input to the modulator. Also, settling errors may have an unfavorable effect on A/D conversion accuracy when considering higher order ΣΔ modulators, such as the one discussed in Section 3.4, which depend on higher frequency components at the integrator output for increased A/D conversion resolution. Along with the two aforementioned non-

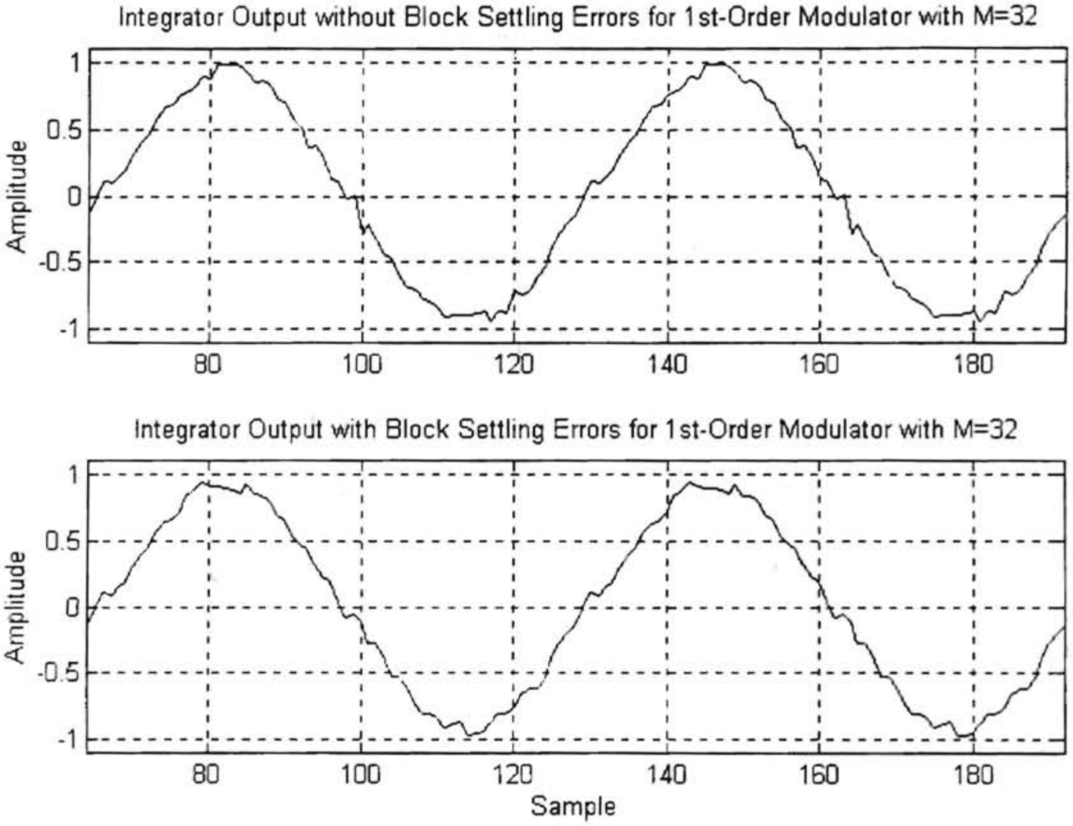


Figure 3.21 Block Settling Error Effects in a 1st-Order ΣΔ Modulator Performing at M = 32.

idealities, another error that has a direct effect on the integrator output is switch charge injection or charge feed-through.

Charge injection is an inherent problem with switched-capacitor sampling circuits [Wegmann, 1987]. Referring to Figure 3.6 showing the switched-capacitor continuous-time integrator, the switches are typically implemented by MOSFETs. Figure 3.22 displays a circuit model of the MOSFET sampling switch just preceding the integrator. The diagram models the middle portion of the switched-capacitor integration circuit around the ϕ_2 dependent switch just before the intermediate node of Figure 3.6. The system of Figure 3.22 consists of the sampled difference signal to be integrated, Vin, the sampling capacitor, C_{si}, the MOSFET switch transistor, and the integration capacitor, C_{int}. The transistor's gate voltage, V_G, controls the on/off state of the transistor during the integration phase ϕ_2. Charge injection limits the accuracy of the integration process by introducing an error charge, ΔQ_G, onto C_{int} each time the transistor is turned off.

The error charge is due to carriers released from the switch's conduction channel and due to coupling through the gate-to-diffusion parasitic overlap capacitance, C_{gd}. This has an adverse consequence on the output voltage of the integrator. A simplified circuit model of this MOSFET switch for charge injection analysis is given in Figure 3.23. This model assumes that there is a relatively long fall time associated with the switch and the capacitances, C_{si} and C_{int}, are much larger than the gate oxide capacitance, C_{ox}. The transistor's C_{ox} is considered as a distributed oxide capacitance associated with the switch's time-varying channel conductance representation, $g[V_g(t)]$. If these assumptions

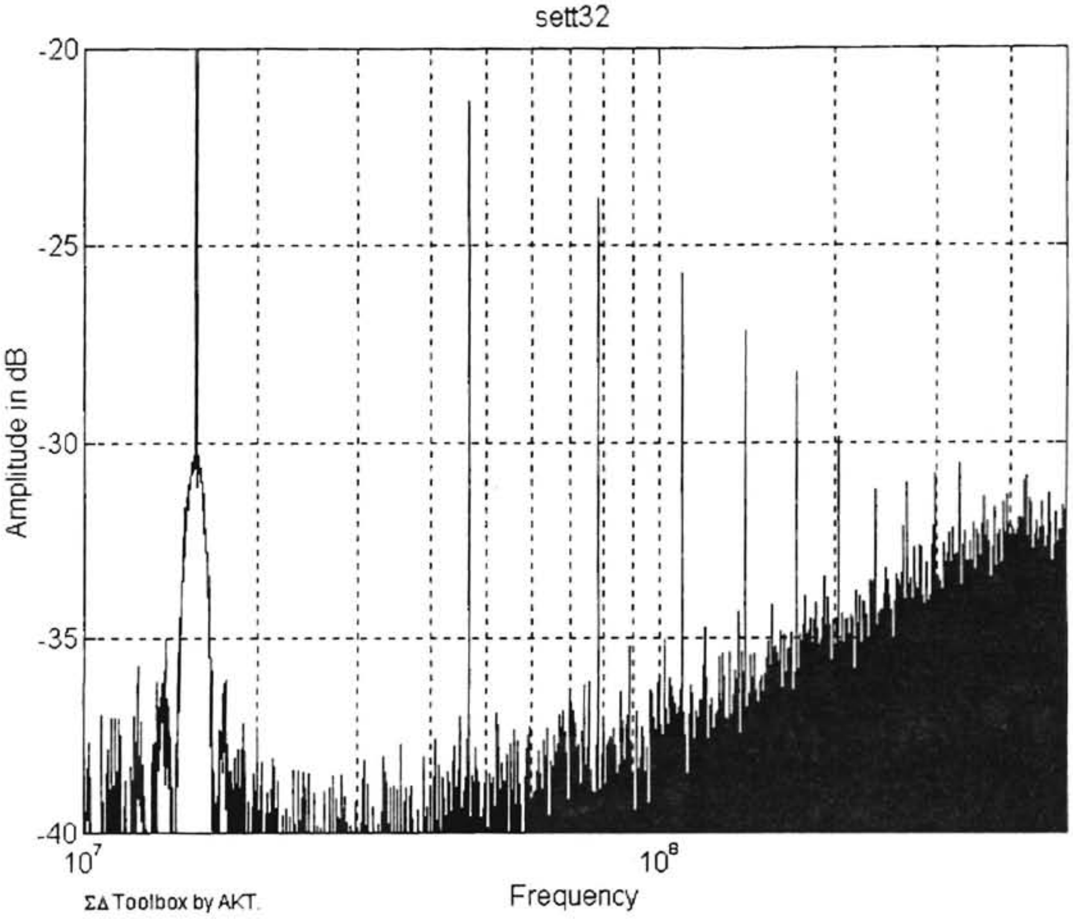


Figure 3.22 Frequency Spectrum of Block Settling Error Effects on the Output of a 1st-Order ΣΔ Modulator Performing at M = 32.

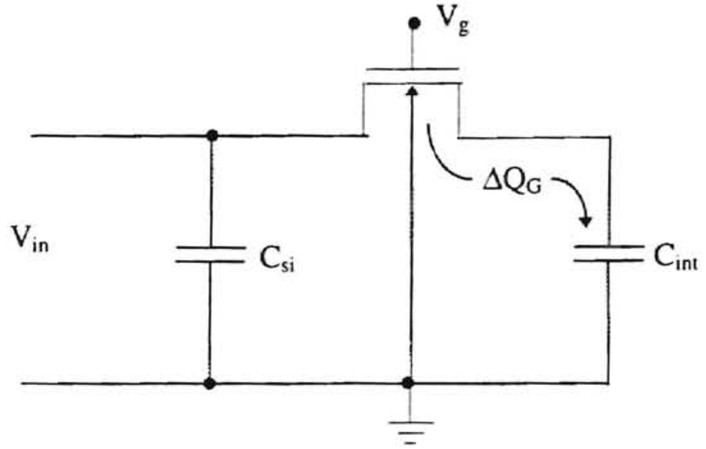


Figure 3.22 Simple Circuit Model for Charge Injection Analysis.

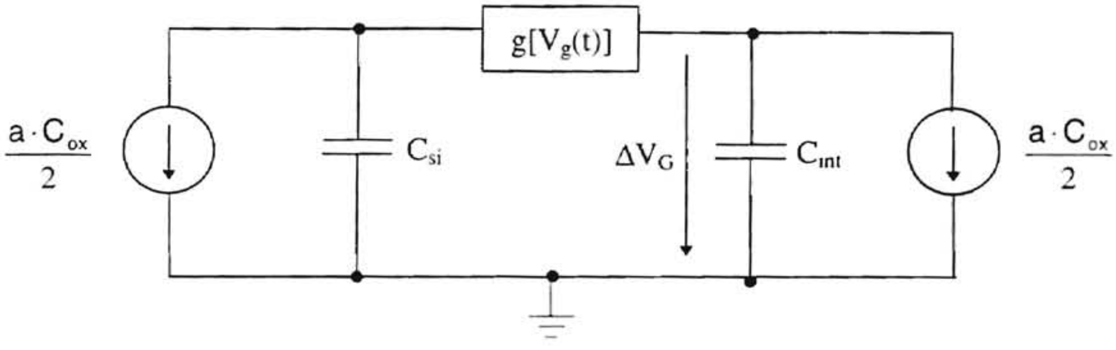


Figure 3.23 Simple Circuit Model for Charge Injection Analysis.

are satisfied, a linear decrease of V_g with slope ‘a’ across C_{ox} is equivalent to a constant current source of (aC_{ox})flowing symmetrically to both ends of the transistor. $g[V_g(t)]$ can be modeled as

$$g\left[V_g(t)\right] = \beta \cdot \left(V_{g,ON} - a \cdot t - V_{TE}\right) \tag{3.22}$$

where β is the transistor current-gain, $V_{g,ON}$ is the gate voltage required to turn on the transistor, ‘a’ is the slope of the slope of the gate’s on /off transition voltage, t is time and V_{TE} is the effective threshold voltage determining the on/off state of the transistor. The transistor current gain, β, is modeled as

$$\beta = \frac{W}{L} \cdot \mu \cdot C_{ox} \tag{3.23}$$

where W is the width of the transistor, L is the length of the transistor, and μ is the transistor’s carrier mobility; while the slope of the gate’s on /off transition voltage, ‘a’, is determined by

$$a \approx \frac{0.8 \cdot V_{g,ON}}{t_{fall}} \tag{3.24}$$

where t_{fall} is the fall time associated with the on/off transition voltage of the transistor. With these models defined, Figure 3.23 is resolved into the following normalized differential equation:

$$\frac{dV}{dT} = (T - B) \cdot \left[\left(1 + \frac{C_{int}}{C_{si}}\right) \cdot V + 2 \cdot T \cdot \frac{C_{int}}{C_{si}}\right] - 1 \tag{3.25}$$

where the normalized terms are

Normalized Voltage Error: $$V = \frac{\Delta V_g}{\frac{C_{ox}}{2} \cdot \sqrt{\frac{a}{\beta \cdot C_{int}}}}, \tag{3.25a}$$

Normalized Time: $$T = \frac{t}{\sqrt{\frac{C_{int}}{a \cdot \beta}}}, \text{ and} \tag{3.25b}$$

Normalized Switching Parameter: $$B = \left(V_{g,ON} - V_{TE}\right) \cdot \sqrt{\frac{\beta}{a \cdot C_{int}}} \tag{3.25c}$$

The ΣΔ toolbox uses these equations and an initial condition of $V = 0$ to calculate the charge injection error added to each sample being integrated. The toolbox calculates V by inserting the required data values and integrating (3.25) from $0 < T < B$. This V is finally substituted into (3.25a) to find the charge injection error voltage, ΔV_g. For the 1st-order ΣΔ modulator example, a charge injection error of -0.000638 V. was determined using the following parameters: C_{si} = 30 pF, C_{int} = 32 pF, W = 10000 μm, L = 22 μm, $V_{g,ON}$ = 1 V., V_{TE} = 0.5 V., μ = 500 cm^2/V., C_{ox} = 6.8X10^{-8} F/cm^2, f_S = 1 GSPS, and M = 8. Although many other switches exist in the clocked switched-capacitor network and the clocked DAC block, the ΣΔ toolbox only determines charge injection error for the switch just preceding the integrator's intermediate node. Charge injection in the other switches are usually fed to ground when turned off.

The fourth non-ideality considered, clock *jitter* error, is also associated with the MOSFET switching circuits. Clock *jitter* error arises from the sampling time uncertainty in the clocked blocks of the ΣΔ modulator. This uncertainty is due to random thermal noise introduced into the clocking network. The ΣΔ toolbox uses a very simple model for this error. Normally, the toolbox uses ideal uniform time-sampled data from the created modulator input waveform to be operated upon during each ΣΔ loop iteration. For example, a sine wave with a maximum voltage V_A is ideally created by the toolbox as

$$\mathrm{V(sample)} = \mathrm{V_A} \cdot \sin\left(2 \cdot \pi \cdot \frac{\mathrm{f_{des}}}{\mathrm{f_S}} \cdot \mathrm{sample}\right) \tag{3.26a}$$

where sample is an integer number from 1 to the number of samples desired. The ΣΔ toolbox introduces a unit amplitude noise term scaled by a user defined clock jitter percentage to account for the non-uniform time-sampling. That is, (3.26a) becomes

$$\mathrm{V(sample)} = \mathrm{V_A} \cdot \sin\left(2 \cdot \pi \cdot \frac{\mathrm{f_{des}}}{\mathrm{f_S}} \cdot \mathrm{sample} \cdot \left(1 + \mathrm{jitter\_err} \cdot \mathrm{rand}(1)\right)\right) \tag{3.26a}$$

where jitter_err is the user defined maximum percentage time deviation around an ideal time-samples sample, and rand(1) is the Matlab term that generates a random noise term with a maximum amplitude of ±1. Figure 3.24 shows how a jitter_err of 2% effects the input waveform created by the toolbox used as an input to a ΣΔ modulator. Once again, this input waveform is applicable to the 1st-order ΣΔ modulator example performing with f_S = 1 GSPS and M=32. The upper plot gives the sine waveform for an ideal time-sampled input. The lower plot graphs the effects of improper sampling in the time-domain. The ΣΔ toolbox distorts the input waveform to account for clock jitter effects effectively representing the non-uniformity or uncertainty at uniformly sampled data points.

The fifth non-ideality the ΣΔ toolbox includes is block component mismatch errors. These errors originate from process inaccuracies inherent to physical IC layout of the variety of capacitors and resistors used in a ΣΔ modulator. The mismatch is most prominent during the computation of the closed-loop gain values for the integrator and any amplifiers in the circuit. For instance, if a process is known to have a capacitor mismatch error of 2%, the desired capacitor gain stage ratios may differ from the actual implemented gain ratio by as much as 4%. Since component mismatch errors are considered independent random variables, they sum in a root-mean-squared fashion, which further increases differences between actual physical gains and computed gains. Although the performance of many ADC's are greatly effected by component mismatch errors, ΣΔ modulation is fairly resistant to them.

