INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

Characterization and Digital Correction of Multi-bit Delta-sigma Modulators

by

Xiang (Shannon) Wang

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

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

© Xiang (Shannon) Wang 1999

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-48476-9

Canada

The undersigned hereby recommended to the Faculty of Graduate Studies
and Research acceptance of this thesis

**“Characterization and Digital Correction
of Multi-bit Delta-sigma Modulators”**

submitted by

Xiang (Shannon) Wang (B.Eng.)

in partial fulfilment of the requirements for the degree of Master of Engineering

Thesis Co-Supervisor

Thesis Co-Supervisor

Chair, Department of Electronics

Carleton University

December 1999

Abstract

Delta-sigma modulators are mostly used in high resolution applications such as digital audio, digital telephony, and instrumentation.

Multi-bit delta-sigma modulators exhibit a number of attractive features such as providing higher resolution than that of single-bit modulators theoretically and improving stability characteristics. However, stringent accuracy requirements are placed on the digital-to-analog converters used in the feedback loop since the delta-sigma modulators are particularly sensitive to static nonlinearity.

Many nonlinearity compensation methods in multi-bit delta-sigma modulators have been published such as the element trimming methods, the dynamic element matching method, the dual-quantizer ADC architecture and many digital correction methods. None of the methods has fully solved the nonlinearity problem yet.

A novel system identification approach is proposed in this thesis. This method can help characterize a multi-bit sigma-delta modulator. Simulation and hardware testing has been done on the system identification method to prove its practicability.

Acknowledgments

In writing this thesis, I had the privilege of working with my two supervisors, Dr. Ralph Mason and Dr. Calvin Plett. I wish to thank Ralph Mason for his continuous guidance, support and his wisdom; Calvin Plett for his insight and encouragements. I am heavily indebted to them. I also want to thank Dr. Martin Snelgrove for his brilliant ideas and unconventional thoughts.

In addition to my supervisors, I also wish to thank Phil Lauzon, Ash, James Cherry, Yu Li, Nagui, and Warren for their helps. On a broader scope, I appreciate the friendships and interactions made with the electronic group graduate students.

Finally, I wish to thank my family and all my friends for their love and support. which made this seemingly endless task into an enjoyable endeavour.

Table of Contents

Abstract	II
Acknowledgments	III
Table of Contents	IV
List of Figures	VI
Chapter 1. Introduction	1
1.1 Motivation	1
1.2 Organization of the Thesis	3
Chapter 2. Linearity of Multi-bit Delta-sigma Modulators	5
2.1 Basics of Delta-sigma Modulators	5
2.2 Single-bit versus Multi-bit Delta-sigma Modulators	8
2.3 Nonlinearity Compensation in Multi-bit Delta-sigma Modulators.	9
2.3.1 Element Trimming	11
2.3.2 Dynamic Element Matching	11
2.3.3 Dual-quantizer ADC Architectures	14
2.3.4 Digital Corrections	15
2.4 Summary	18
Chapter 3. IIR System Identification	19
3.1 System Identification in Adaptive Signal Processing	20
3.2 Infinite Impulse Response System Identification	22
3.2.1 Adaptive FIR Approach	26
3.2.2 Output error approach	27
3.2.3 Equation Error Approach	28
3.2.4 Improved Versions of Equation Error Family	31
3.2.4.1 A New Adaptive IIR Filter	32
3.2.4.2 Sign-sign Equation Error Identifier	35
3.2.4.3 Parallel-Form Realizations	35

3.2.4.4	BRLE or CRA	37
3.2.4.5	Bias-Removal in Equation Error Adaptive IIR Filters	38
3.3	Summary	39
 Chapter 4. Characterization and Digital Correction in		
Delta-sigma Modulators		40
4.1	Delta-sigma Loop Identification	41
4.1.1	Selection of the Adaptive Method	41
4.1.2	Selection of Dither Location	42
4.1.3	Simulation Examples	46
4.2	Digital Correction of a Crystal Semiconductor Circuit	54
4.2.1	Theory Analysis	54
4.2.2	Simulation Examples	60
4.3	Summary	66
 Chapter 5. Hardware Testing		68
5.1	System Identification	68
5.1.1	Simulation Result	68
5.1.2	Circuit Structure and Tested Results	72
5.2	Summary	78
 Chapter 6. Conclusions and Possible Future Work		79
6.1	Contribution and Conclusions	79
6.2	Future Work	80
 References		82
 Appendix A: System Identification in C		85
 Appendix B: Digital Correction in C		94