The ΣΔ toolbox accounts for component mismatch errors in a similar approach to the inclusion of clock jitter errors. The toolbox adds a scaled version of a unit noise term to each of the capacitances and resistances given in the desired architecture file. That is, the impedance of a particular component is modeled as

$$Z'_i = \left(1 + \text{mis\_err} \cdot \text{rand}(1)\right) \cdot Z_i \tag{3.27}$$

where Z'_i is the new component impedance including component mismatch error effects, mis_err is the maximum process mismatch error percentage, rand(1) is the Matlab term which calculates a random value between ± 1, and Z_i is the desired component

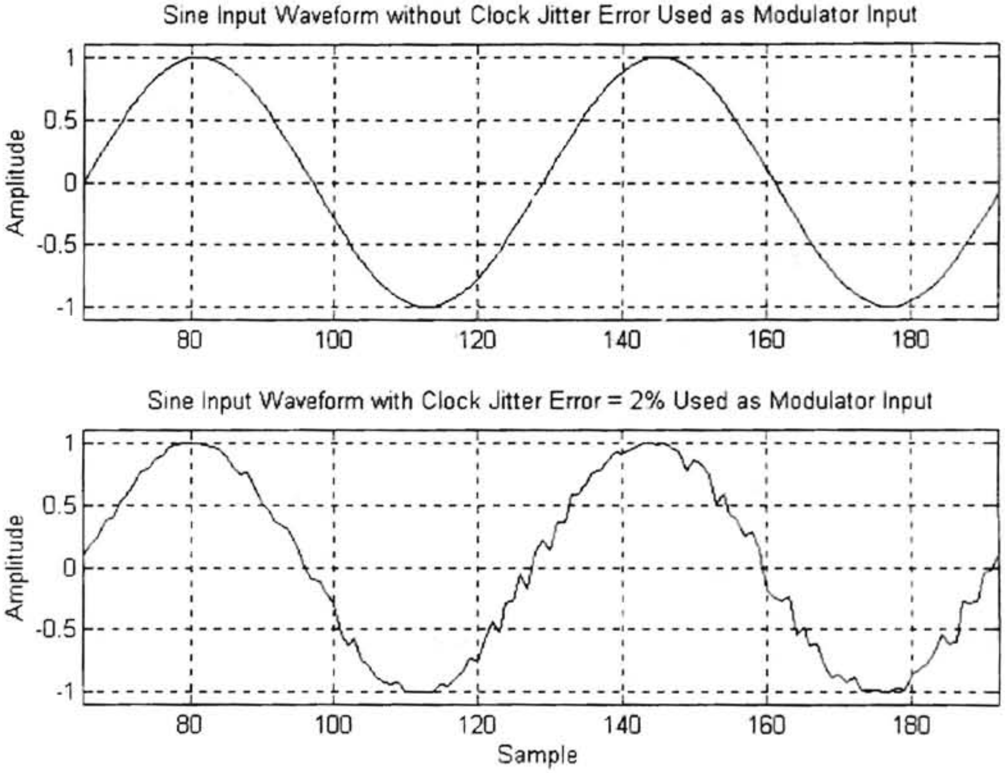


Figure 3.24 Clock Jitter Error Effects of 2% on Modulator Input Waveform for a 1st-Order ΣΔ Modulator Performing at M = 32.

impedance. The new resistances and capacitances are used in the iteration of the ΣΔ loop resulting in the inclusion of these mismatch errors.

Figure 3.25 displays the effects of component mismatch errors on the integrator output of the 1st-order ΣΔ modulator example performing at f_S = 1 GSPS and M = 32. The upper plot shows the effects for a maximum mis_err = 2%, while the lower plot exhibits the effects for a maximum mis_err = 20%. Figure 3.26 shows the effects of component mismatch error over and above quantization error. The upper plot gives the effects for a maximum mis_err = 2%, while the lower plot demonstrates the effects for a maximum mis_err = 20%. There is an obvious drop in performance in the case of 20% error when compared to the plot of 2% error. In addition to adding an additional amount of error over and above the quantization noise, the plot of 20% mismatch error begins to show a correlation to the input waveform. This observation is of concern since it violates Bennett's noise model discussed in Section 2.1. Figures 3.27 and 3.28 give the frequency spectrums for the output of the 1st-order ΣΔ example including component mismatch errors of 2% and 20%, respectively. Once again, the 3rd through 7th harmonic frequencies are not highly affected. But, harmonic frequencies above the 7th order are affected. The graphs show the general resilience of ΣΔ modulation to component mismatch errors.

Although these errors should have less effect on a 1-bit quantizer implementation, they can have a significant effect on the performance of ΣΔ modulators using multi-bit quantizers by limiting the quantizer's effective DR. In addition, some of the aforementioned errors may be removed by the differential aspect of the integrator.

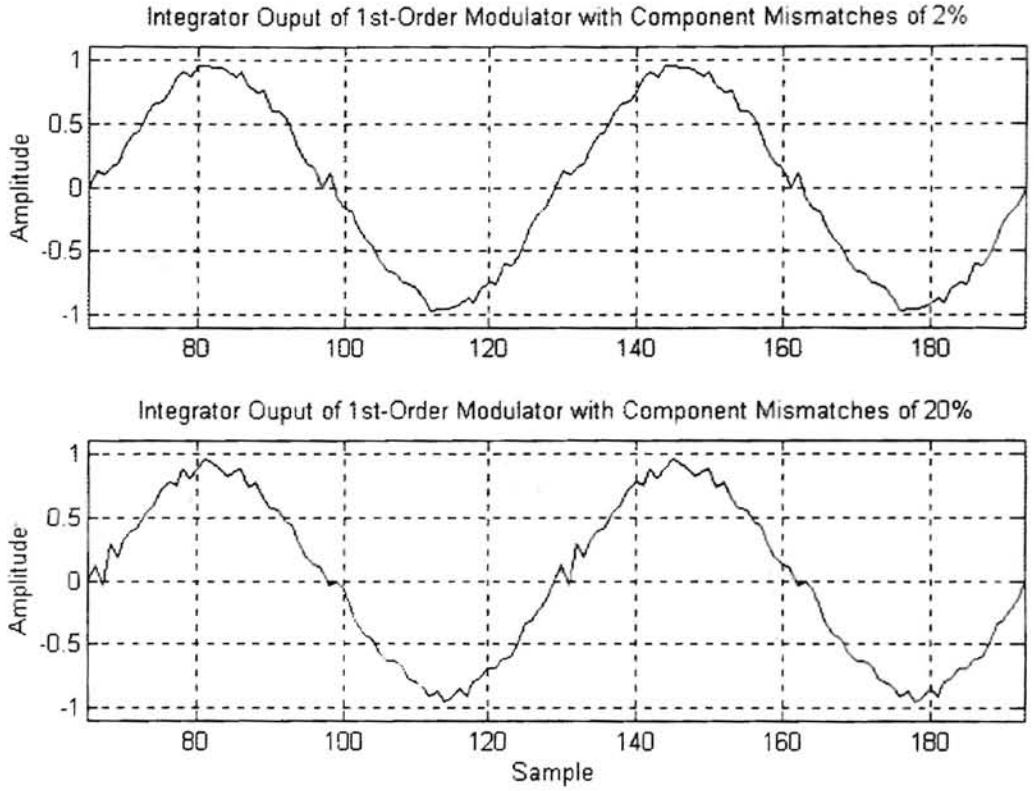


Figure 3.25 Effects of 2% and 20% Component Mismatch Errors in a 1st-Order ΣΔ Modulator Performing at M = 32.

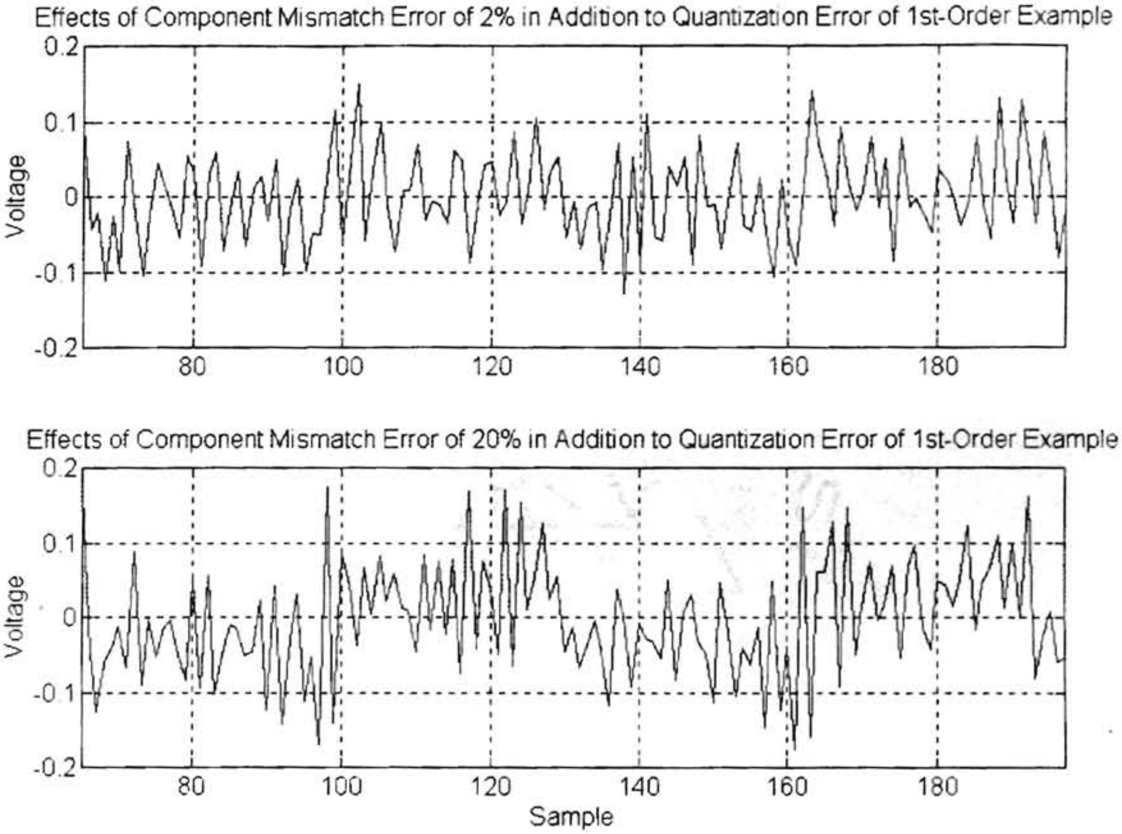


Figure 3.26 Effects of 2% and 20% Component Mismatch Errors in a 1st-Order ΣΔ Modulator Performing at M = 32.

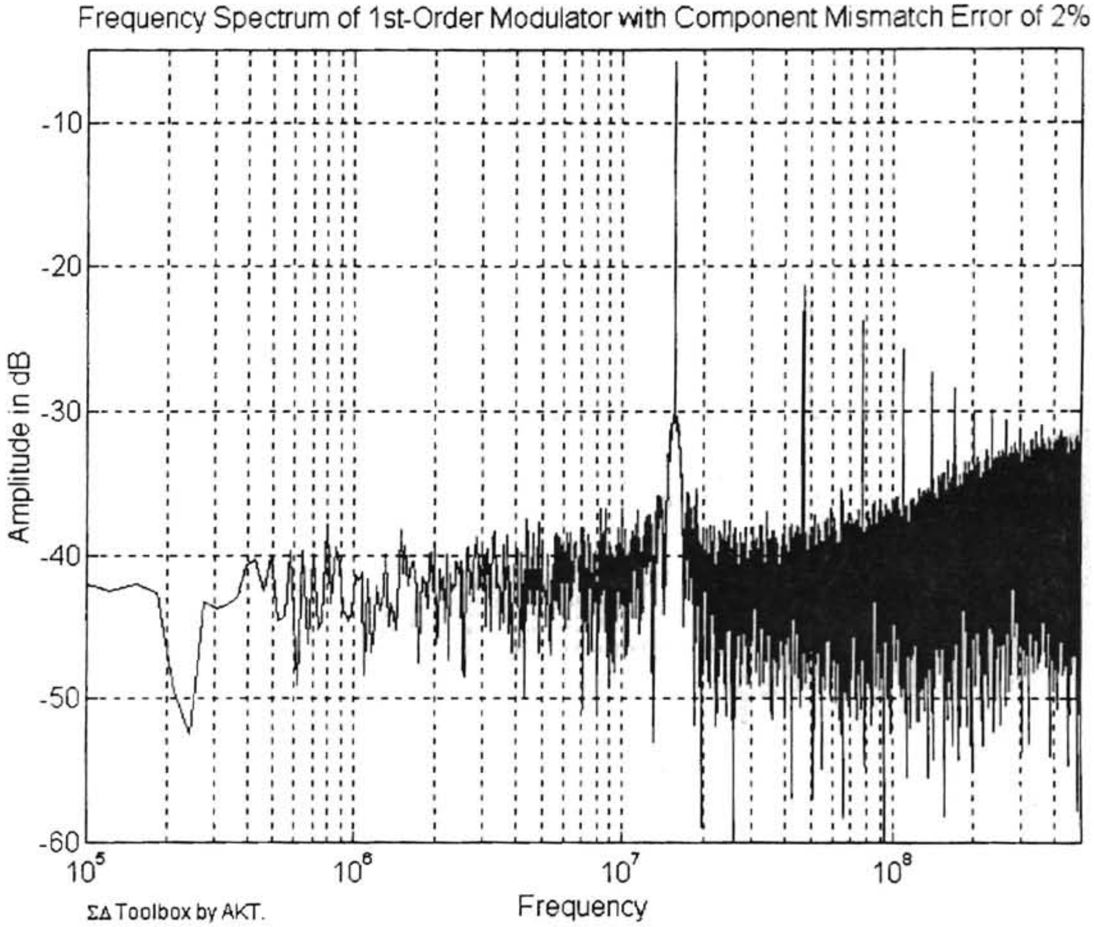


Figure 3.27 Frequency Spectrum of 2% Component Mismatch Error Effects on the 1st-Order ΣΔ Modulator Output Performing at M = 32.

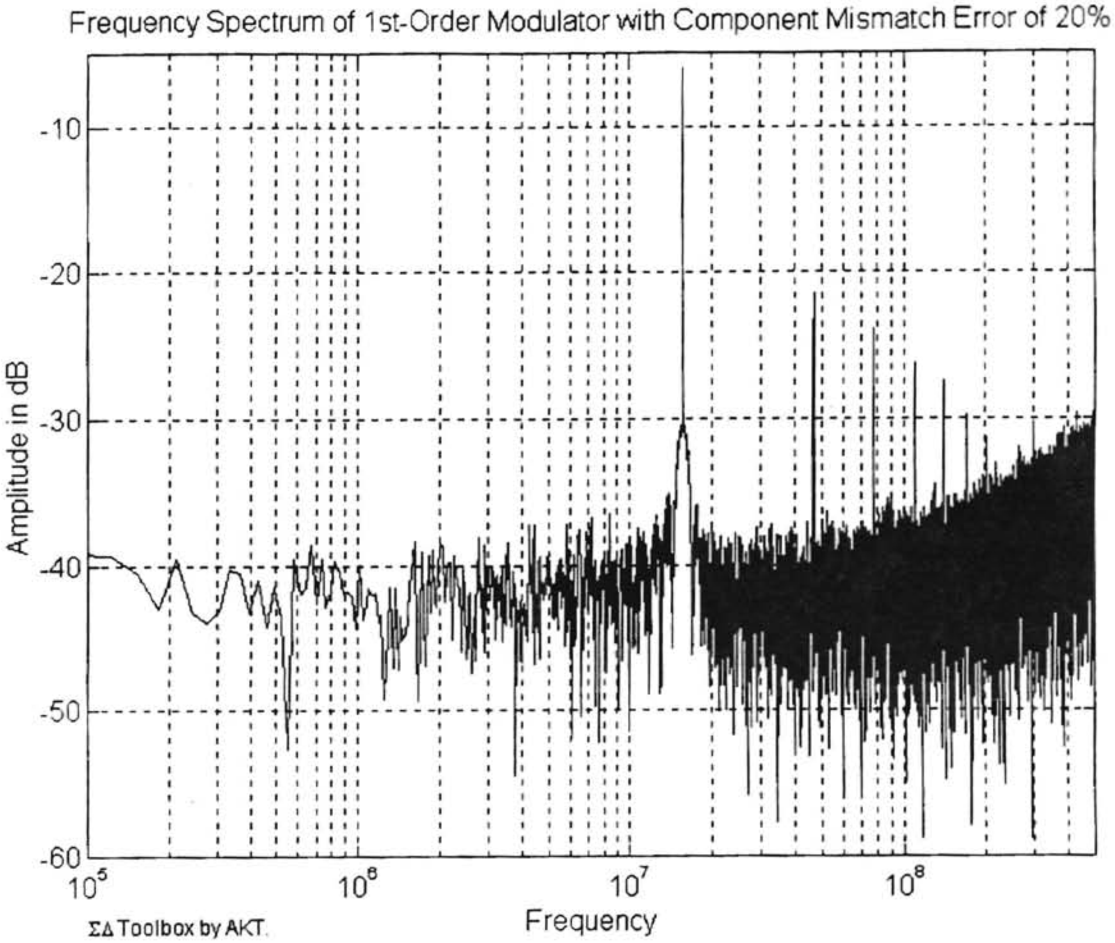


Figure 3.28 Frequency Spectrum of 20% Component Mismatch Error Effects on the 1st-Order ΣΔ Modulator Output Performing at M = 32.

As mentioned in section 3.1.3, the ΣΔ toolbox implements a differential integrator circuit to aid in the rejection of common-mode errors. Figure 3.29 displays a block diagram of a 1st-order ΣΔ modulator implemented in a differential form. The input to the differential modulator is fed to the +Input node, while its inverse is fed to the -Input node. The toolbox iterates both modulation loops specified in the architecture file. The digital. quantizer output of the lower half of the diagram is subtracted from the quantizer output of the upper half to give the system's final output. The ΣΔ toolbox uses this fact to validate common-mode error removal or non-zero common-mode gain. Common-mode errors arise when similarly signed errors are introduced to the upper and lower halves of the differential circuit. Common-mode errors are thus reduced or eliminated by the final subtraction operation for the resulting modulator output.

The non-idealities described in this section combine to give the ΣΔ toolbox the ability to simulate ΣΔ modulators more realistically. As seen by the plots in this section, these non-idealities do hamper the performance of ΣΔ modulators, especially for the 4-bit quantizer implementations. After a desired design is theoretically designed, the inclusion of the errors in simulations may validate the designer's expectations of the design's performance. If the errors tend to unexpectedly reduce the anticipated A/D conversion resolution, the designer may redesign the modulator to take into account any performance degradation before the circuit is simulated with some other transistor-level simulator. This approach and use of the ΣΔ toolbox will aid in the rapid prototyping of desired ΣΔ modulators.

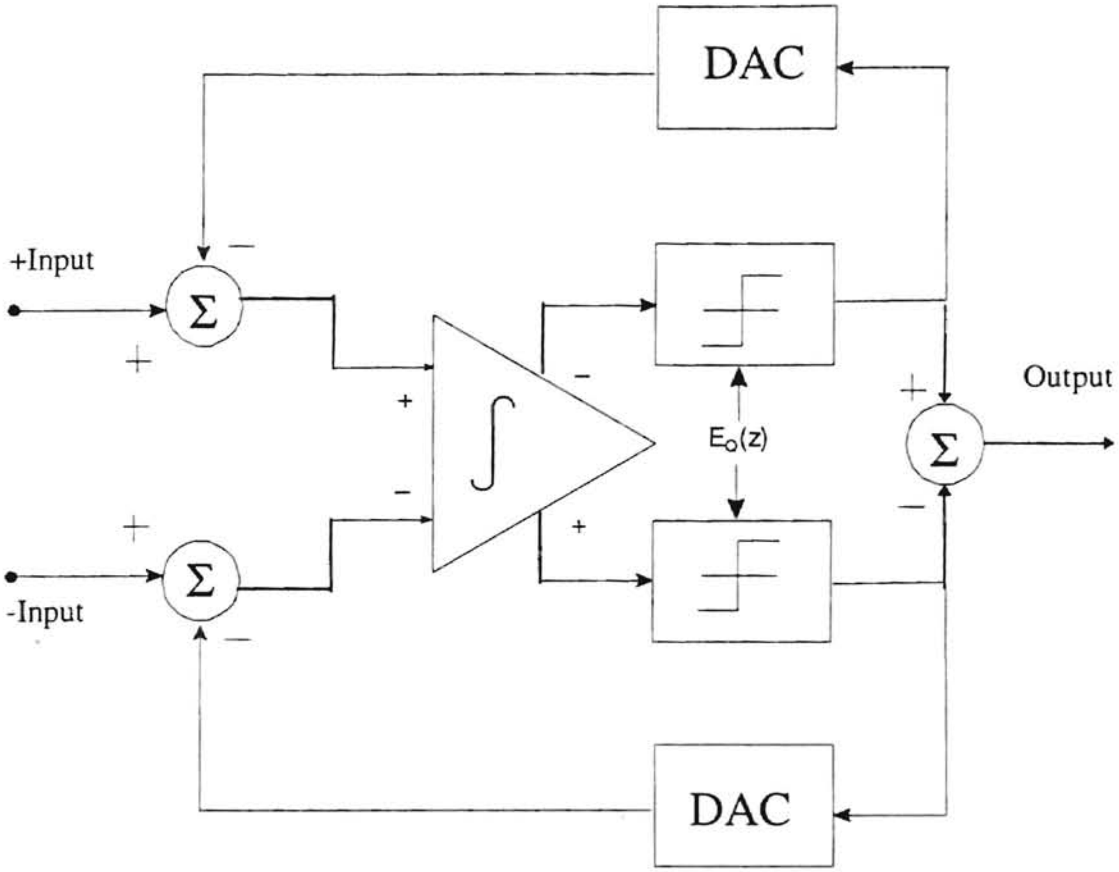


Figure 3.29 Block Diagram of the Differential form of a 1^{st}-Order ΣΔ Modulator.

Now with an understanding of 1^{st}-order ΣΔ modulator behavior and limitations, two higher order ΣΔ modulators are discussed in the next two sections of this work. Section 3.3 develops a 3^{rd}-order ΣΔ modulator which employs digital error correction functions to reduce the inband quantization noise power at the output of the modulator. The following section 3.4 explains a novel approach for reduction of the noise power by attempting to control the shape of the inband quantization noise power. This circuit, initially proposed by Nadeem, shapes the quantization noise in a Chebyshev Type-II transfer function form. Both these modulators are currently being investigated by the AAVDC for NRaD.

3.3 A 3^{rd} Order Residual Scaling ΣΔ Modulator

Higher ordered ΣΔ modulators allow for greater noise-shaping realization. One higher order ΣΔ architecture being developed by the AAVDC is a 3^{rd}-order Residual Scaling modulator. It proposes a greater reduction in quantization noise power over typical 3^{rd}-order ΣΔ modulators by use of digital error correction functions. A block diagram of this innovative architecture is given in Figure 3.30. This modulator does not follow the typical cascaded form described in Figure 3.2. Instead, a parallel structure is implemented with 3 ΣΔ loop stages.

From the figure, it is seen that the difference between the first stage's input and its estimated analog output before quantization, w_1 - x_1, is amplified and fed as the input to the second stage, x_2. Thus, the input to the second stage is basically the amplified noise that has been added to the first stage during its one loop ΣΔ modulation. Similarly, the noise added to the second stage by its one loop ΣΔ modulation is again amplified and fed as the input to the third stage. The interstage gains are required to amplify the previous stage's noise in order to maximize its A/D conversion process. That is, the interstage gains normalize the inputs to the second and third stages to utilize all of V_{FS}. After each stage completes its ΣΔ A/D conversion, the individual stage's digital signals, $y_{1,2,3}$, are fed through digital error correction functions, $H_{1,2,3}$, and summed to produce the final system output, y:

$$y = H_1 \cdot y_1 - H_2 \cdot y_2 + H_3 \cdot y_3 \tag{3.27}$$

When designed properly, these error correction functions tend to cancel the quantization noises form the first and second stages, $e_{1,2}$ [Walden, n.d.]. The noise remaining at the output is a scaled version of the quantization error introduced by the third stage. This error correction process is derived in Appendix B. The derived outputs of each stage are

$$y_1 = q \cdot x_1 + (1 - q) \cdot e_1 \tag{B.10}$$

$$y_2 = 0 \cdot x_1 - q^3 \cdot g_1 \cdot e_1 + (1 - q) \cdot e_2 \tag{B.12}$$

$$y_3 = 0 \cdot x_1 + 0 \cdot e_1 - q^3 \cdot g_2 \cdot e_2 + (1 - q) \cdot e_3 \tag{B.14}$$

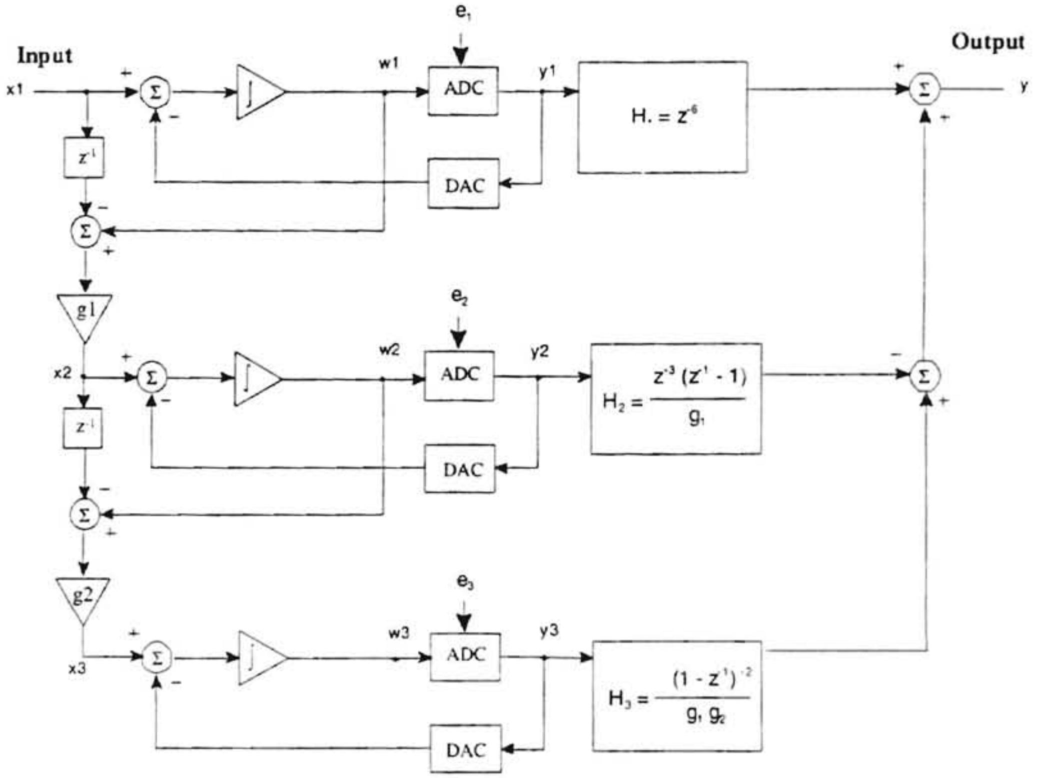


Figure 3.30 Block Diagram of a 3^{rd}-Order Residual Scaling ΣΔ Modulator using 4-Bit Quantizers and Digital Error Correction Functions.

where $q = z^{-1}$, x_i are the inputs to the i^{th} stage, e_i are the quantization errors associated with each i^{th} stage, and g_1 and g_2 are the interstage gains. Along with these, the proper error correction functions are also derived in Appendix B:

$$H_1 = q^6 \tag{B.19}$$

$$H_2 = q^3 \cdot \frac{(q-1)}{g_1} \tag{B.20}$$

$$H_3 = \frac{(1-q)^2}{g_1 \cdot g_2} \tag{B.21}$$

These functions when substituted into (3.27) yield the equation for the final output, y:

$$y = q^7 \cdot x_1 + 0 \cdot e_1 + 0 \cdot e_2 + \frac{(1-q)^3}{(g_1 \cdot g_2)} \cdot e_3 \tag{B.22}$$

(B.22) describes the output of the system to be a 7 sample delay of the input signal to the system added with a scaled, 3rd-order, noise-shaped quantization error form the third stage. (B.22) verifies the removal of the 2nd and 3rd stages' quantization noise from the output by use of the error correction functions.

From (B.22), the theoretical SNR for the system assuming an input of $(V_{FS}/2)\sin(\omega t)$ is:

$$SNR_{Residual} = \left(\frac{3}{16} \cdot 2^{2 \cdot B}\right) \cdot (g_1 \cdot g_2)^2 \cdot \frac{7 \cdot M^7}{\pi^6} \tag{3.28}$$

This theoretical SNR formula denotes the Residual Scaling ΣΔ architecture's dependence on the number of bits in the quantizers, B, the oversampling ratio, M, and the interstage gains, g_1 and g_2. (3.28) is similar to a typical 3rd-order $\Sigma\Delta$ modulator's SNR, calculated in (3.7), except that $SNR_{Residual}$ is reduced by the effect of the interstage gains, i.e., $(g_1g_2)^2$. This reduction is key to this architecture's proposed ability to further reduce the output's quantization noise power over a typical cascaded 3rd-order $\Sigma\Delta$ modulator.

This $\Sigma\Delta$ modulator was specified to achieve 16 bits of A/D conversion resolution utilizing 4-bit quantizers and operating with a sampling rate of 1 GSPS at 8 times oversampling. During the system-level design process, the only unspecified values were the interstage gains. These values were found by creating a Residual Scaling $\Sigma\Delta$ architecture file and utilizing the $\Sigma\Delta$ toolbox. Introducing a modulator input of $\pm V_{FS}$ white noise, the input to the first interstage gain, g_1, was viewed. It is shown in the upper plot in Figure 3.31. In order to normalize the output of the first interstage gain to V_{FS}, the inverse of the maximum error shown in the plot was determined. The maximum effective gain for g_1 was calculated from this to be 8. With this new value inserted as the first interstage gain in the architecture file, the same method was used to determine the second interstage gain, g_2. The lower plot in Figure 3.31 showing the input to the 2nd amplifier, g_2, was analyzed. It too resulted in a maximum effective gain for g_2 to be 8.

Therefore, substituting $B = 4$, $M = 8$, and $g_1 = g_2 = 8$ into (3.28) gives the theoretical SNR for the 3rd-order Residual Scaling $\Sigma\Delta$ modulator to be about 3×10^9 or 94.77 dB. The $\Sigma\Delta$ toolbox was used to verify this theoretical measurement. With all the system parameters prescribed, a 5000 sample *ideal* simulation was performed in 15 minutes on the same 486DX2-66 computer. The simulated performance results are given in Figure 3.32. The upper plot of the inband noise spectral density demonstrates the large

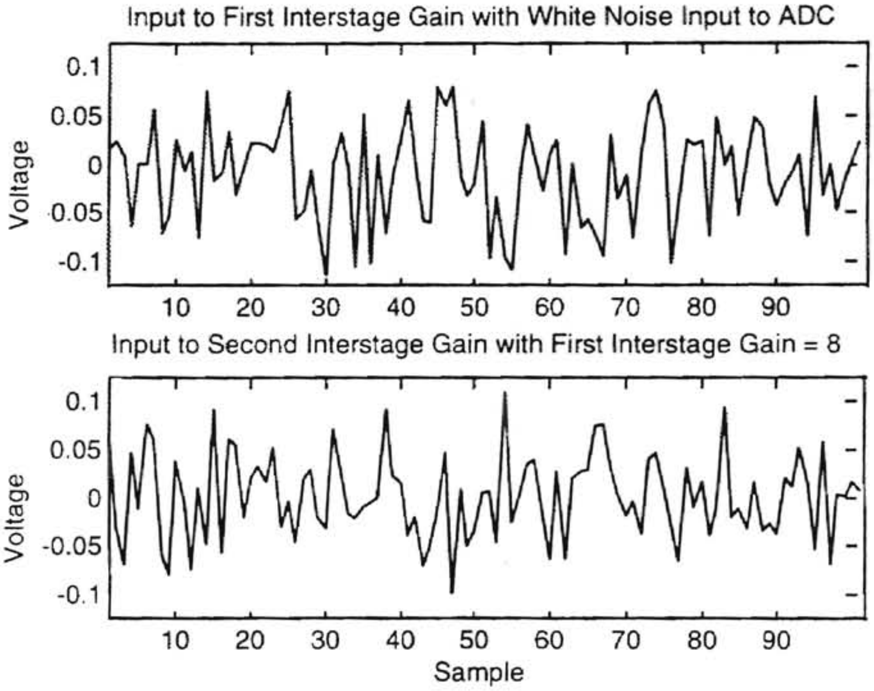


Figure 3.31 Interstage Gain Inputs for Gain Calculations in the 3rd-Order Residual Scaling ΣΔ Modulator using 4-Bit Quantizers.

noise-shaping characteristic of a 3rd-order ΣΔ modulator. This is also seen in the lower plot of the system's output spectrum. The simulated SNR of 93.80 dB matches the theoretical value to within 1 dB. The minor deviation between simulated and theoretical SNR values is probably the result of the limited number of sample points. A longer simulation of perhaps 32,000 sample points may allow the simulator to more precisely parallel the theoretical A/D conversion performance measurements.

As a whole, the ΣΔ toolbox gave an accurate measurement for this multi-bit, higher order architecture. In addition, the toolbox's ability to show intermediate architecture nodes aided in the design of this particular ΣΔ modulator. To complete the verification of the toolbox, one last higher order architecture is implemented in the next section.