List of Figures

Figure 2.1:General block diagram of delta-sigma modulator	6.
Figure 2.2:Mismatch effect comparison.	10.
Figure 2.3:General block diagram of internal DAC for dynamic element matching.	12.
Figure 2.4:General scheme of digital correction in delta-sigma modulators.	16.
Figure 2.5:Digital correction of N-bit delta-sigma converter.	17.
Figure 3.1:Mathematical model of system identification.	21.
Figure 3.2:Theoretical model of a delta-sigma modulator.	23.
Figure 3.3:Block diagram of the adaptive IIR filter.	25.
Figure 3.4:Block diagram of an adaptive FIR filter.	26.
Figure 3.5:Simple adaptive IIR output error scheme.	27.
Figure 3.6:Equation error method of IIR adaptive filter.	29.
Figure 3.7:Equation error versus Output Error.	31.
Figure 3.8:Theoretical mode of the new adaptive IIR filter.	33.
Figure 3.9:Adaptation mode of the new adaptive IIR filter.	34.
Figure 3.10:Parallel form adaptive filter.	36.
Figure 4.1:Possible dither adding points on an delta-sigma loop.	42.
Figure 4.2:An 2nd-order low-pass sigma-delta modulator.	46.
Figure 4.3:Simulation results for the second order low-pass sigma-delta modulator.	48.
Figure 4.4:Simulation model of a fourth-order band-pass sigma-delta modulator.	49.
Figure 4.5:Pseudo noise generator.	50.
Figure 4.6:Output comparison for the 4th-order bandpass delta-sigma modulator.	51.
Figure 4.7:Simulation results for the 4th-order bandpass delta-sigma modulator.	53.
Figure 4.8:Illustration of DAC output versus modulator output.	56.
Figure 4.9:Block diagram of correcting and decimator chip.	57.
Figure 4.10:Theoretical model for the system.	57.
Figure 4.11:The general block diagram of the calibration system.	59.
Figure 4.12:A 4th order band-pass delta-sigma modulator with a 3-level quantizer.	61.
Figure 4.13:General algorithm structure for band-pass delta-sigma modulator nonlinearity cancellation	62.
Figure 4.14:Simulation example for nonlinearity cancellation.	63.
Figure 4.15:A 6th order band-pass delta-sigma modulator with a 3-level quantizer.	64.
Figure 4.16: Simulation example: (a) Convergence of W. (b) Corrected vs. uncorrected	

output.	65.
Figure 5.1:Block diagram of a second-order band-pass delta-sigma modulator.	68.
Figure 5.2:Simulation result of coefficients convergence.	70.
Figure 5.3:Simulation results: (a) With a 5-bit quantizer. (b) With a 2-bit quantizer.	71.
Figure 5.4:Tolerance to number of quantization bits ratio.	72.
Figure 5.5:Block diagram of the testing circuit.	73.
Figure 5.6:Top level schematics of the chip.	74.
Figure 5.7:Layout view of the chip in Cadence.	74.
Figure 5.8:Block diagram of simplified testing circuit.	75.
Figure 5.9:Altera board configuration.	76.
Figure 5.10:Experimental result of the delta-sigma chip.	77.

Chapter 1. Introduction

1.1 Motivation

Data converters provide the link between the analog world of transducers and the digital world of signal processing, computing and other digital data collection or data processing systems. There are numerous types of converters. Speed, resolution, cost and power consumption are generally the major concerns.

Oversampling methods have recently become popular. Noise-shaping techniques are often combined with oversampling and constitute the well-known $\Delta\Sigma$ family of converters. The chief feature of a $\Delta\Sigma$ converter is that speed can be traded for resolution, and quantization noise can be shaped out of the band of interest. Although the concept of sigma-delta modulation appeared almost six decades ago, [Cut54], it was not formally referred to as delta sigma modulation until 1962 [Ino62] and it was not until the mid

1980s, when the semiconductor technology reached very large scale integrated proportions, that the digital filtering required in $\Delta\Sigma$ converters became cost effective. Since their sampling rate usually needs to be several orders of magnitude higher than the Nyquist rate, $\Delta\Sigma$ modulators are mostly used in digital audio, digital telephony, and instrumentation. In high speed applications, flash, half-flash, folding, pipelined and sub-ranging converters have been developed. [Car89] . As VLSI technology is better suited to providing fast digital circuits than providing precise analog circuits, future applications for $\Delta\Sigma$ converters will include video and radar system. [Car89] .

Despite the promising future of delta-sigma modulators, a typical $\Delta\Sigma$ ADC (analog-to-digital Converter) is still limited to using a single-bit comparator and a single-bit DAC (digital to analog converter) in a feedback configuration. Multi-bit delta-sigma ADCs exhibit a number of attractive features, including significantly lower quantization noise for a given oversampling ratio, as well as improved stability characteristics. However, the principle drawback of the multi-bit delta-sigma ADC is the stringent accuracy requirements placed on the feedback DACs. The multi-bit delta-sigma ADC is particularly sensitive to the static nonlinearity of the coarse DAC in the feedback loop, since the resulting error is directly added to the input without being noise shaped.

Many nonlinearity compensation methods in delta-sigma modulators have been published: the element trimming methods, the dynamic element matching method, the dual-quantizer ADC architectures and many digital correction methods. Each method is

unique but has its own limitation. None of the methods has fully solved the nonlinearity problem yet.

A novel system identification approach is proposed in this thesis. This method can help characterize a multi-bit sigma-delta modulator. The practical side of this method is that if a delta-sigma modulator is fabricated and does not work as well as predicted, then by applying this approach the gain of each stage can be calculated and compared to its original design to determine which parts inside the delta-sigma loop are not operating correctly. Examples of such undesirable operation might be a leaky switch or that the op-amp gain is too low. Simulation and hardware testing has been done on the system identification method to prove its practicability.

1.2 Organization of the Thesis

This thesis first studies the nonlinearity problems involved in multi-bit delta-sigma modulators. A comparison of existing techniques for solving the problem is provided. A novel system identification approach is presented. This approach can help characterize a multi-bit sigma-delta modulator. Then, a detailed simulation of a Crystal Semiconductor patent [Tho96] on digitally correcting the multi-bit nonlinearity error is given.