3.4 A 3rd Order ΣΔ Modulator Based on the 'Nadeem' Interpolative Architecture

The previous section gave an example of a multi-bit, higher-order ΣΔ architecture that greatly attenuated the quantization noise in the passband. But, due to its interstage gains and multiple quantizers, it did not have an area efficient or low-power implementation. The architecture discussed in this section is proposed to be a low-power ΣΔ modulator and is more area efficient than the Residual Scaling ΣΔ modulator. The

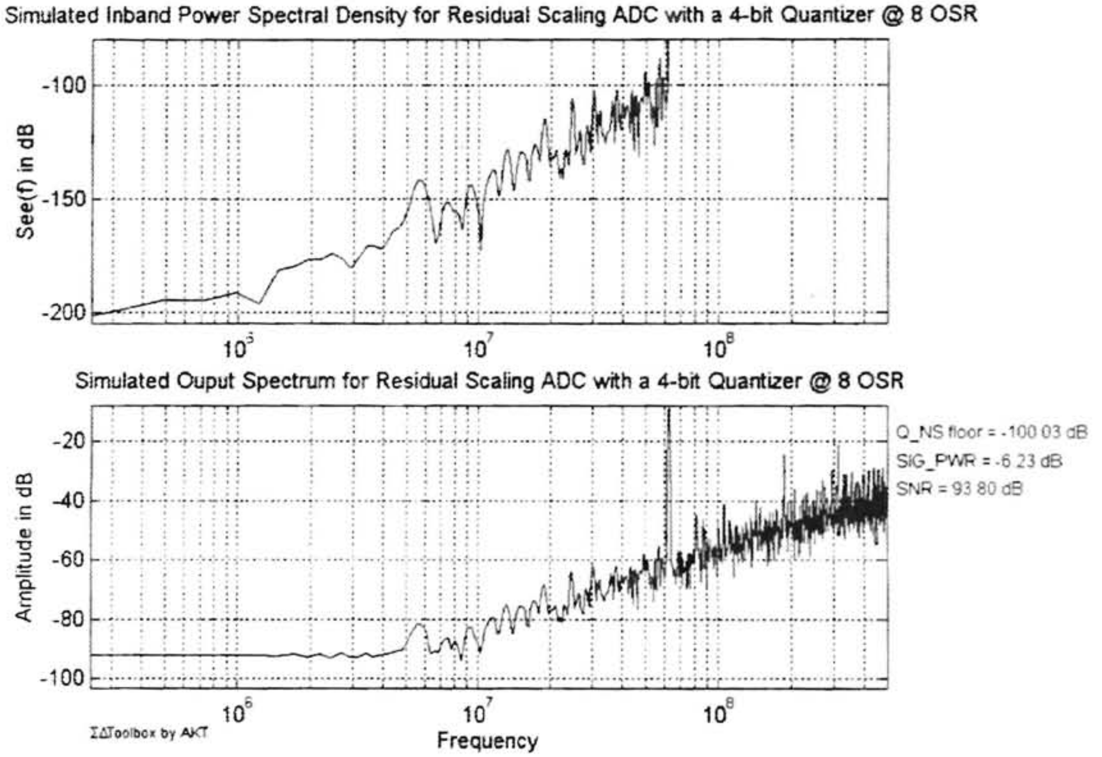


Figure 3.32 Simulated Performance Measurements for the 3^{rd}-Order Residual Scaling ΣΔ Modulator using 4-Bit Quantizers and Digital Error Correction Functions.

AAVDC has just begun investigation of this architecture. A similar architecture was introduced by S. Nadeem of the Massachusetts Institute of Technology that refines the noise-shaping process [Nadeem, 1994]. Common higher-order ΣΔ modulators designs have placed all the zeros of the NTF at 0 Hz and all the poles at $f_S/2$. This architecture, shown in Figure 3.33, refines the NTF into a Chebyshev Type-II filter form by feeding back scaled node outputs to intermediate nodes. With the proper design, these feedback coefficients are used to place the poles and zeros to achieve a the desired Chebyshev Type-II NTF response in order to reduce the average noise power within the band-of-interest. Specifications for the design being investigated by the AAVDC are 18 bits of A/D conversion resolution at 10 kSPS and 20 bits of resolution at 2 kSPS, using a 4-bit quantizer and a sampling rate of 1.28 MSPS with 64 times oversampling.

Appendix C analyzes Figure 3.33 and develops the feedback gains, $A_{0,1,2}$ and B_0. B_0 sets the zero, while a combination of $A_{0,1,2}$ and B_0 define the poles for the refined NTF. The integrator closed-loop gains also have a distinct effect on pole and zero placement.

On the whole, the stability of the system is strongly dependent on the values of these gains. The first step in calculating the gains is the development of the Chebyshev Type-II transfer function desired for the NTF noise-shaping. An Elliptical implementation was developed for easier zero and pole placement. Elliptical transfer functions allows for ripple in both the passband and stopband by proper placement of zeros and poles. Conversely, Chebyshev Type-II transfer functions allow for ripple only in the stopband by placement of zeros and no passband ripple by placing all poles at half the sampling frequency. Therefore, an Elliptical NTF with a 0.1 dB allowable ripple in the passband sufficiently models the Chebyshev Type-II NTF required for this design.

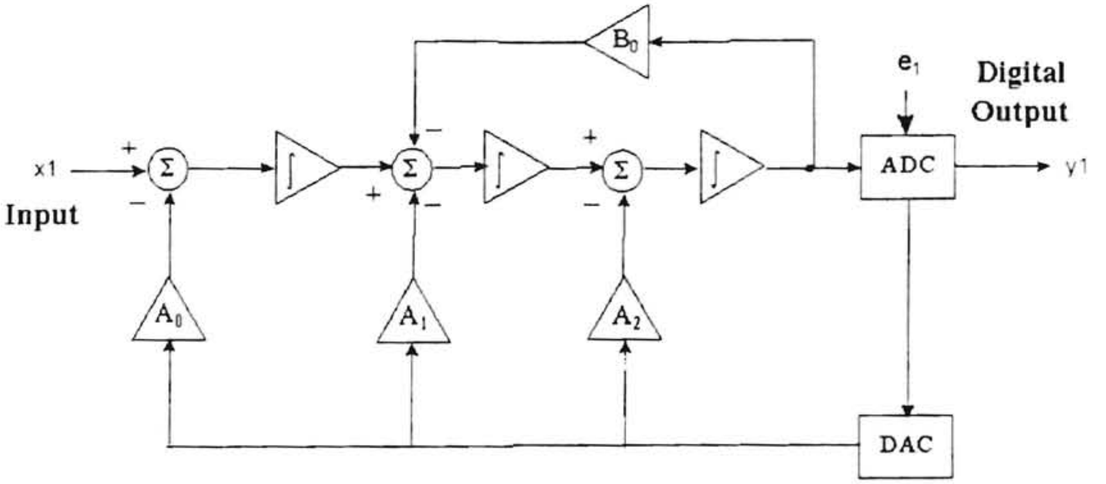


Figure 3.33 Block Diagram of 'Nadeem' Interpolative ΣΔ Modulator.

It is desired that a zero is placed at the passband frequency such that the NTF is attenuated at that point. This allows for a greater reduction of noise power at the modulator's output with respect to placing that zero at 0 Hz. In addition, the design of the Elliptical transfer function assumes a 1-bit quantizer implementation. Therefore, the desired transfer function is designed for an attenuation of the quantization noise floor of at least 15 bits or 90 dB. The remaining bits of resolution are accomplished using a 4-bit quantizer. Appendix C illustrates the process by determining the following optimal z-domain form from calculated Elliptical poles and zeros:

$$H_D(z)=\frac{\left(z^3-2.9975128z^2+2.99751276727696z-.99999996727696\right)}{\left(z^3-.192246117z^2+.3945465385431941z-1.32584740537294193450^{3}\right)} \qquad (C.3)$$

Both the normal Elliptical form has an associated transfer function constant that should be multiplied by $H_D(z)$. Unfortunately, this constant is not accounted for in Nadeem's interpolative ΣΔ modulator. Thus, the transfer function's constant multiplier is required to be unity. Using Figure 3.33, the z-domain representation of the interpolative ΣΔ modulator STF, $H_X(z)$, and NTF, $H_E(z)$, are calculated to be the following:

$$H_X(z)=\frac{K_2\cdot K_1\cdot K_0\cdot z}{\begin{array}{l} z^3+\left(-A_2\cdot K_2-3-K_2\cdot K_1\cdot B_0\right)\cdot z^2\ldots \\ +\left(-K_2\cdot K_1\cdot A_1+K_2\cdot K_1\cdot B_0+2\cdot A_2\cdot K_2+3\right)\cdot z\,.. \\ +\ K_2\cdot K_1\cdot K_0\cdot A_0+K_2\cdot K_1\cdot A_1-A_2\cdot K_2-1 \end{array}} \qquad (C.4)$$

$$H_E(z)=\frac{\left[z^3+\left(-3-K_2\cdot K_1\cdot B_0\right)\cdot z^2+\left(3+K_2\cdot K_1\cdot B_0\right)\cdot z-1\right]}{\left[\begin{array}{l} z^3+\left(-A_2\cdot K_2-3-K_2\cdot K_1\cdot B_0\right)\cdot z^2\ldots \\ +\left(-K_2\cdot K_1\cdot A_1+K_2\cdot K_1\cdot B_0+2\cdot A_2\cdot K_2+3\right)\cdot z\ldots \\ +-\left(K_2\cdot K_1\cdot K_0\cdot A_0\right)+K_2\cdot K_1\cdot A_1-A_2\cdot K_2-1 \end{array}\right]} \qquad (C.5)$$

where $A_{0,1,2}$ and B_0 are the feedback gains, and K_i is the closed-loop gain for the $(i+1)^{th}$ integrator. (C.5) clearly shows that the interpolative design does not have a NTF constant multiplier factor. As will be seen, this has an adverse effect on the proposed A/D conversion resolution.

The modulator's $H_E(z)$ has a direct relationship with $H_D(z)$. Matching the numerator and denominator z-coefficient-terms' from both functions and solving the resulting equations give the feedback gains in terms of the integrator gains. That is,

$$A = \begin{bmatrix} \dfrac{-1.2009745741378211781}{\left[K_2\cdot\left(K_1\cdot K_0\right)\right]} \\ \dfrac{-3.00756707182015412}{\left(K_2\cdot K_1\right)} \\ \dfrac{-2.80526665027696}{K_2} \end{bmatrix} \tag{C.10}$$

$$B_0 = -2.48723272304\frac{10^{-3}}{\left(K_2\cdot K_1\right)} \tag{C.11}$$

The next step in determining the coefficient values is to determine appropriate integrator gains, $K_{0,1,2}$. *The ΣΔ toolbox proved its usefulness in finding integrator gains which are suitable for (C.10) and (C.11) and will not cause instability within the ΣΔ modulator.* By developing an interpolative ΣΔ modulator architecture file using (C.10) and (C.11) and iterating for different integrator gains, the following integrator closed-loop gains were found as proper values:

$$K_0 = \frac{39}{40}; K_1 = \frac{39}{40}; K_2 = \frac{40}{40} \tag{C.12}$$

where, K_i is the $(i+1)^{th}$ integrator's closed-loop gain. (C.13) and (C.14) give the actual feedback coefficients after (C.12) has been substituted into (C.10) and (C.11). These values become

$$A_0 = -1.263; A_1 = -3.085; A_2 = -2.805 \tag{C.13}$$

$$B_0 = -0.003 \tag{C.14}$$

(C.12) through (C.14) give the gain values required by the interpolative architecture to achieve the Chebyshev Type-II NTF form. Note that these gains are usually limited to integer fractions for implementation as switched-capacitor ratio gain stages. Figures 3.34 and 3.35 show the theoretical plots for the modulator's NTF and STF, respectively, by substituting (C.12) through (C.14) into (C.4) and (C.5). Although the NTF plot of Figure 3.34 clearly shows the Chebyshev Type-II noise-shaping, the effects of forcing the transfer function's constant multiplier factor to unity is also evident. This theoretical NTF plot exhibits a gain greater than 0 dB at higher frequencies and a less pronounced attenuation around the zero placement at 10 kSPS. In addition, the STF plot also shows the unity multiplier's effect in a gain spike around 300 kSPS. Even with this problem, the theoretical solution developed seems to be optimal. This completes the derivation of the Interpolative ΣΔ modulator['s feedback and integrator gains. But before continuing on to a simulation, the ΣΔ toolbox requires an approximation for the STF of Figure 3.35 to determine the appropriate NSTerm to be integrated, as required by (3.11). An 11th-order power series expansion of (C.4) is used as a good approximation for the STF, $H_X(z)$, as seen in Appendix C:

$$\begin{aligned} H_x(z) = {} & .950625z^{-2} + .18351815625z^{-3} - .2695798511859375z^{-4} - .1241384723448827343z^{-5} \ldots \\ & + 5.9978746096105778759 \cdot 10^{-2} \cdot z^{-6} + 5.5156054204600368381 \cdot 10^{-2} \cdot z^{-7} \ldots \\ & + -6.8707020685983524388 \cdot 10^{-3} \cdot z^{-8} - 1.9856951083341120525 \cdot 10^{-2} \cdot z^{-9} \ldots \\ & + -2.3956232739070449332 \cdot 10^{-3} \cdot z^{-10} \end{aligned} \tag{C.17}$$

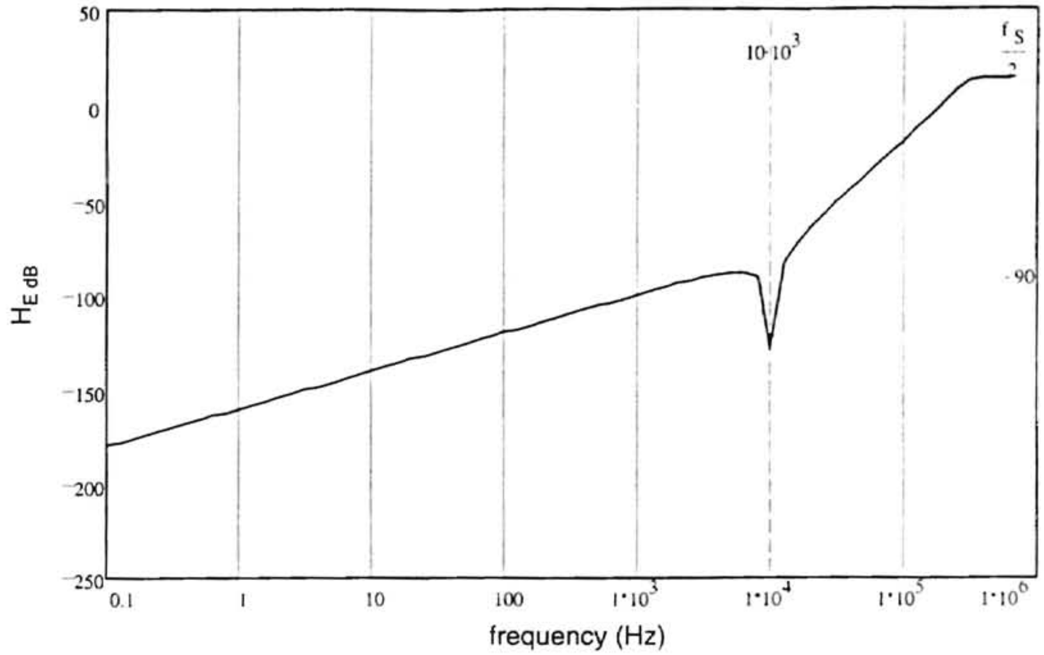


Figure 3.34 Quantization Noise Transfer Function for Interpolative 3rd-Order ΣΔ Modulator.

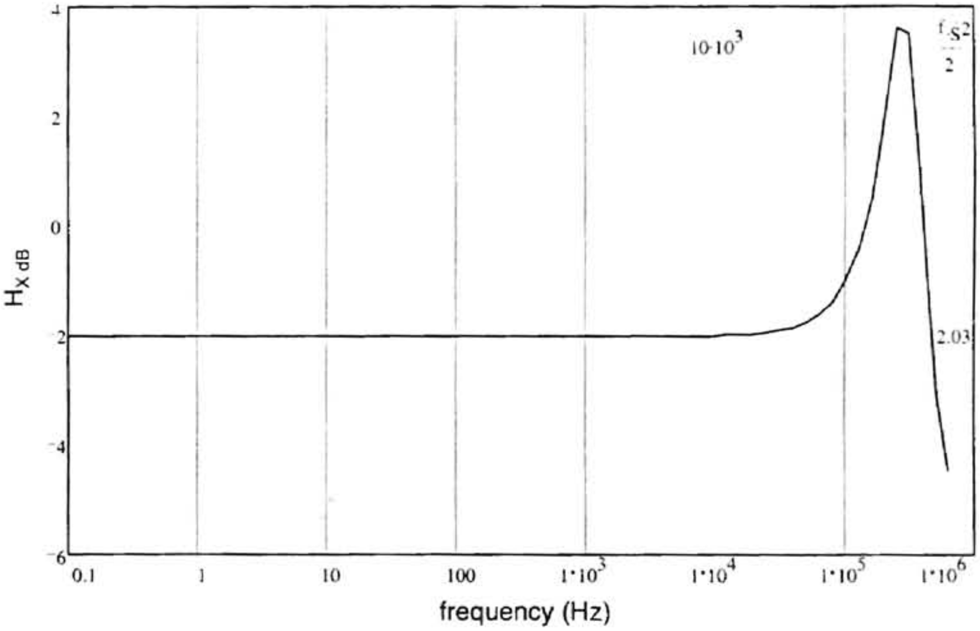


Figure 3.35 Signal Transfer Function for Interpolative 3rd-Order ΣΔ Modulator.

With the approximate STF defined, the gains found in (C.12) through (C.14) were substituted in the Interpolative ΣΔ modulator's architecture file. An *ideal* simulation was performed with a modulator input frequency of 2 kHz. The resulting outputs for each integrator and the final modulator output are respectively given in Figures 3.36, 3.37, 3.38, and 3.39 for a couple of cycles. These plots exemplify how the Interpolative ΣΔ modulator operates. Each consecutive integrator attempts to interpolate between its input waveform's values corresponding to adjacent digital quantization steps in an analog fashion. This process coupled with oversampling consequently produces a quantizer output that interpolates between adjacent quantization levels. Thus, the modulator's output provides a more precise estimation of the modulator input waveform. Also, the possible effects of the improper NTF multiplication factor is seen in these plots. As aforementioned, it is necessary that there is no integrator clipping in a ΣΔ modulator. Even though Figures 3.36 through 3.37 do not show any clipping, the output's of the 1st and 2nd integrators are not within the quantizer's full-scale voltage set at 1V. for this simulation. This may be a cause for the deviation of the simulated modulator's SNR from the theoretical SNR seen below.