The thesis is divided into five chapters. Chapter 2 introduces the basic concepts of delta-sigma modulators, including low-pass versus band-pass and single-bit versus multi-

bit implementations. A comparison of existing technique to solve the multi-bit non-linear problem is given. Chapter 3 introduces the basics of adaptive signal processing and system identification. IIR adaptive filters are introduced in details. Chapter 4 first presents a novel idea of using system identification on multi-bit delta-sigma modulators. The original purpose of this method was to solve nonlinearity in a 1.5-bit delta-sigma modulator by characterizing the modulator loop. Simulations show some limitations which make the identification process unsuitable for modulators with less than three bits. However, the method is still valid and useful for characterizing the delta-sigma loop, for example, if something inside the loop malfunctions, we can determine where the problem is. Simulation results are given as well as discussions. In the second part of the chapter, a detailed implementation example and analysis are given for the Crystal Semiconductor patent, which is an effective way of solving multi-bit delta-sigma nonlinearity. Chapter 5 deals with the hardware testing. The test set-up for the system identification method is described and the testing results are presented. Chapter 6 discusses the contributions of this thesis and possible future enhancements.

Chapter 2. Linearity of Multi-bit Delta-sigma Modulators

2.1 Basics of Delta-sigma Modulators

Delta-sigma modulators are the combination of oversampling and noise shaping. Oversampling refers to sampling above the Nyquist rate. Oversampling methods can provide very high resolutions even when relatively inaccurate analog components are used. The ever-increasing speed capabilities of new VLSI technology will allow larger oversampling ratios and possibly higher resolution, but this will soon be limited by circuit noise[Nor97]. Noise shaping refers to the process of using feedback to control the spectrum of quantization noise. The quantization errors in a noise-shaping system can be removed from the signal band of interest. The suppressed quantization errors appear enlarged as out-of-band noise in the system. Using a simple filter, out of band errors are

removed, and increased dynamic range is obtained. Noise -shaping can be very useful when speed can be traded off for accuracy.

A general diagram of a delta-sigma modulator is shown in Figure 2.1. Oversampling the continuous-time signal before quantization reduces the quantization noise density by the factor of the oversampling ratio. The inband quantization noise is reduced by 3dB for every doubling of the oversampling ratio. [Car89]. The linear system in the modulator loop shapes the quantization noise by placing nulls in the quantization noise spectrum at the band of interest. This in turn enhances the output bit stream signal-to-quantization noise ratio.

$$Y(z) = \frac{H(z)}{1 + H(z)} \times X(z) + \frac{1}{1 + H(z)} \times Q(z) \quad Q(z): \text{Quantization Noise}$$

Figure 2.1: General block diagram of delta-sigma modulator

There has been keen interest in delta-sigma modulators for low-speed high-resolution and high linearity applications. They have had limited success in high-speed high-bandwidth applications since high performance delta-sigma modulators can

typically only be obtained by using a high oversampling ratio. For high bandwidth applications this requires a very high sampling rate which severely limits real world implementations.

Great interest has developed in band-pass delta-sigma converters which offer efficient signal processing for digital wireless device. As previously mentioned, the modulator loop filter puts nulls in the quantization noise across the band of interest. In a low pass delta-sigma modulator the zeros of the quantization noise are near DC. One can extend this principle to bandpass by moving nulls to some non-DC frequencies which produces a band-reject noise-shaping property[Sho96].

In a linear model of a delta sigma modulator, the quantizer noise is often replaced by additive white noise $Q(k)$ [Sne95]. From this linear model we can obtain the noise transfer function $NTF(z)$ and signal transfer function $STF(z)$ as follows:

$$NTF(z) = \frac{Y(z)}{Q(z)} = \frac{1}{1 + H(z)} \quad (2.1)$$

$$STF(z) = \frac{Y(z)}{X(z)} = \frac{H(z)}{1 + H(z)} \quad (2.2)$$

A low-pass modulator can be converted into a band-pass modulator using the transformation $Z^{-1} \rightarrow -Z^{-2}$ at $f_{sample} = 4 \cdot f_{Centre}$ [Opp75]. It is apparent that the order of the obtained bandpass modulator is twice as high as that of the original lowpass and with the same SNR for a given bandwidth. The transformation does not

change the dynamics of the low-pass prototype, so a stable low-pass modulator produces a stable band-pass one.

2.2 Single-bit versus Multi-bit Delta-sigma Modulators

When the maximum sampling frequency is limited, there are two ways to enhance the performance of a delta-sigma modulator. One is to go to a higher order noise-shaping filter. The by-product is potential stability problems[Nor97]. Stability for systems higher than second-order is still an unanswered mystery of delta-sigma theory so far. Another approach is to use a multibit internal quantizer.

One-bit delta-sigma modulators have been popular for use in integrated circuits (IC) due to the fact that they employ a one-bit internal DAC which is inherently linear and does not require precision component matching. The resolution of a one-bit delta-sigma modulator is limited by the oversampling ratio. Although the achievable resolution does improve with increasing loop filter order, these improvements diminish rapidly due to instability. In addition, higher order delta-sigma modulators produce higher out-of-band quantization noise and the design of analog output filters to remove this out-of-band noise can be quite difficult [Nor97].

One solution to the above problems is to use a multi-bit quantizer in the oversampled converter loop. This adds stability to the high-order loops and makes the

design of such loops easier. Also the decreased quantization noise power increases converter resolution by more than 3 dB per additional quantizer bit. Equivalently, the multi-bit noise-shaping coder can achieve resolution comparable to that of a single-bit modulator at a lower sample rate. This can be a significant advantage in applications requiring high bandwidth. Another advantage of the lower clock rate is that it can decrease power consumption in the digital circuitry. The decrease in quantization noise power that improves resolution also relaxes the requirements on the output filter that must remove the out-of-band quantization noise power.

A critical side effect of the multi-level quantizer and feedback DAC is the loss of intrinsic linearity enjoyed by the two-level (one bit) system. This nonlinearity, if not corrected, results in intermodulation of the high-frequency quantization noise components near $F_s/2$ which are modulated back into the passband of interest.

Figure 2.2 shows a comparison between a third-order low-pass delta-sigma modulator with and without analog component mismatch. The power spectrum density is plotted against normalized frequency. The peak tone is the input signal we feed in and the noise floor shows a typical noise-shaping effect of a delta-sigma modulator. We can see that even small mismatch on the order of one percent render the multibit approach unusable. A secondary disadvantage of multi-bit modulators is that more analog circuitry, which is generally more difficult to design, is required.

Figure 2.2: Mismatch effect comparison.

(a) Without mismatch (b) With 1% mismatch.

2.3 Nonlinearity Compensation in Multi-bit Delta-sigma Modulators

Using a multibit internal quantizer necessitates the inclusion of a multibit DAC in the feedback loop. Quantization noise is shaped while DAC mismatch noise is not. This is because the quantizer is in the upper half of the loop while the DAC is in the lower half of the loop which brings errors directly into the input. (See Equations 2.1 and 2.2). The linearity of the DAC limits the linearity of the complete ADC, and hence its design can be a difficult task. Extensive research has been carried out in this area, such as element

trimming, [Car89], dynamic element matching and intervening multibit ADCs with a single-bit DAC. Also, various calibration methods and digital correction technique have been presented as well[Nor97].

2.3.1 Element Trimming

The most straightforward approach to improve the accuracy of the internal DAC is to improve the matching of the individual elements. Approaches of this type can generally be divided into two groups: [A] one-time trimming and [B] repeated trimming. [Nor97]

One-time trimming is part of the manufacturing process and no extra on-chip circuitry is required. There are two drawbacks. First, variations in temperature, voltage, and component aging, cannot be compensated for. Second, laser trimming, which is the most common one-time trimming method, is very expensive. Repeated trimming has better tracking ability, it does however requires some on-chip measurement hardware to determine how to trim the elements.

Another idea is to use one element repeatedly instead of trimming multiple elements. Although it is much simpler and more accurate, it is restricted to low frequency and low resolution applications. If a capacitor is used as the matching element, charge injection can also affect the accuracy.

2.3.2 Dynamic Element Matching

As element trimming could be very expensive, dynamic element matching was presented. If the static error of mismatch between the components can be converted from a dc error into a wide-band noise signals, it can be fully or partially removed by a subsequent filter in the delta-sigma modulator. Several dynamic element matching approaches have been presented, [Nor97], each has its own method for choosing different elements at different times. The general idea is shown in Figure 2.3, the control logic part

Figure 2.3: General block diagram of internal DAC for dynamic element matching.

is different for different approaches.

In [Car89], L.R. Carly presented the idea of randomly using each element. With an ideal randomizer, and assuming the mismatch between elements is independent of the

element's position, there will be no correlation between the mismatch error at a given time and at any other time. Therefore, the mismatch error has been converted into a white noise. With a high oversampling ratio, nearly all of the error power is out of band and hence can be filtered out. However, the randomizer will certainly need more circuitry, and in reality, physically adjacent elements are normally more matched than distant ones which results in larger noise power than predicted from theory. Another disadvantage is the output filter, which will not help remove the in-band noise power and therefore in-band SNR degradation, is unavoidable.

Y. Sakina [Sak93] mentioned an improved method as dynamic element rotation, which modulates the nonlinearity error around sub-harmonics of the sampling clock frequency by making the mismatch noise a periodic signal. A shift register is used for this purpose. Although this method still has some noise power and harmonics inside the passband, the degradation of passband SNR is much smaller compared to the randomization method.

Leung proposed another form of dynamic element matching called individual level averaging [Leu92], [Che95]. The fundamental idea is to guarantee that each of the elements is used with equal probability for each digital input code. To satisfy this, a single digital register is needed. Experiments also show that addition style of level averaging is better than the rotation style.

Recently, Rex T. Baird and Terri S. Fiez published a data weighted averaging method [Bai95]. Each element is used in a rotation manner and it is claimed to be more

efficient than individual level averaging with less circuitry. Distortion aliasing related to the technique is reduced by adding a random dither. The dither causes minor degradation to the data converters performance. Unfortunately, this technique does not work as well when the number of bits increases since each bit increase doubles the elements to cycle through before errors average to zero.

It has also been realized that it is possible to apply the noise-shaping principle to the errors caused by element mismatch. [Sch95] . A few multi-bit noise-shaping DACs have been proposed that use digital signal processing techniques to cause the DAC noise to be spectrally shaped. In [Gal97], the practicality of the noise-shaping DAC approach is extended by presenting a general noise-shaping DAC architecture. A rigorous theoretical explanation of the technique is also provided. High performance can be achieved even with moderately well matched components and it could be easily generalized to any number of bits. The first and second order configuration is hard-ware efficient, however additional research is needed for higher-order noise-shaping.

2.3.3 Dual-quantizer ADC Architectures

Another alternative to the design of a delta-sigma modulator with multibit internal quantizer is to use two quantizers. One is a single-bit quantizer with a single-bit DAC in the feedback path between the input of the modulator and the output. The other is a multi-bit quantizer. The principle is to use the inherent linearity of a single-bit DAC to its full advantage. The single-bit DAC plays a key role in determining the linearity of the modulator. The multi-bit quantizer is used to convert and cancel the large quantization

error generated by the single-bit quantizer. By carefully choosing the transfer functions, this cancellation will then reduce the overall quantization error of the modulator, theoretically, to that of one with a linear multibit internal quantizer.