To determine this architecture's simulated SNR for bandwidths of 2 kSPS and 10 kSPS, an *ideal* 32,000 sample simulation was performed with f_S = 1.28 MSPS and M = 64. A modulator input of 50 kHz was introduced to reduce the simulation time and to rid the resulting performance plots of harmonic contributions around the desired bandwidths.

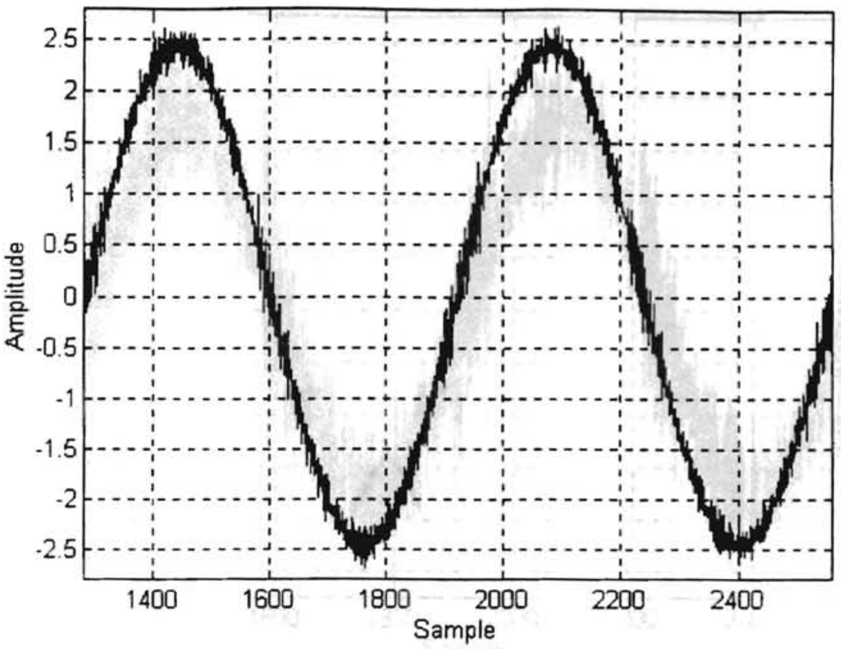


Figure 3.36 Output of 1st Integrator of Interpolative ΣΔ Modulator with 1V., 2 kHz Sine Modulator Input at f_s=1.28 MSPS and M=64.

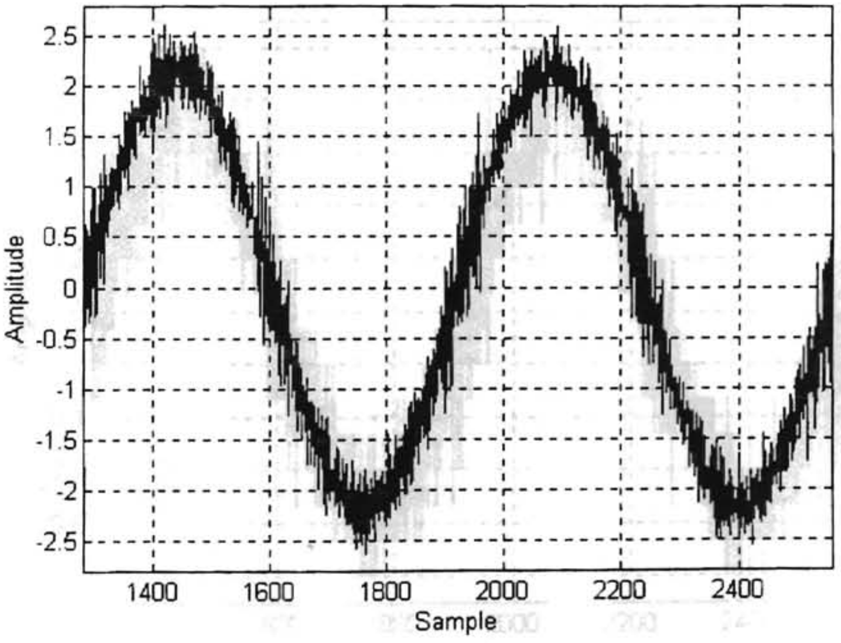


Figure 3.37 Output of 2nd Integrator of Interpolative ΣΔ Modulator with 1V., 2 kHz Sine Modulator Input at f_s=1.28 MSPS and M=64.

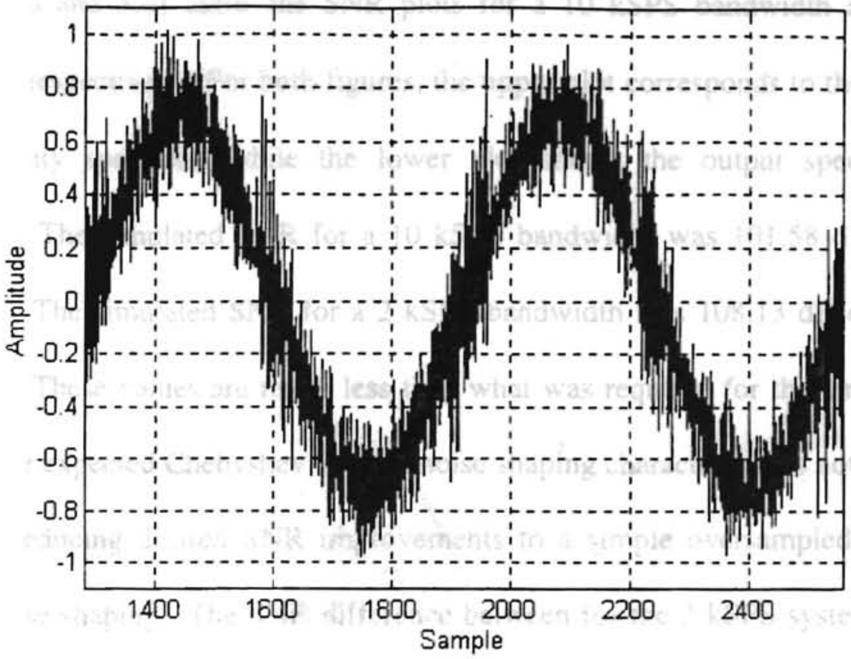


Figure 3.38 Output of 3rd Integrator of Interpolative ΣΔ Modulator with 1V., 2 kHz Sine Modulator Input at f_s=1.28 MSPS and M=64.

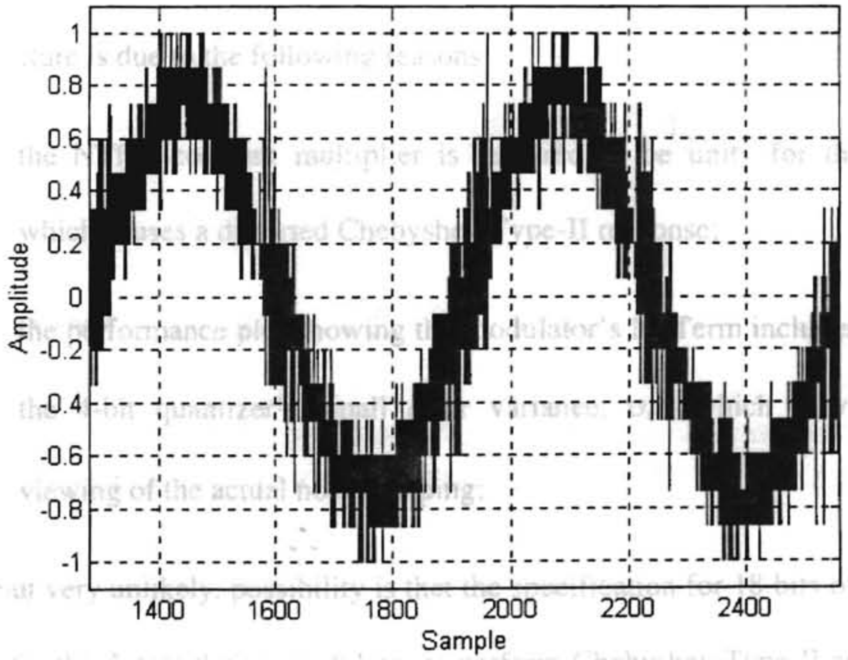
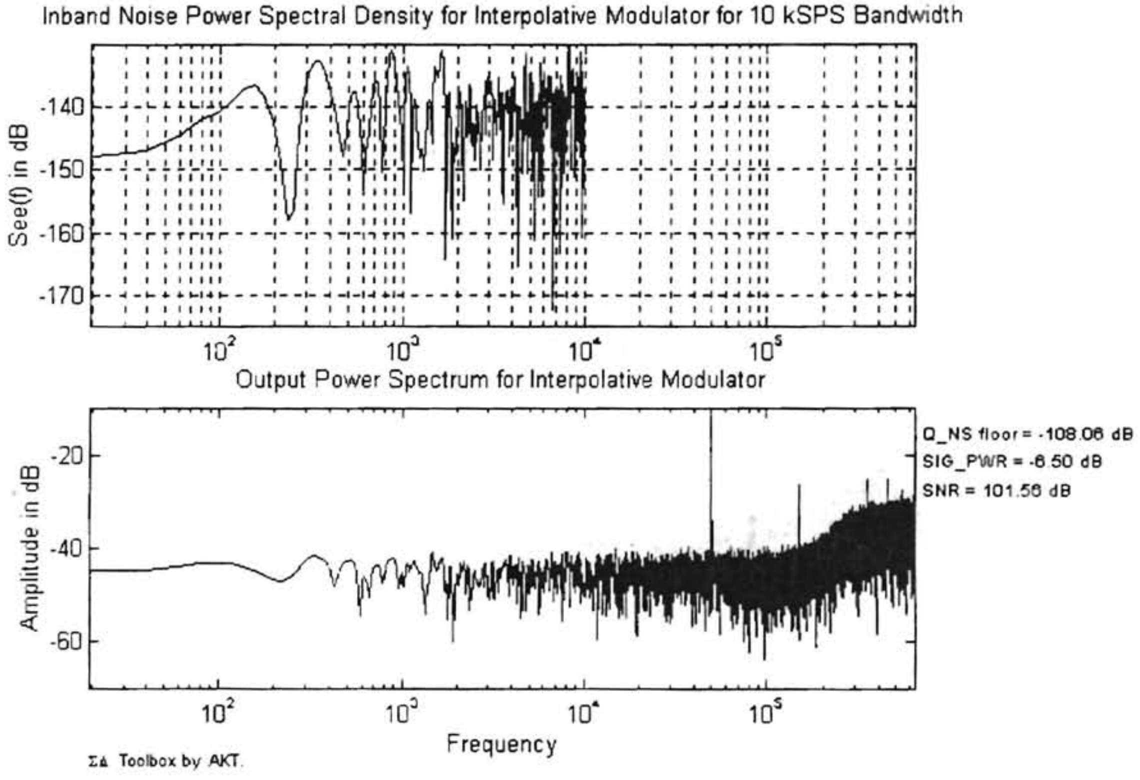


Figure 3.39 Output of Interpolative ΣΔ Modulator with 1V., 2 kHz Sine Modulator Input at f_s=1.28 MSPS and M=64.

Figures 3.40 and 3.41 show the SNR plots for a 10 kSPS bandwidth and a 2 kSPS bandwidth, respectively. For both figures, the upper plot corresponds to the inband noise power density spectrum, while the lower plot shows the output spectrum for the modulator. The simulated SNR for a 10 kSPS bandwidth was 101.58 dB or 16.6 bits resolution. The simulated SNR for a 2 kSPS bandwidth was 108.13 dB or 17.7 bits of resolution. These values are much less than what was required for this architecture. In addition, the expected Chebyshev Type-II noise shaping characteristic is not seen in either plot, thus reducing desired SNR improvements to a simple oversampled improvement without noise shaping. The SNR difference between for the 2 kSPS system over the 10 kSPS system, i.e. by a simulated 6.55 dB, is solely due to oversampling the system by 5 times more. Using equation (2.14), oversampling a system by 5 times yields a 6.98 dB improvement which concurs with the above observation. The lack of performance from this architecture is due to the following reasons:

- the NTF's constant multiplier is required to be unity for this architecture which causes a distorted Chebyshev Type-II response;
- the performance plot showing the modulator's NSTerm includes the effects of the 4-bit quantizer's small error variance, σ_e^2, which may prevent easy viewing of the actual noise-shaping;

Another, but very unlikely, possibility is that the specification for 18 bits of SNR exceeds the ability for this Interpolative modulator to perform Chebyshev Type-II noise-shaping at



Inband Noise Power Spectral Density for Interpolative Modulator for 10 kSPS Bandwidth

Output Power Spectrum for Interpolative Modulator

Figure 3.40 Simulated Performance Measurements for the Interpolative ΣΔ Modulator using a 4-Bit Quantizer for a 10 kSPS Bandwidth.

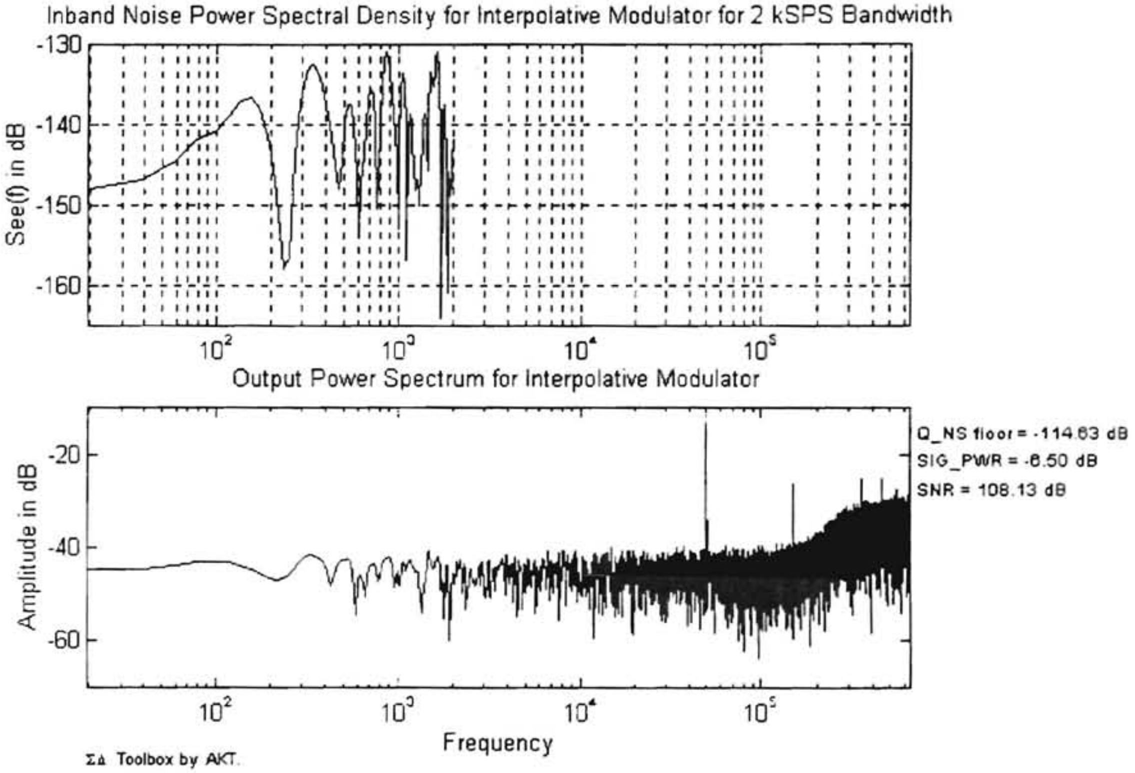


Figure 3.41 Simulated Performance Measurements for the Interpolative ΣΔ Modulator using a 4-Bit Quantizer for a 2 kSPS Bandwidth.

M = 64 and f_S = 1.28 MSPS. These toolbox results suggest further investigation and refinement of Nadeem's Interpolative ΣΔ modulator is necessary to resolve the apparent problems.

The ΣΔ toolbox aided the initial investigation of Nadeem's Interpolative ΣΔ modulator. It verified that at least two problems to exist and that further system-level examination is required before actual IC designs are implemented. This section along with the previous has proved the utility of the ΣΔ toolbox. Higher-order architectures can be simulated and refined using the ΣΔ toolbox to determine their A/D conversion potential.

Chapter 4 Conclusions

Interest in oversampled, ΣΔ modulation has recently grown due to its reported increase in A/D conversion resolution over conventional A/D conversion methods. ADCs employing ΣΔ modulation are becoming commonplace due to their simpler design and resilience to limited device matching. The verification of the theoretical performance of novel ΣΔ modulators has been hindered by inefficient or incomplete simulations provided by common commercially available simulators. This research work described the development of a new rapid-prototyping simulator, the ΣΔ toolbox, that attempts to overcome these hindrances.