There are several structures in this category. In [Che95], A. Hairapetian proposed a dual-feedback single-path structure used on a third-order analog-to-digital converter. The first two integrators were fed back through a one-bit quantizer and DAC loop while the third integrator was fed back through a multi-bit quantizer and DAC loop. B.P. Brandt suggested another useful dual-quantizer structure in [Bra91]. It could be used commonly in cascaded ADCs. The single-bit quantizer and DAC are used in the first stage while the multi-bit ones are used in the second stage. The Leslie-Singh architecture in [Les90] is also an alternative. The multi-bit quantizer is used in the forward path but only the most significant bit is fed back to the single-bit internal DAC.

In practice, theoretical constraints on loop functions are not always satisfied since it is an analog component loop. The finite gains of the input op-amps add another source of noise. This noise can leak to the output. This usually limits the achievable accuracy in a high resolution multi-bit analog-to-digital converter.

2.3.4 Digital Corrections

Unlike the dynamic element matching techniques, which try to shift most mismatch noise out-of-band and remove it through output digital filtering, digital correction can be considered a totally different strategy. The idea is to convert the noise

due to DAC error into a digital form and then cancel it in the digital domain. A good digital correction system should provide precise calibration with the least complexity and minimal cost. An on-line calibration is preferred since it can bring continuous adaptation.

The General block diagram of digital correction in delta-sigma modulators is shown in Figure 2.4 Here the digital correction part is outside the loop. An alternative way of employing digital correction into the delta-sigma ADC is to place it in the feedback path right before the multi-bit DAC. The advantage of doing the correction as shown in Figure 2.4 is that it could be implemented off-chip without tampering with the delta-sigma loop operation.

Figure 2.4: General scheme of digital correction in delta-sigma modulators.

A few error correction and self-calibration techniques have been published to date. L.E. Larson proposed a technique that incorporates the digital correction on the feedback path as shown in Figure 2.5. [Lar88]. No analog components are required to have precision equal to that of the final conversion. The precision of the DAC, and hence the overall precision, is achieved by the feedback action.

Figure 2.5: Digital correction of N-bit delta-sigma converter.

In fact the noise transfer function of the DAC can be approximately deduced as:

$$VTF(z)_{DAC} = \frac{Y(z)}{X(z)} = \frac{H_1(z)}{1 + H_1(z)} \cdot \left[\frac{1}{1 + H_2(z)} \right] \quad (2.3)$$

Compared to adding DAC noise directly into the input, this approach allows noise to go through the signal flow path. This structure leads to a greatly reduced noise gain.

Generally speaking, digital correction methods present effective ways of solving nonlinearity. Calibration can be implemented off-chip to avoid tampering with the delta-sigma loop operation, therefore, no re-configuration of analog circuitry is required. However, most known calibrating techniques for achieving high linearity require some kind of highly linear calibration reference. The precision of the calibrated DAC is limited by the precision of the linearity reference, and it is difficult to obtain a good linearity reference. In [Tho94], a digital correction technique was presented, which was later

patented by the Crystal Semiconductor Inc. This approach eliminates the need for a calibration reference. In Chapter 4, this technique will be addressed in detail.

2.4 Summary

In this chapter, basic principles of Delta-sigma modulators were shown. Specifically, over-sampling and noise shaping allow us to get increased resolution from an A/D converter. We also looked at the ways to increase the resolution of a delta-sigma modulator. Emphasis was placed upon the single-bit to multi-bit change. We saw that multi-bit quantizers and DACs have a critical drawback in that they do not have the linearity enjoyed by a one-bit structure. This can greatly degrade their performance.

An introduction of existing compensation methods was provided. Each method is unique but has its own limitation. None of the methods has fully solved the nonlinearity problem yet. This is a stimulus for the further efforts made in this thesis.

Chapter 3. IIR System Identification

Introduction:

In Chapter two we looked at several ways to solve nonlinearity problems brought from the feedback of digital-to-analog converters. Digital correction is a clean and effective way of correcting it. This throws a light on taking advantage of adaptive signal processing in delta-sigma modulators. For example, if we can characterize the delta-sigma loop, it may be very useful to figure out the actual design gain of each op-amp, nonlinearity coefficients, etc.

3.1 System Identification in Adaptive Signal Processing

Adaptive signal processing methods form a common tool in the “box of tricks”. Some applications are already briefly mentioned in last chapter. Although optimum signal processing is desirable in many applications, it generally requires exact knowledge of the system model and signal and noise statistics. However, this information may not be available, or maybe non-stationary. Adaptive filters can estimate and track a changing environment so that nearly optimum performance can be attained. Due to their robustness, they are of great importance to telecommunications, industrial monitoring, navigation, air traffic control, and numerous other area. Their main application includes system identification, inverse modelling, system prediction and interference cancelling [Wid85] [Hay96].

The following discussion will concentrate on system identification only. Modelling a single-input, single-output dynamic system or “plant” is illustrated in Figure 3.1. Both the unknown system and the adaptive filter are driven by the same input. The adaptive filter adjusts itself with the goal of causing its output to match that of the unknown system, generally to cause its output to be a best least-squares fit to that of the unknown system. If enough flexibility resides in the adaptive system, (i.e. It is sufficiently adjustable and contains enough “degree of freedom”), a close fit or perhaps a perfect fit is possible. Upon convergence, the structure and parameter values of the adaptive system may or may not resemble those of the unknown system, but the input-output response

relationships will match. In this sense, the adaptive system becomes a model of the unknown system. If the input is wide-band and if the structure of the adaptive system is such that when its adjustable parameters are suitably chosen, an exact match is possible, adapting to minimize the mean-square error will result in an exact match in every detail.

Figure 3.1: Mathematical model of system identification.

$$Y(n) = W^T \cdot X(n) \qquad W_{Optimum} = R^{-1} \cdot P \qquad (3.1)$$

Where:

W stands for the vector of coefficients

R stands for auto-correlation of X(n)

P stands for cross correlation between d(n) and X(n)

A wide variety of recursive algorithms have been developed in the literature for the operation of linear adaptive filters. In the final analysis, the choice of one algorithm over another is determined by one or more of the factors such as rate of convergence, misadjustment, tracking ability, robustness, and complexity.

The least-mean-square algorithm, or LMS algorithm, is famous for its simplicity and ease of computation, and it does not require off-line gradient estimations or repetitions of data. If the adaptive system is a linear combiner, and if the input vector and the desired response are available at each iteration, the LMS algorithm is generally the best choice for many different applications of adaptive signal processing. Most of the algorithms which will be addressed in the following are updated in LMS.

3.2 Infinite Impulse Response System Identification

If we take a closer look at a delta-sigma modulator as shown in Figure 3.2, it is easy to find out that both the noise transfer function and the signal transfer function are recursive due to the loop. To characterize the delta-sigma loop we need to look into the issue of Infinite Impulse Response (IIR) system identification.

$$Y = \frac{k_0 \cdot X(z) \cdot H(z) + Q}{1 + k_0 \cdot k_1 \cdot H(z)} = \frac{k_0 \cdot X(z) \cdot H(z)}{1 + k_0 \cdot k_1 \cdot H(z)} + Noise \quad (3.2)$$

An IIR system can be realized adaptively either by a Finite Impulse Response (FIR) digital filter or an IIR filter. The branch of FIR adaptive filters, in which only the

Figure 3.2: Theoretical model of a delta-sigma modulator.

zeros of the filter are adapted, are fairly well developed. The design theory is well mastered, allowing reliable application development. However, practical experience with adaptive FIR filters has also revealed performance limitations that might be overcome with alternate adaptive filtering formulations. Adaptive FIR filters adjust finite duration impulse responses and, by no surprise, have inherent limitations where infinite duration impulse response would seem more appropriate. These limitations have become particularly apparent upon modelling acoustic impulse responses which arise in echo cancellers, or physical and industrial processes that are the domain of control engineers. With increased experience in matching filters to applications, the need for recursive or IIR filters is increasingly warranted, particularly in applications where a FIR filter would require an unreasonably large number of coefficients to obtain the same satisfactory performance compared to its IIR counterpart.

This basic recognition has spawned considerable, though irregular, activity over the years in adaptive IIR filtering. Such adaptive filters adjust rational transfer functions, in contrast to their FIR counterparts which adjust polynomial transfer functions. Most scientists accept that rational functions have more versatile and powerful modelling capabilities than polynomial functions, they can usually match physical systems well [Wid85].

Adaptive IIR filtering basically falls into two classes: equation error methods and output error methods. A general direct form adaptive IIR filter is shown in Figure 3.3. We use direct modelling to synthesize the forward or non-recursive portion of the IIR filter, and inverse modelling to synthesize the feedback or recursive portion. The transfer function has L zeros and M poles. The filter thus has $L+1$ feed-forward weights and M feedback weights. The goal is to develop an adaptive process that will automatically

adjust these weights so that the filter transfer function is a best fit to a set of design specifications [Wid85].

Figure 3.3: Block diagram of the adaptive IIR filter.

$$\frac{Y(z)}{X(z)} = \frac{A(z)}{1 - B(z)} = \frac{a_0 + a_1 \cdot Z^{-1} + \dots + a_L \cdot Z^{-L}}{1 - b_1 \cdot Z^{-1} - \dots - b_M \cdot Z^{-M}} \quad (3.3)$$

Adaptive IIR algorithms have some associated difficulties. Performance improvements have yet to be satisfactorily demonstrated. The theory of adaptive IIR filtering algorithms is quite immature. The error surface for adaptive IIR filters, unlike adaptive FIR filtering, may not be unimodal. This, together with the problem of maintaining stability during adaptation, makes the adaptive IIR filtering problem much more difficult. Research is still ongoing to develop better adaptive IIR algorithms and filters. [Wid85].

A detailed description and comparison between the existing methods of IIR system identification will now be presented.

3.2.1 Adaptive FIR Approach

An adaptive FIR digital filter is shown in Figure 3.4. The adaptive filter, upon convergence of the adaptive process, will assume an impulse response that best satisfies a set of design specifications, which is tied to a plant modelling problem.

Figure 3.4: Block diagram of an adaptive FIR filter.

This approach is fairly well developed. Famous for its robustness, the design theory is already mature. The adaptive FIR filter is guaranteed to be stable, thus reliable application has been assured. Simple gradient search algorithms can be employed, however, mismatch between the adaptive filter and the plant is inevitable since only the zeros of the filter are adapted. To match a pole closely, the filter has to use an unreasonably large number of taps to compensate. The larger number of taps means increased computation, which is not desirable. Another problem associated with adaptive FIR implementation of IIR system is that the W_i of the filter (see Figure 3.4) are not

directly related to a_i or b_i in Equation(3.3). This limits the usage of the adaptive FIR approach in IIR system identification. The generation of an adaptive IIR approach is highly desirable.

3.2.2 Output error approach

Figure 3.5 shows a filter synthesis scheme. This is a fairly convenient and simple structure.

$$E(z) = D(z) - \frac{A(z)}{1 - B(z)} \bullet X(z) \quad (3.4)$$

Errors (Equation(3.4)) are directly generated by comparing the plant output with the adaptive synthesized IIR filter $\frac{A(z)}{1 - B(z)}$, therefore, both $A(z)$ and $B(z)$ must adapt to the plant. This will ensure a theoretically unbiased solutions.