After an introductory discussion of Nyquist-rate and conventional oversampled ADCs, ΣΔ modulation, its characteristic noise-shaping properties, and implementation of ΣΔ modulators in the ΣΔ toolbox was discussed. The toolbox can implement a variety of ΣΔ architectures by use of the following modular component blocks: integrator, quantizer, DAC, summation node, and amplifier. Following the ideal description of each block, the effects of six significant non-idealities which hamper A/D conversion were investigated. The non-idealities considered were: integrator harmonic distortion errors, block component settling errors, MOSFET switch charge injection errors, clock *jitter* errors, circuit component mismatch errors, and circuit common-mode errors. The incorporation of these non-idealities allows for more realistic A/D conversion performance measurements by the ΣΔ toolbox. Each non-ideality's effect on ΣΔ modulation was presented individually with a 1st-order ΣΔ modulator example. Simulations from the ΣΔ toolbox were performed with respectable efficiency on a IBM-compatible 486DX2-66 computer. With the discussion of the basic simulator complete, the toolbox's utility was seen in the initial investigations of two 3rd-order ΣΔ modulators being developed by the AAVDC for NRaD.

The first higher-order architecture described was the 3rd-order Residual Scaling ΣΔ modulator. This novel architecture uses digital error correction functions on the parallel, 3rd-order ΣΔ modulator to achieve a greater reduction in quantization noise power over a cascaded, 3rd-order ΣΔ modulator. The ΣΔ toolbox was useful in the development of this design by providing both frequency and time-domain views of intermediate nodes. This was integral in defining the maximum values for the interstage gains. Also, the toolbox verified the derived error cancellation functions and the modulator's theoretical SNR by use of simulated performance measurements.

The second higher-order architecture presented was a 3rd-order Interpolative ΣΔ modulator similar to a design reported by S. Nadeem of MIT. This interesting architecture attempts to refine ΣΔ modulation noise-shaping by modifying the NTF into a Chebyshev Type-II form. After mathematical derivation of the architecture's required feedback gains, the required integrator gains were found by viewing integrator outputs for stability using the ΣΔ toolbox. A problem with the design was verified by the toolbox's simulation results. Thus, the ΣΔ toolbox once again proved its utility in the initial system-level design of ΣΔ modulators.

Noting the potential of this simulator along with the increased use of ΣΔ modulators, a variety of future prospects for the ΣΔ toolbox are possible. First, a graphical or user-friendly interface must be integrated into the simulator. Second, power measurements could be introduced. This could further aid the designer in refining potential low-power designs. Lastly, the simulator could be developed into a type of VLSI hardware description language in order to further provide even faster prototyping of potential designs.

Bibliography

1. Boser, B. E., and B. A. Wooley. "The design of Sigma-Delta Modulation Analog-to-Digital Converters." *IEEE Journal of Solid-State Circuits*, Vol. 23, No. 6, pp. 1298-1308, December 1988.

2. Candy, J. C. "A use of Double Integration in Sigma Delta Modulation." *IEEE Transactions on Communications,* Vol. COM-33, No. 3, pp. 249-258, March 1985.

3. Chao, K. C. H., C. G. Sodini, S. Nadeem, and W. L. Lee. "A Higher Order Topology for Interpolative Modulators for Oversampling A/D Converters." *IEEE Journal of Solid-State Circuits*, Vol. 37, No. 3, pp. 309-318, March 1990.

4. Dias, V. F., G. Palmisano, P. O'Leary, and F. Maloberti. "Fundamental Limitations of Switched-Capacitor Sigma-Delta Modulators." *IEE Proceedings-G*, Vol. 139, No. 1, pp. 27-32, February 1992.

5. Gersho, A. "Principles of Quantization." *IEEE Communications Society Magazine,* Vol. 15, No. 5, pp. 16-29, September 1977.

6. Groenewold, G. and W. J. Lubbers. "Systematic Distortion Analysis for MOSFET Integrators with Use of a New MOSFET Model. " *IEEE Transactions on Circuits and Systems -- II*, Vol. 41, No. 9, pp. 569-580, September 1994.

7. Laker, K. R., and W. M. C. Sansen. *Design of Analog Integrated Circuits and Systems.* McGraw-Hill, Inc., New York, NY, 1994.

8. Ludeman, L. C. *Fundamentals of Analog Digital Signal Processing.* John Wiley & Sons, New York, NY, 1986.

9. Nadeem, S., C. G. Sodini, and H. S. Lee. "16-Channel Oversampled Analog-to-Digital Converter." *IEEE Journal of Solid-State Circuits*, Vol. 29, No. 9, pp. 1077-1084, September 1994.

10. Oppenhiem, A. V., and R. W. Schafer. *Discrete-Time Signal Processing.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1989.

11. Park, S. "Principles of Sigma-Delta Modulation for Analog-to Digital Converters." Motorola Literature Distribution Center, Phoenix, AZ.

12. Rabiner, L. R., and R. W. Schafer. *Digital Processing of Speech Signals.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1978.

13. Van De Plassche, R. *Integrated Analog-to-Digital and Digital-to-Analog Converters.* Kluwer Academic Publishers, Boston, MA, 1994.

14. Walden, R. H., T. Cataltepe, and G. C. Temes. "Architectures for Higher-Order Multibit ΣΔ Modulators." Naval Ocean Systems Center, San Diego, CA, n.p., n.d.

15. Wegmann, G., E. A. Vittoz, and K. Rahali. "Charge Injection on Analog MOS Switches." *IEEE Journal of Solid-State Circuits*, Vol. SC-22, No. 6, pp. 1091-1097, December 1987.

16. Williams, L. A., and B. A. Wooley. "MIDAS - A Functional Simulator for Mixed Digital and Analog Sampled Data Systems." *1992 IEEE International Symposium on Circuits and Systems,* pp. 2148-2151, May 1992.

17. Williams, L. A. "Modeling and Design of High-Resolution Sigma-Delta Modulators." Ph.D. Dissertation, Stanford University, CA, 1993.

Appendices

Appendix A

Development of Settling Error for Two-Pole Representation of an Integrator

Below is the two-pole, small-signal model used for the integrator in the ΣΔ toolbox. By using this mathematical model, the time constants associated with the integrator may be found.

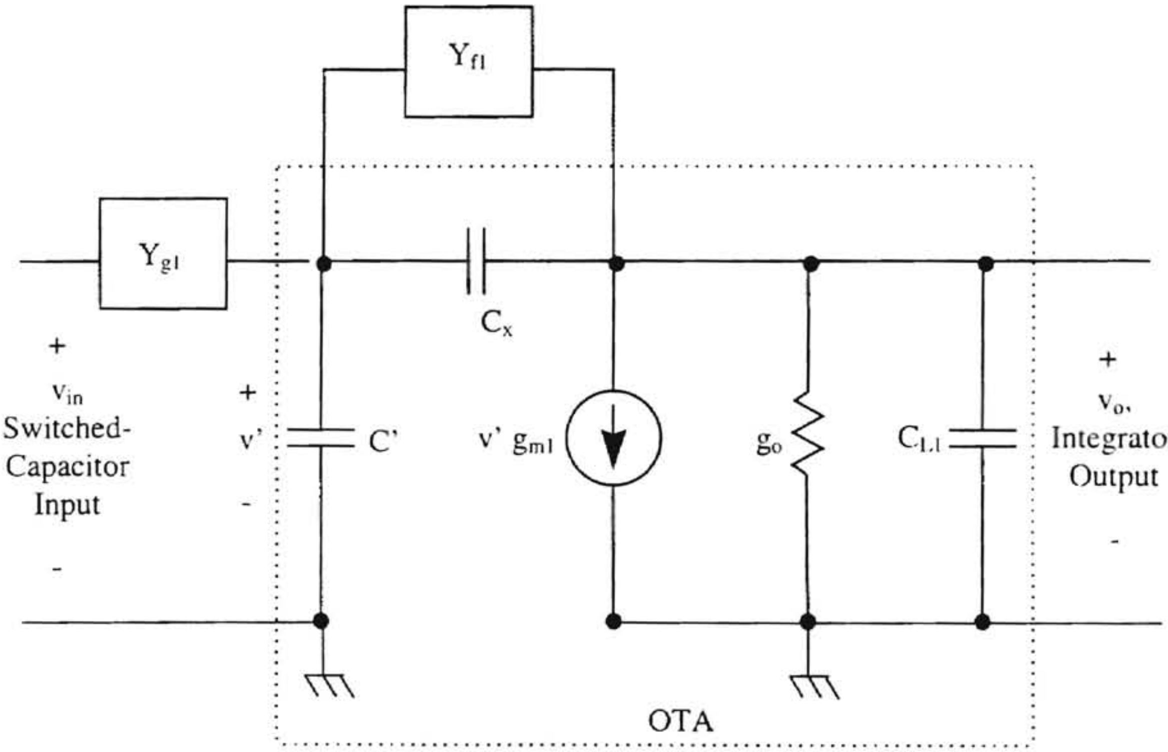


Figure A.1 Schematic of Two-Pole, Small-Signal Integrator Model.

These time constants are combined to determine the settling time required for integration. Incomplete integration will cause an integration settling error which is accounted for in the ΣΔ toolbox. Using Kirchhoff's Current Law at the gate and drain nodes, (A.1) and (A.2) are found.

$$(v' - v_{in}) \cdot Y_{g1} + v' \cdot C' \cdot s + (v' - v_o) \cdot (s \cdot C_x + Y_{f1}) = 0 \qquad \text{(A.1)}$$

$$v_o \cdot (g_o + s \cdot C_{L1}) + g_{m1} \cdot v' + (v_o - v') \cdot (s \cdot C_x + Y_{f1}) = 0 \qquad \text{(A.2)}$$

where, v_{in} is the switched-capacitor input voltage; v' is the gate to source voltage; v_o is the output voltage; C' is the OTA input parasitic capacitance; C_x is the gate to output parasitic capacitance; C_{L1} is the load capacitance; g_{m1} is the OTA's effective transconductance; Y_{g1} is the switch's admittance; Y_{f1} is the integration capacitor's admittance; and s is the Laplacian operator.

(A.2) can be rewritten as :

$$(g_{m1} - s \cdot C_x - Y_{f1}) \cdot v' + \left[(g_o + s \cdot C_{L1} + s \cdot C_x + Y_{f1}) \cdot v_o \right] = 0 \qquad \text{(A.3)}$$

(A.1) can be rewritten as:

$$v' = \frac{(Y_{g1} \cdot v_{in} + v_o \cdot C_x \cdot s + v_o \cdot Y_{f1})}{(Y_{g1} + s \cdot C' + s \cdot C_x + Y_{f1})} \qquad \text{(A.4)}$$

Substituting (A.4) into (A.3) gives,

$$(g_{m1} - s \cdot C_x - Y_{f1}) \cdot \frac{(Y_{g1} \cdot v_{in} + v_o \cdot C_x \cdot s + v_o \cdot Y_{f1})}{(Y_{g1} + s \cdot C' + s \cdot C_x + Y_{f1})} = 0$$

which yields:

$$\left\{\begin{array}{l}\left[(g_{m1}-s\cdot C_x-Y_{f1})\cdot\dfrac{(s\cdot C_x+Y_{f1})}{(Y_{g1}+s\cdot C'+s\cdot C_x+Y_{f1})}+g_o+s\cdot C_{L1}+s\cdot C_x+Y_{f1}\right]\cdot v_o\ldots\\ +(g_{m1}-s\cdot C_x-Y_{f1})\cdot\dfrac{(Y_{g1}\cdot v_{in}+v_o\cdot C_x\cdot s+v_o\cdot Y_{f1})}{(Y_{g1}+s\cdot C'+s\cdot C_x+Y_{f1})}\end{array}\right\}=0$$

Solving this equation for the integrator's voltage gain, A_V, gives

$$A_v=\frac{v_o}{v_{in}}=\frac{(g_{m1}-s\cdot C_x-Y_{f1})\cdot\dfrac{-Y_{g1}}{(Y_{g1}+s\cdot C'+s\cdot C_x+Y_{f1})}}{\left[(g_{m1}-s\cdot C_x-Y_{f1})\cdot\dfrac{(s\cdot C_x+Y_{f1})}{(Y_{g1}+s\cdot C'+s\cdot C_x+Y_{f1})}+g_o+s\cdot C_{L1}+s\cdot C_x+Y_{f1}\right]}$$

C_x is neglected since it is much smaller than C_{int}. This simplification results in

$$A_v=\frac{(g_{m1}-Y_{f1})\cdot\dfrac{-Y_{g1}}{(Y_{g1}+s\cdot C'+Y_{f1})}}{\left[(g_{m1}-Y_{f1})\cdot\dfrac{(Y_{f1})}{(Y_{g1}+s\cdot C'+Y_{f1})}+g_o+s\cdot C_{L1}+Y_{f1}\right]}$$

which may be rewritten as

$$A_v=Y_{g1}\cdot\frac{(-g_{m1}+Y_{f1})}{\left[\begin{array}{l}s^2\cdot C_{L1}\cdot C'+(g_o\cdot C'+C_{L1}\cdot Y_{g1}+C_{L1}\cdot Y_{f1}+Y_{f1}\cdot C')\cdot s\ldots\\ +g_{m1}\cdot Y_{f1}+g_o\cdot Y_{g1}+g_o\cdot Y_{f1}+Y_{f1}\cdot Y_{g1}\end{array}\right]} \qquad (A.5)$$

Now, the full form for the admittance Y_{f1} & Y_{g1} are substituted into (A.5). Y_{g1} and Y_{f1} are

$$Y_{g1} = \frac{1}{\left(R_{g1} + \frac{1}{s \cdot C_{g1}}\right)} = \frac{C_{g1}}{\left(R_{g1} \cdot s \cdot C_{g1} + 1\right)} \tag{A.6}$$

$$Y_{g1} = s \cdot C_{int} \tag{A.7}$$

where, R_{g1} is the effective switch resistance; C_{g1} is the effective switch parasitic capacitance; and C_{int} is the integration capacitor. Thus (A.5) becomes:

$$A_v = Y_{g1} \cdot \frac{(-g_{m1} + C_{int} \cdot s)}{\begin{bmatrix} (C_{L1} \cdot C') \cdot s^2 + (g_o \cdot C' + C_{L1} \cdot Y_{g1} + C_{L1} \cdot C_{int} \cdot s + C_{int} \cdot C' \cdot s) \cdot s \ldots \\ + g_{m1} \cdot C_{int} \cdot s + g_o \cdot Y_{g1} + g_o \cdot C_{int} \cdot s + C_{int} \cdot Y_{g1} \cdot s \end{bmatrix}}$$

or,

$$A_v = (-g_{m1} + C_{int} \cdot s) \cdot \frac{C_{g1}}{\begin{bmatrix} (C_{L1} \cdot C_{int} \cdot R_{g1} \cdot C_{g1} + C' \cdot C_{int} \cdot R_{g1} \cdot C_{g1} + C_{L1} \cdot C' \cdot R_{g1} \cdot C_{g1}) \cdot s^2 \ldots \\ + \begin{pmatrix} C' \cdot C_{L1} + C_{L1} \cdot C_{g1} + C_{int} \cdot g_{m1} \cdot R_{g1} \cdot C_{g1} + g_o \cdot C' \cdot R_{g1} \cdot C_{g1} \ldots \\ + C_{int} \cdot C' + C_{L1} \cdot C_{int} + C_{int} \cdot g_o \cdot R_{g1} \cdot C_{g1} + C_{int} \cdot C_{g1} \end{pmatrix} \cdot s \ldots \\ + C_{int} \cdot g_o + C_{int} \cdot g_{m1} + g_o \cdot C' + g_o \cdot C_{g1} \end{bmatrix}}$$

Deleting insignificant terms from this expanded equation gives the following equation used to determine the integrator's settling time constants:

$$A_v = \frac{\dfrac{C_{g1}\cdot(-g_{m1}+C_{int}\cdot s)}{(C_{int}\cdot g_{m1})}}{\dfrac{\left[\begin{array}{l}(C_{L1}\cdot C_{int}\cdot R_{g1}\cdot C_{g1}+C_{int}\cdot C'\cdot R_{g1}\cdot C_{g1}+C_{L1}\cdot C'\cdot R_{g1}\cdot C_{g1})\cdot s^2\ldots\\ \left(\begin{array}{l}C_{L1}\cdot C'+C_{L1}\cdot C_{g1}+C_{int}\cdot g_{m1}\cdot R_{g1}\cdot C_{g1}\ldots\\ +C_{int}\cdot C'+C_{L1}\cdot C_{int}+C_{int}\cdot C_{g1}\end{array}\right)\cdot s+C_{int}\cdot g_{m1}\end{array}\right]}{C_{int}\cdot g_{m1}}} \qquad (A.8)$$

The denominator of (A.8) is

$$\frac{\left[\begin{array}{l}(C_{L1}\cdot C_{int}\cdot R_{g1}\cdot C_{g1}+C_{int}\cdot C'\cdot R_{g1}\cdot C_{g1}+C_{L1}\cdot C'\cdot R_{g1}\cdot C_{g1})\cdot s^2\ldots\\ \left(\begin{array}{l}C_{L1}\cdot C'+C_{L1}\cdot C_{g1}+C_{int}\cdot g_{m1}\cdot R_{g1}\cdot C_{g1}\ldots\\ +C_{int}\cdot C'+C_{L1}\cdot C_{int}+C_{int}\cdot C_{g1}\end{array}\right)\cdot s+C_{int}\cdot g_{m1}\end{array}\right]}{C_{int}\cdot g_{m1}} \qquad (A.9)$$

With each value provided in a particular ΣΔ modulator architecture file inserted in (A.9), the ΣΔ toolbox numerically solves for the Laplacian operator, s. The inverse of these roots are the 2 settling time constants associated with the integrator. That is, (A.9) is calculated to be of the form:

$$(s \quad p_1)\cdot(s - p_2) \qquad (A.10)$$

where p_1 and p_2 are the calculated poles of the denominator. (A.10) gives the final result for the settling time constants τ_1 and τ_2:

$$\tau_1 = \frac{1}{p_1} \qquad (A.11)$$

$$\tau_2 = \frac{1}{p_2} \qquad (A.12)$$

These settling time constants are used to determine if complete settling of the integrator output is achieved within the specified clock phase time as discussed in Section 3.2. If not, a settling error arises and is incorporated in the output voltage of the integrator.