Figure 3.5: Simple adaptive IIR output error scheme.

LMS is used for coefficients updation. Where:

$$A(n+1)=A(n)+u x(n) e(n) \quad B(n+1)=B(n)+u y(n) e(n) \quad u: \text{step size}$$

There are also associated difficulties with this approach. Stability and global convergence are not guaranteed. Instability may occur when IIR poles are outside the unit circle. If a particular application requires that the poles be located close to the unit circle, and if adaptation is too rapid, one or more poles could accidentally move outside the unit circle and lead to a potentially unstable filter. Furthermore, local optima may exist in some cases. There is a well known conjecture by Stearns based on numerical experiments, which states that “if the adaptive filter is of ‘sufficient order’ and if the driving input $x(t)$ is white noise, then the error surface $E\{e^2(n)\}$ is unimodal”. [Ste81]. With all other cases, the error surfaces may be multimodal or unimodal. In general, knowledge about error surfaces is quite limited.

3.2.3 Equation Error Approach

To overcome the associated difficulties with the output error approach, another scheme has been presented. The structure is shown in Figure 3.6. It is called “simultaneous direct and inverse modelling” or “equation error” modelling. In this scheme, $A(z)$ and $B(z)$ are adjusted separately as adaptive transversal filters. The mean square value of $e'(z)$ is minimized. Notice that $e'(z)$ is different from $e(z)$ in the output error approach given in Figure 3.5. However, in most cases, the optimum values which $A(z)$ and $B(z)$ settles to are very close to minimizing the mean square of $e(z)$.

Accordingly, once $A(z)$ and $B(z)$ are found from Figure 3.5, the IIR filter is constructed by copying the values of $A(z)$ and $B(z)$ into the system of Figure 3.3.

Figure 3.6: Equation error method of IIR adaptive filter.

$$E'(z) = D(z) \cdot ((1 - B(z)) - A(z) \cdot X(z)) = \left(D(z) - \frac{A(z)}{1 - B(z)} \cdot X(z) \right) \cdot (1 - B(z))$$

referring to Equation(3.4), we can get:

$$\text{Equation error } E'(z) = E(z) (1 - B(z)) \quad (3.5)$$

This roundabout approach is used because the mean square of $E'(z)$ is a quadratic function of both $A(z)$ and $B(z)$ coefficients. The error in this case is unimodal, and so LMS adaptation can be easily used. Good convergence behaviour is expected, a unique solution is obtained and stability is robust. Note that $A(z)$ is adjusted in the “direct

modelling” mode to cancel the zeros of the plant, while $B(z)$ is adjusted in the “direct modelling” mode to cancel the poles of the plant.

One fatal disadvantage of this scheme is that good convergence is obtained at the expense of possible biased solutions. Figure 3.7 shows a better illustration. It is clear that $E'(z)$ and $E(z)$ are in proportion to the factor $1-B(z)$, which is not fixed but varies as $B(z)$ adapts. Accordingly, adjusting $A(z)$ and $B(z)$ to minimize the mean square of $E'(z)$ would not necessarily cause the mean square of $E(z)$ to be minimized. If $A(z)$ and $B(z)$ are designed with adequate numbers of weights, there would exist a solution so that the mean square of $E(z)$ is zero. These same settings would certainly bring the mean square of $E'(z)$ to zero. However, when an inadequate number of degrees of freedom is allowed for any of $A(z)$ or $B(z)$, or if there is noise $V(z)$ presented in the plant as shown in Figure 3.7, which happens commonly in a real case, the solution of the equation error method will be biased.

Figure 3.7: Equation error versus Output Error.

3.2.4 Improved Versions of Equation Error Family

There are several different versions of adaptive IIR filters in the Equation Error family, a brief comparison is provided in the following table, and a more detailed description of each algorithm is followed.

	Advantage	Disadvantage
A new adaptive IIR filter	Good stability.	Biased solution with noise or under-modelling.
Sign-sign EE	Simplicity.	Bad stability.
Parallel -Form Realization	Good stability.	1)High complexity. 2)malfuctions for repeated poles 3)Diverge in certain conditions.

	Advantage	Disadvantage
BRLE or CRA	Better stability than the output error method.	1)Compromised stability. 2)Biased solution with noise. 3)Bad stability for high order poles or strongly correlated signals.
QCEE	Good stability.	1)Coefficients are scaled. 2)Two stage filtering needed for non-white noise.

3.2.4.1 A New Adaptive IIR Filter

The equation error method mentioned above has good convergence. It is usually parabolic with one optimum. To take advantage of this, in [Hon86] the idea of

transforming an IIR adaptive structure into an equation error style is presented. The first cut design is given in Figure 3.8.

Figure 3.8: Theoretical mode of the new adaptive IIR filter.

For slow adaptation, $E(z)$, the equation error with respect to $X'(z)$, can be thought of as the output error for the input $X(z)$. Thus the quadratic error surface is preserved. Note that since additive noise $V(z)$ is not measurable, the $V'(z)$ can not be calculated either. To get around this problem, another version of the algorithm is given in Figure 3.9.

This structure saves some complexity since the number of extra-filters is reduced from three to two.

Figure 3.9: Adaptation mode of the new adaptive IIR filter.

Simulations show that global convergence is obtained regardless of local minima. However, stability is not guaranteed by the proposed algorithms themselves, and a stability monitoring device should be incorporated with these algorithms. Fortunately, with sufficiently small adaptation steps, little instability has been encountered. The theoretical proof on this is still being investigated.