Appendix B

Development of Residual Scaling Error Removal Functions

The Appendix develops the Error Cancellation Functions, H_1, H_2, and H_3 beginning with the definition of each stage's input, intermediate, and output equations. These equations were entered and solved in Mathcad. Due to Mathcad's habit of simplification of inverse exponentials to fractions, the following substitution is used throughout the Appendix:

$$q = z^{-1}$$

The intermediate node equations are

$$w_1 = \frac{q}{1-q} \cdot \left(x_1 \quad y_1\right) \tag{B.1}$$

$$w_2 = \frac{q}{1-q} \cdot \left(x_2 - y_2\right) \tag{B.2}$$

$$w_3 = \frac{q}{1-q} \cdot \left(x_3 - y_3\right) \tag{B.3}$$

Later, the following substitutions will be made to determine the effects of mismatches in analog gains, γ_i, and digital gains, g_i:

$$g_1 = \gamma_1 \cdot (1 + \delta_1) \qquad g_2 = \gamma_2 \cdot (1 + \delta_2)$$

Initially, the development of each stage's input equations is done.

$$x_1 = \text{input of the } \Sigma\Delta \text{ Residual Scaling Modulator} \tag{B.4}$$

$$x_2 = g_1 \cdot q \cdot (w_1 - x_1 \cdot q) \tag{B.5}$$

Substituting (B.1) and (B.4) into (B.5):

$$x_2 = g_1 \cdot q \cdot \left[\frac{q}{(1-q)} - q \right] \cdot x_1 - g_1 \cdot \frac{q^2}{(1-q)} \cdot y_1$$

$$x_2 = -g_1 \cdot q^3 \cdot \frac{x_1}{(-1+q)} - g_1 \cdot \frac{q^2}{(1-q)} \cdot y_1 \tag{B.6}$$

$$x_3 = g_2 \cdot q \cdot (w_2 - x_2 \cdot q) \tag{B.7}$$

Substituting (B.2), (B.4), and (B.6) into (B.7):

$$x_3 = g_2 \cdot q \cdot \left[\frac{q}{(1-q)} - q \right] \cdot \left[-g_1 \cdot q^3 \cdot \frac{x_1}{(-1+q)} - g_1 \cdot \frac{q^2}{(1-q)} \cdot y_1 \right] - g_2 \cdot \frac{q^2}{(1-q)} \cdot y_2$$

$$x_3 = -g_2 \cdot q^5 \cdot g_1 \cdot \frac{y_1}{(-1+q)^2} + g_2 \cdot q^6 \cdot g_1 \cdot \frac{x_1}{(-1+q)^2} - g_2 \cdot \frac{q^2}{(1-q)} \cdot y_2 \tag{B.8}$$

Next, the output equations for each stage are developed:

$$y_1 = w_1 + e_1 \tag{B.9}$$

Substituting (B.1) and (B.4) into (B.9) yields:

$$y_1 = q \cdot x_1 + e_1 - e_1 \cdot q$$

$$y_1 = q \cdot x_1 + (1 - q) \cdot e_1 \tag{B.10}$$

$$y_2 = w_2 + e_2 \tag{B.11}$$

Substituting (B.1) - (B.6) into (B.11) gives:

$$y_2 = q \cdot \left[-g_1 \cdot \frac{q^3}{(-1 + q)} - g_1 \cdot \frac{q^3}{(1 - q)} \right] \cdot x_1 - q^3 \cdot g_1 \cdot e_1 + (1 - q) \cdot e_2$$

$$y_2 = 0 \cdot x_1 - q^3 \cdot g_1 \cdot e_1 + (1 - q) \cdot e_2 \tag{B.12}$$

$$y_3 = w_3 + e_3 \tag{B.13}$$

Substituting (B.1) - (B.7) into (B.13) gives:

$$y_3 = \frac{q}{(1 \quad q)} \cdot \left(x_3 - y_3 \right) + e_3$$

$$y_3 = q \cdot x_3 + (1 - q) \cdot e_3$$

$$y_3 = q \cdot \left[-g_2 \cdot q^5 \cdot g_1 \cdot \frac{y_1}{(-1 + q)^2} + g_2 \cdot q^6 \cdot g_1 \cdot \frac{x_1}{(\ 1 + q)^2} \quad g_2 \cdot \frac{q^2}{(1 - q)} \cdot y_2 \right] + (1 \quad q) \cdot e_3$$

$$y_3 = q \cdot \left[-g_2 \cdot q^5 \cdot g_1 \cdot \frac{(1 \quad q)}{(\ 1 + q)^2} + g_2 \cdot \frac{q^5}{(1 - q)} \cdot g_1 \right] \cdot e_1 - q^3 \cdot g_2 \cdot e_2 + (1 \quad q) \cdot e_3$$

$$y_3 = 0 \cdot x_1 + 0 \cdot e_1 \quad q^3 \cdot g_2 \cdot e_2 + (1 - q) \cdot e_3 \tag{B.14}$$

The error cancellation functions H_1, H_2, H_3 are chosen to cancel e_1 and e_2 errors. Using the following equation

$$y = H_1 \cdot y_1 - H_2 \cdot y_2 + H_3 \cdot y_3$$

and the equations developed above

$$y_1 = q\cdot x_1 + (1-q)\cdot e_1 \tag{B.10}$$

$$y_2 = 0\cdot x_1 - q^3\cdot g_1\cdot e_1 + (1-q)\cdot e_2 \tag{B.12}$$

$$y_3 = 0\cdot x_1 + 0\cdot e_1 - q^3\cdot g_2\cdot e_2 + (1-q)\cdot e_3 \tag{B.14}$$

we can create a final output equation:

$$\begin{aligned} y = {} & H_1\cdot\left[q\cdot x_1 + (1-q)\cdot e_1\right] - H_2\cdot\left[0\cdot x_1 - q^3\cdot g_1\cdot e_1 + (1-q)\cdot e_2\right] \ldots \\ & + H_3\cdot\left[0\cdot x_1 + 0\cdot e_1 - q^3\cdot g_2\cdot e_2 + (1-q)\cdot e_3\right] \end{aligned}$$

$$\begin{aligned} y = {} & H_1\cdot q\cdot x_1 + \left[H_1\cdot(1-q) + H_2\cdot q^3\cdot g_1\right]\cdot e_1 \ldots \\ & + \left[-H_2\cdot(1-q) - H_3\cdot q^3\cdot g_2\right]\cdot e_2 + H_3\cdot(1-q)\cdot e_3 \end{aligned} \tag{B.15}$$

Beginning with a desired H_3, such that we have a 2 equation - 2 unknown system, we'll try to remove the quantization errors, e_1 and e_2:

$$H_3 = \frac{(1-q)^2}{g_1\cdot g_2} \tag{B.16}$$

The coefficient for e_1 in (B.15):

$$\left[H_1\cdot(1-q) + H_2\cdot q^3\cdot g_1\right] = 0$$

The coefficient for e_2 in (B.15):

$$\left[H_2\cdot(1-q) \quad H_3\cdot q^3\cdot g_2\right] = 0$$

$$\left[-H_2\cdot(1-q) - \frac{(1-q)^2}{g_1\cdot g_2}\cdot q^3\cdot g_2\right] = 0$$

Given,

$$\left[H_1\cdot(1-q)+H_2\cdot q^3\cdot g_1\right]=0 \tag{B.17}$$

$$H_2=(-1+q)\cdot\frac{q^3}{g_1} \tag{B.18}$$

Mathcad solves (B.16), (B.17), and (B.18), yielding:

$$H_1=q^6 \tag{B.19}$$

$$H_2=q^3\cdot\frac{(q-1)}{g_1} \tag{B.20}$$

$$H_3=\frac{(1-q)^2}{g_1\cdot g_2} \tag{B.21}$$

<u>Now. Let's trv with these H_1. H_2, and H_3 Functions.</u>

Using (B.19) - (B.21) in (B.15) produces:

$$y=\left(q^6\right)\cdot\left[q\cdot x_1+(1-q)\cdot e_1\right]-q^3\cdot\frac{(q-1)}{g_1}\cdot\left[0\cdot x_1-q^3\cdot g_1\cdot e_1+(1-q)\cdot e_2\right]\ldots$$
$$+\frac{(1-q)^2}{g_1\cdot g_2}\cdot\left[0\cdot x_1+0\cdot e_1-q^3\cdot g_2\cdot e_2+(1-q)\cdot e_3\right]$$

$$y=q^7\cdot x_1+0\cdot e_1+0\cdot e_2+\frac{(1-q)^3}{\left(g_1\cdot g_2\right)}\cdot e_3 \tag{B.22}$$

Note that e_1 and e_2 really do drop out!

Now, to consider mismatches in analog and digital gains, we insert the mismatch terms

$$g_1 = \gamma_1 \cdot \left(1 + \delta_1\right) \qquad g_2 = \gamma_2 \cdot \left(1 + \delta_2\right)$$

into (B.10), (B.12), (B.14), and (B.19) - (B.21):

$$y_1 = q \cdot x_1 + (1 - q) \cdot e_1 \tag{B.23}$$

$$y_2 = 0 \cdot x_1 - q^3 \cdot \gamma_1 \cdot \left(1 + \delta_1\right) \cdot e_1 + (1 - q) \cdot e_2 \tag{B.24}$$

$$y_3 = 0 \cdot x_1 + 0 - q^3 \cdot \gamma_2 \cdot \left(1 + \delta_2\right) \cdot e_2 + (1 - q) \cdot e_3 \tag{B.25}$$

$$H_1 = q^6 \tag{B.26}$$

$$H_2 = q^3 \cdot \frac{(q - 1)}{\gamma_1} \tag{B.27}$$

$$H_3 = \frac{(1 - q)^2}{\gamma_1 \cdot \gamma_2} \tag{B.28}$$

Substituting (B.23) - (B.28) into (B.15) gives:

$$y = \left(q^6\right) \cdot \left[q \cdot x_1 + (1 - q) \cdot e_1\right] - q^3 \cdot \frac{(q - 1)}{\gamma_1} \cdot \left[0 \cdot x_1 - q^3 \cdot \gamma_1 \cdot \left(1 + \delta_1\right) \cdot e_1 + (1 - q) \cdot e_2\right] \ldots$$
$$+ \frac{(1 - q)^2}{\gamma_1 \cdot \gamma_2} \cdot \left[0 \cdot x_1 + 0 - q^3 \cdot \gamma_2 \cdot \left(1 + \delta_2\right) \cdot e_2 + (1 - q) \cdot e_3\right]$$

$$y = q^{7} \cdot x_{1} + q^{6} \cdot \delta_{1} \cdot (-1+q) \cdot e_{1} + -q^{3} \cdot (-1+q)^{2} \cdot \frac{\delta_{2}}{\gamma_{1}} \cdot e_{2} + \frac{(1-q)^{3}}{(\gamma_{1} \cdot \gamma_{2})} \cdot e_{3} \tag{B.29}$$

*** Note: The δ's create the additional appearance of quantization errors form the 1st and 2nd stages, e_1 and e_2.

Appendix C

Development of 'Nadeem' Interpolative Architecture

This appendix develops the gains used in Nadeem's Interpolative ΣΔ Modulator. The equations throughout this appendix were entered and solved in Mathcad. Due to Mathcad's habit of simplification of inverse exponentials to fractions, the following substitution is used throughout the Appendix:

$$q = z^{-1}$$

Digital Specifications and variables needed for this Mathcad Worksheet are:

$i := 0..100$ $\qquad$ $j := \sqrt{-1}$ $\qquad$ $L := 3$ $\qquad$ $f_B := 10 \cdot 10^{3}$ $\qquad$ $f_S := 1.28 \cdot 10^{6}$

$f(i) := \dfrac{\pi \cdot 10^{\frac{i}{10} - 10}}{2 \cdot \pi \cdot \dfrac{1}{f_S}}$ $\qquad$ $M := \dfrac{f_S}{2 \cdot f_B}$ $\qquad$ $M = 64$ $\qquad$ $bits := 15$

$z(i) := e^{j \cdot \left(2 \cdot \pi \cdot \frac{f(i)}{f_S}\right)}$

From Matlab, the filtering implementation gives the following elliptical zeros and poles required to compute the feedback coefficients:

$$\mathrm{zero} := \begin{pmatrix} 1 \\ 0.9987564 + j\cdot 0.049856 \\ 0.9987564 - j\cdot 0.049856 \end{pmatrix} \qquad \mathrm{pole} := \begin{pmatrix} 0.003365857 \\ 0.09444013 + j\cdot 0.62047712 \\ 0.09444013 - j\cdot 0.62047712 \end{pmatrix} \tag{C.1}$$

So, by modifying Nadeem's method, we'll develop all the coefficients from the elliptical poles and zeros starting with the z-domain transfer function of the desired Chebyshev Type-II transfer function:

$$H_D(z) = \frac{1\cdot\left(z - \mathrm{zero}_0\right)\cdot\left(z - \mathrm{zero}_1\right)\cdot\left(z - \mathrm{zero}_2\right)}{\left(z - \mathrm{pole}_0\right)\cdot\left(z - \mathrm{pole}_1\right)\cdot\left(z - \mathrm{pole}_2\right)} \tag{C.2}$$

Normally there is a constant gain associated with this transfer function. But, it is forced to 1 since there is no way to implement it in Nadeem's architecture. Substituting the poles and zeros of (C.1) into (C.2) yields:

$$H_D(z) = \frac{\left(z^3 - 2.9975128z^2 + 2.99751276727696z - .99999996727696\right)}{\left(z^3 - .192246117z^2 + .3945465385431941z - 1.32584740537294193450^{3}\right)} \tag{C.3}$$

(C.3) is the desired Chebyshev Type-II transfer function which must be matched with the NTF, $H_E(z)$, for Nadeem's modulator. For the 3rd-order interpolative system, the following STF and NTF were derived:

$$H_X(z) = \frac{K_2\cdot K_1\cdot K_0\cdot z}{\begin{array}{l} z^3 + \left(-A_2\cdot K_2 - 3 - K_2\cdot K_1\cdot B_0\right)\cdot z^2 \;... \\ + \left(-K_2\cdot K_1\cdot A_1 + K_2\cdot K_1\cdot B_0 + 2\cdot A_2\cdot K_2 + 3\right)\cdot z \;.. \\ + K_2\cdot K_1\cdot K_0\cdot A_0 + K_2\cdot K_1\cdot A_1 - A_2\cdot K_2 - 1 \end{array}} \tag{C.4}$$

$$H_E(z) = \frac{\left[z^3 + \left(-3 - K_2 \cdot K_1 \cdot B_0\right) \cdot z^2 + \left(3 + K_2 \cdot K_1 \cdot B_0\right) \cdot z - 1\right]}{\left[\begin{array}{l} z^3 + \left(-A_2 \cdot K_2 - 3 - K_2 \cdot K_1 \cdot B_0\right) \cdot z^2 \ldots \\ + \left(-K_2 \cdot K_1 \cdot A_1 + K_2 \cdot K_1 \cdot B_0 + 2 \cdot A_2 \cdot K_2 + 3\right) \cdot z \ldots \\ + -\left(K_2 \cdot K_1 \cdot K_0 \cdot A_0\right) + K_2 \cdot K_1 \cdot A_1 - A_2 \cdot K_2 - 1 \end{array}\right]} \tag{C.5}$$

There is a simple relationship between $H_D(z)$ and $H_E(z)$:

$$H_D(z) = H_E(z)$$

Using this relationship, the calculation of the feedback coefficients is accomplished by:

$$\frac{\left(\begin{array}{l} z^3 - 2.9975128z^2 \ldots \\ + 2.99751276727696z - .99999996727696 \end{array}\right)}{\left(\begin{array}{l} z^3 - .192246117z^2 \ldots \\ + .3945465385431941z \ldots \\ + -1.325847405372941934 \cdot 10^{-3} \end{array}\right)} = \frac{\left[z^3 + \left(-3 - K_2 \cdot K_1 \cdot B_0\right) \cdot z^2 + \left(3 + K_2 \cdot K_1 \cdot B_0\right) \cdot z - 1\right]}{\left[\begin{array}{l} z^3 + \left(-A_2 \cdot K_2 - 3 - K_2 \cdot K_1 \cdot B_0\right) \cdot z^2 \ldots \\ + \left(-K_2 \cdot K_1 \cdot A_1 + K_2 \cdot K_1 \cdot B_0 + 2 \cdot A_2 \cdot K_2 + 3\right) \cdot z \ldots \\ + -K_2 \cdot K_1 \cdot K_0 \cdot A_0 + K_2 \cdot K_1 \cdot A_1 - A_2 \cdot K_2 - 1 \end{array}\right]}$$

Equating like 'z' terms in the numerator and denominator creates the following 4 solvable equations:

Given

$$\left(3 + K_2 \cdot K_1 \cdot B_0\right) = 2.99751276727696 \tag{C.6}$$

$$\left(-A_2 \cdot K_2 - 3 - K_2 \cdot K_1 \cdot B_0\right) = -.19224611 \tag{C.7}$$

$$\left(-K_2 \cdot K_1 \cdot A_1 + K_2 \cdot K_1 \cdot B_0 + 2 \cdot A_2 \cdot K_2 + 3\right) = .3945465385431941 \tag{C.8}$$

$$K_2 \cdot K_1 \cdot K_0 \cdot A_0 + K_2 \cdot K_1 \cdot A_1 - A_2 \cdot K_2 - 1 = -1.325847405372941934 \cdot 10^{-3} \tag{C.9}$$

These equations can be solved simultaneously.

$$\mathrm{Find}\left(A_0, A_1, A_2, B_0\right) \rightarrow \begin{bmatrix} \dfrac{-1.2009745741378211781}{\left[K_2\cdot\left(K_1\cdot K_0\right)\right]} \\ \dfrac{-3.00756707182015412}{\left(K_2\cdot K_1\right)} \\ \dfrac{-2.80526665027696}{K_2} \\ -2.48723272304\dfrac{10^{-3}}{\left(K_2\cdot K_1\right)} \end{bmatrix}$$

$$A = \begin{bmatrix} \dfrac{-1.2009745741378211781}{\left[K_2\cdot\left(K_1\cdot K_0\right)\right]} \\ \dfrac{-3.00756707182015412}{\left(K_2\cdot K_1\right)} \\ \dfrac{-2.80526665027696}{K_2} \end{bmatrix} \tag{C.10}$$

$$B_0 = -2.48723272304\frac{10^{-3}}{\left(K_2\cdot K_1\right)} \tag{C.11}$$

Thus, these are the derived values for the feedback gains, A_0, A_1, A_2, and B_0, in terms of the integrator gains, K_1, K_2, and K_3. Assuming the following integrator gains,

$$K = \begin{bmatrix} \dfrac{39}{40} \\ \dfrac{39}{40} \\ \dfrac{40}{40} \end{bmatrix} \tag{C.12}$$

the overall gain of the modulator from input to output is

$$K_0\cdot K_1\cdot K_2 = 0.951$$

The derived coefficients to be used in the interpolative ΣΔ architecture with the prescribed integrator gains are:

$$A = \begin{pmatrix} -1.263 \\ -3.085 \\ -2.805 \end{pmatrix} \tag{C.13}$$

$$B = -0.003 \tag{C.14}$$

Let's check these coefficients in the STF and NTF.

$$H_X(i) := \frac{K_2 \cdot K_1 \cdot K_0 \cdot z(i)}{\begin{array}{l} z(i)^3 + \left(-A_2 \cdot K_2 - 3 - K_2 \cdot K_1 \cdot B_0\right) \cdot z(i)^2 \ldots \\ + \left(-K_2 \cdot K_1 \cdot A_1 + K_2 \cdot K_1 \cdot B_0 + 2 \cdot A_2 \cdot K_2 + 3\right) \cdot z(i) \,.. \\ + -K_2 \cdot K_1 \cdot K_0 \cdot A_0 + K_2 \cdot K_1 \cdot A_1 - A_2 \cdot K_2 - 1 \end{array}}$$

$$H_E(i) := \frac{\left[z(i)^3 + \left(-3 - K_2 \cdot K_1 \cdot B_0\right) \cdot z(i)^2 + \left(3 + K_2 \cdot K_1 \cdot B_0\right) \cdot z(i) - 1\right]}{\left[\begin{array}{l} z(i)^3 + \left(-A_2 \cdot K_2 - 3 - K_2 \cdot K_1 \cdot B_0\right) \cdot z(i)^2 \ldots \\ + \left(-K_2 \cdot K_1 \cdot A_1 + K_2 \cdot K_1 \cdot B_0 + 2 \cdot A_2 \cdot K_2 + 3\right) \cdot z(i) \ldots \\ + -\left(K_2 \cdot K_1 \cdot K_0 \cdot A_0\right) + K_2 \cdot K_1 \cdot A_1 - A_2 \cdot K_2 - 1 \end{array}\right]}$$

$$H_{Xdb}(i) := 20 \cdot \log\left(\left|H_X(i)\right|\right) \qquad H_{Edb}(i) := 20 \cdot \log\left(\left|H_E(i)\right|\right)$$

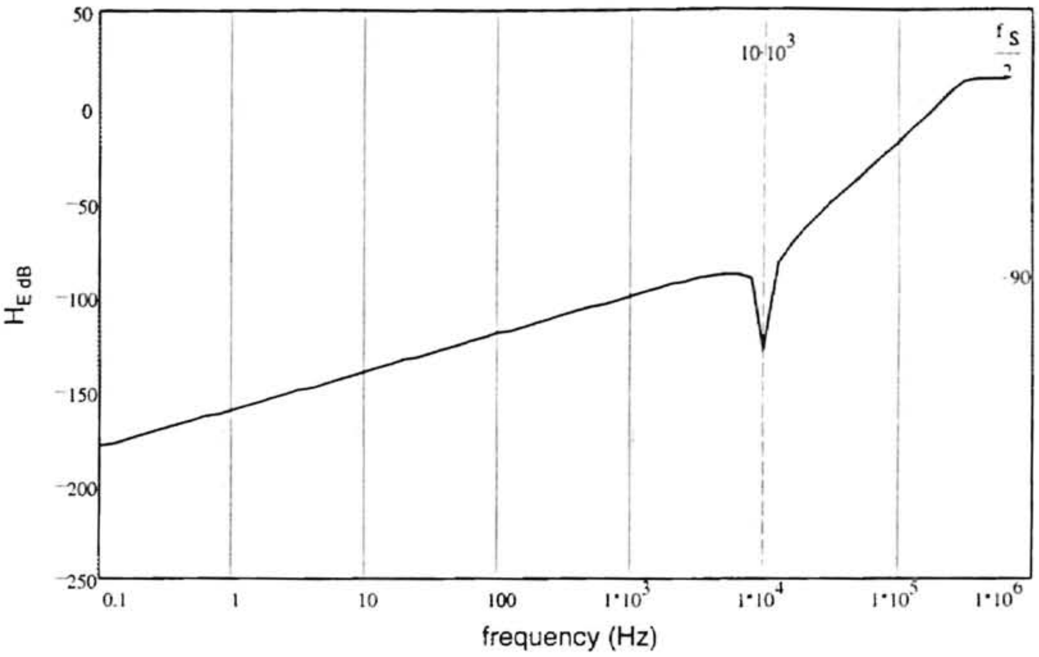


Figure C.1 Quantization Noise Transfer Function for Interpolative 3rd-Order ΣΔ Modulator.

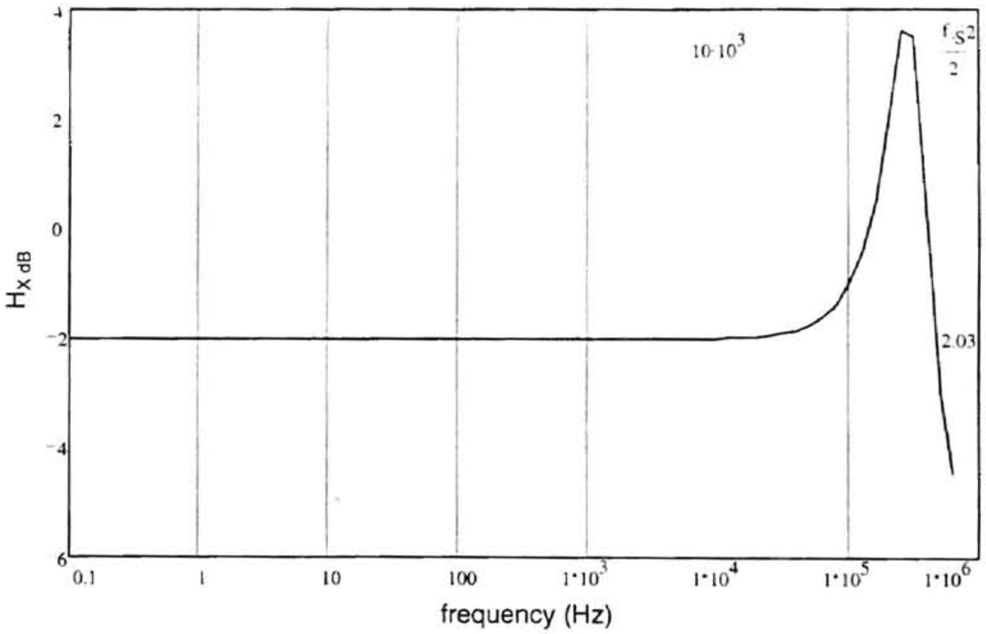


Figure C.2 Signal Transfer Function for Interpolative 3rd-Order ΣΔ Modulator.

These graphs show the effects of forcing the Chebyshev's constant gain to 1. Figure C.1 has an error gain of greater than unity for higher frequencies. In addition, there is a gain spike in the signal transfer function for higher signal frequencies.

Finally, the ΣΔ Toolbox needs a power series approximation for the STF(z) in order to integrate the NSTerm for an average inband SNR. This is done using the STF(z) found in (C.4). Substituting (C.12), (C.13), and (C.14) into (C.4), the STF(q), where $q = z^{-1}$, becomes:

$$H_x(q) = \frac{\left(q^2\right)\cdot\left(K_2\cdot K_1\cdot K_0\right)}{\begin{array}{l}\left(-1 + 1.201\cdot K_2\cdot K_1\cdot K_0 - 3.008K_2\cdot K_1 + 2.805K_2\right)\cdot q^3 + \left(3.006K_2\cdot K_1 - 5.61\cdot K_2 + 3\right)\cdot q^2 \ldots \\ + \left(-3 + 2.0\cdot 10^{-3}\cdot K_2\cdot K_1 + 2.805K_2\right)\cdot q + 1\end{array}} \tag{C.15}$$

A 9th-order power series approximation of (C.15) is:

$$\begin{array}{ll} H_{x1}(i) & .950625z(i)^{-2} + .18351815625z(i)^{-3} - .26957985118593752(i)^{-4} - .1241384723448827343z(i)^{-5} \ldots \\ & + 5.99787460961057787590^{-2}\cdot z(i)^{-6} + 5.51560542046003683810^{-2}\cdot z(i)^{-7} \ldots \\ & + 6.87070206859835243880^{-3}\cdot z(i)^{-8} \end{array} \tag{C.16}$$

An 11th-order power series approximation of (C.15) is:

$$\begin{array}{ll} H_{x2}(i) & .950625z(i)^{-2} + .18351815625z(i)^{-3} - .26957985118593752(i)^{-4} - .1241384723448827343z(i)^{-5} \ldots \\ & + 5.99787460961057787590^{-2}\cdot z(i)^{-6} + 5.51560542046003683810^{-2}\cdot z(i)^{-7} \ldots \\ & + 6.87070206859835243880^{-3}\cdot z(i)^{-8} \quad 1.98569510833411205250^{-2}\cdot z(i)^{-9} \ldots \\ & + 2.39562327390704493320^{-3}\cdot z(i)^{-10} \end{array} \tag{C.17}$$

Finally, a comparison between the 9th-order series and the 11th-order series can be made:

$$H_{X1db}(i) := 20\cdot\log\left(\left|H_{x1}(i)\right|\right) \qquad H_{X2db}(i) := 20\cdot\log\left(\left|H_{x2}(i)\right|\right)$$

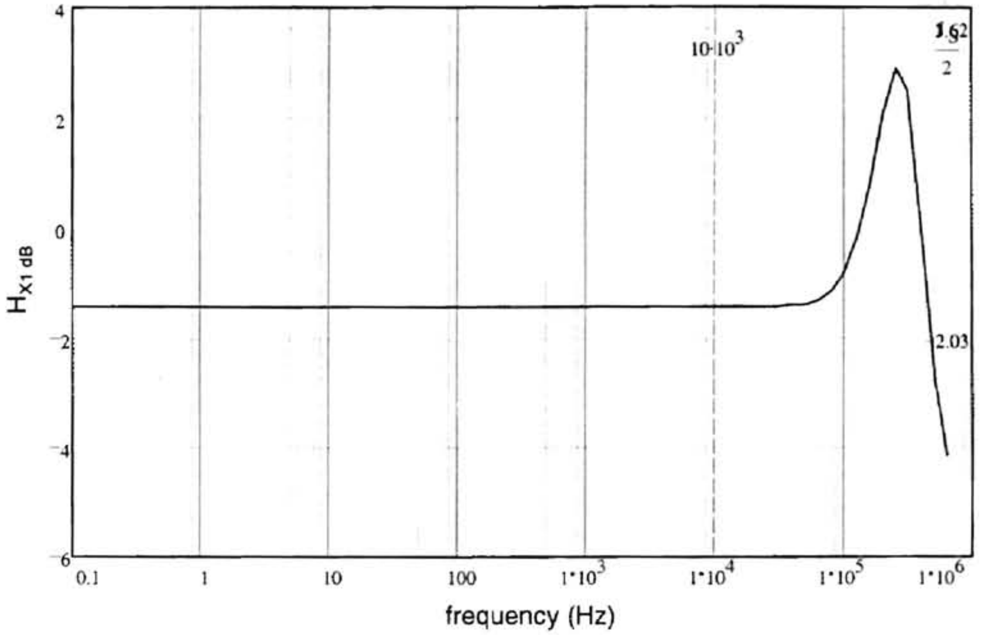


Figure C.3 9th-Order Approximation for Signal Transfer Function of Interpolative 3rd-Order ΣΔ Modulator.

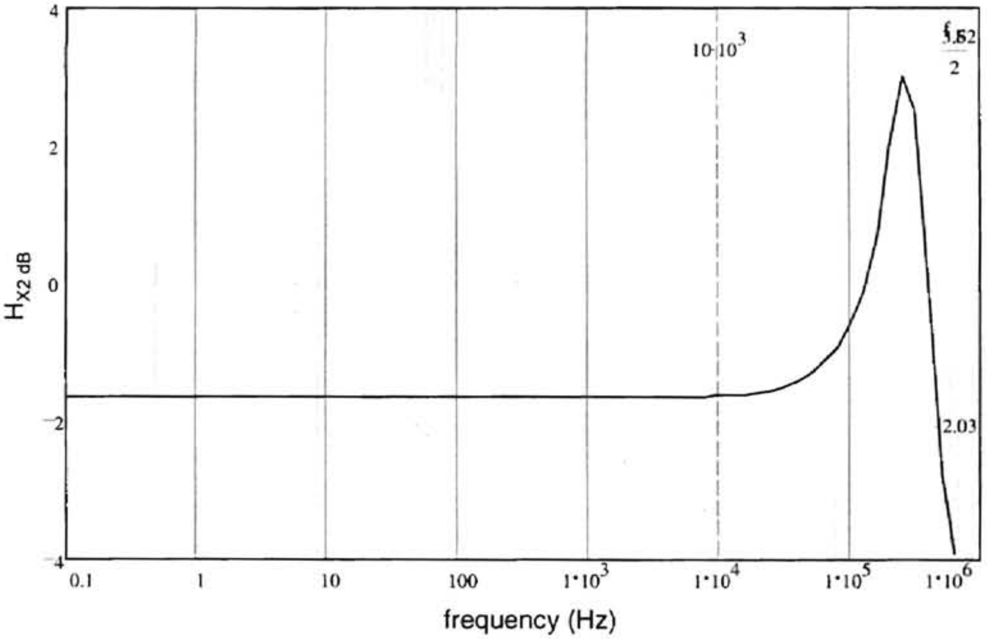


Figure C.4 11th-Order Approximation for Signal Transfer Function of Interpolative 3rd-Order ΣΔ Modulator.

The 11th-order power series of (C.17) provides a better approximation of the STF for the modulator. It is used in the architecture file for the ΣΔ toolbox.

VITA

Anurag K. Tyagi

Candidate for the Degree of

Master of Science

Thesis: A BEHAVIORAL SIMULATOR FOR SIGMA-DELTA ANALOG TO DIGITAL CONVERTERS.

Major Field: Electrical Engineering.

Biographical Information:

Personal Data: Born in Oakland, California on February 23, 1969.

Education: Graduated from Stillwater High School, Stillwater, Oklahoma in May 1987; completed major course work at the University of Texas at Austin, Texas; graduated with a Bachelor of Science degree in Electrical and Computer Engineering from Oklahoma State University, Stillwater, Oklahoma in August 1994. Completed the requirements for the Master of Science degree at Oklahoma State University in December 1996.

Experience: Employed by Lockheed-Martin (1996-present) working on Wireless Communications Systems and Hardware Engineering in Manassas, Virginia; employed by Oklahoma State University, Department of Electrical and Computer Engineering as a Graduate Teaching Assistant and as a Graduate Research Assistant (1994, 1994-1996, respectively), including research work with Naval Research and Development, San Diego, California (summer 1995); employed by HydroLogic, Inc. (1992-1995), Stillwater, Oklahoma.

Professional Memberships: IEEE: Signal Processing Society, Circuits and Systems Society; Oklahoma Society at Professional Engineers; National Society of Professional Engineers.