This method also suffers from the same fatal drawback as for the Equation error family, that is, when $V(z)$ is not zero, especially in the case where it is not in white noise distribution or if at some time the plant is not sufficiently modelled, the final solution will be biased.

3.2.4.2 Sign-sign Equation Error Identifier

Most of the realizations of adaptive IIR filters are relatively complicated. To minimize complexity, a sign-sign equation error method was presented in [Sou93]. It is obtained by introducing sign functions on both the regressor and the prediction error multiplicands in the update kernel of the LMS algorithm. This greatly reduces the amount of calculation, however, performance degradation may occur and simulations show that the algorithm may diverge in some cases.

3.2.4.3 Parallel-Form Realizations

Parallel-form realizations for adaptive IIR filtering can be generalized as in Figure 3.10. In [Shy89], a frequency-domain adaptive IIR filter has been presented. The method employs a DFT that operates as a bank of bandpass filters to preprocess the input signal and transform a wide-band adaptive filter into several narrow-band adaptive filters. Gauss-Newton (GN) adaptation was used.

The primary advantage of the parallel configuration is that stability monitoring is not necessary comparing to that of the direct form. All stable poles are updated, while all unstable poles can be projected back inside the unit circle while updating. As a

Figure 3.10: Parallel form adaptive filter.

consequence, stability monitoring in this configuration is robust and it does not increase the complexity of the algorithm. It is also demonstrated that the adaptive filters can model any rational system with distinctive poles.

Although the parallel form has several advantages, there is a potential drawback with the structure in Figure 3.10. If the subfilters are identically initialized, then it is possible that the GN algorithm will not converge. Preprocessing can be added to solve this problem. It should also be noted that preprocessing of the input does not guarantee convergence, but it appears that certain types of preprocessing can improve the convergence properties of the GN algorithm. Another fatal problem of this scheme is that

the parallel-form generally does not work for systems with multiple repeated poles (or un-split poles). As we will see later in chapter 4, the delta-sigma modulator coefficients are not well determined for each case, they are likely to constitute an IIR transfer function with un-split poles. This greatly reduced the usability of this algorithm. In addition, the preprocessing will inevitably increase the complexity of the algorithm.

3.2.4.4 BRLE or CRA

It is well known that the equation error algorithms used for recursive estimation of an unknown plant are regarded as having good convergence behaviour. (i.e., robust stability of the parameter estimates and uniqueness of the convergent solution). However, they are plagued by the problem of biased parameter estimates caused by inevitable non-zero disturbances. The BRLE (Bias-Remedy Least Mean Square Equation Error algorithm) [Nan92] and CRA (Composite Regressor Algorithm) [Ken89] families are based on the LMSEE (Least Mean Square Equation Error) algorithm, and manage to remedy the bias without sacrificing much of its convergence property. They can be considered as a composition of the LMSEE and output error method through a so-called “remedy parameter” of the range between zero and one. This is done by transforming the updating equation so that when the remedy parameter is zero, the updating equation turns into LMSEE method and when the remedy parameter is one, the updating equation is close to the output error method.

Since the BRLE is installed by the remedy parameter between two methods which have different stability properties, it is easily seen that its stability is strongly affected by

the parameter. Simulations show that the BRLE suffers from the inherent drawback of the LMS-type algorithms with strongly autocorrelated signals. and its convergence becomes worse with high-order poles of the plant near the unit circle.

3.2.4.5 Bias-Removal in Equation Error Adaptive IIR Filters

QCEE (Quadratically Constrained Equation Error IIR Filter) is a typical method of this family [Ho96]. The idea of bias removing is to maintain a quadratic constraint on the feedback coefficient so that the noise contributes only a constant term to the mean-square error. This term does not affect minimization and thus the bias is eliminated. A quadratically constrained stochastic gradient search method is applied for optimization and convergence behaviour.

Research shows the stability condition of the technique is good. A unique unbiased solution exists when the noise is white and the model order is sufficient. An extension of the method to handle the non-white noise situation, which was rarely addressed before, is also explored in this paper [Ho96]. When the noise is not white, the technique requires an adaptive noise whitening filter. Two stage filtering will certainly have impact on stability, so the additional filter is preferable a FIR type.

Another big disadvantage which makes this method unusable in many application is that the converged parameters are scaled. This is extremely inconvenient for some cases which needs precise coefficients.

3.3 Summary

The purpose of this chapter has been to introduce the basic concepts of system identification. A detailed review of adaptive FIR approach, Equation Error, Output Error, Modified Equation Error algorithms, including their properties of stability, solution characteristics, computational complexity and robustness is provided. The advantage and disadvantage of each algorithm as well as some of the issues involved in the choice of an adaptive IIR filtering algorithm are outlined. Emphasis was placed in providing a simple and general explanation that enables easy understanding of the interrelationships and convergence properties of the algorithms.

Through extensive research on this subject, it seems that no general purpose optimal algorithm exists. In fact, all available information must be considered when applying adaptive IIR filtering in order to determine the most appropriate algorithm for a given problem.

Chapter 4. Characterization and Digital Correction in Delta-sigma Modulators

Introduction:

In this chapter a system identification method will be introduced to characterize the delta-sigma loop, it may be very useful to figure out the actual design gain of each op-amp, nonlinearity coefficients, etc. Also, a digital correction method to improve the nonlinearity of multi-bit delta-sigma modulator will be discussed in detail.

4.1 Delta-sigma Loop Identification

4.1.1 Selection of the Adaptive Method

From the detailed description and comparison of different algorithms in the last chapter, we can see that the adaptive FIR will not render the exact coefficients as we required to characterize a delta-sigma modulator (i.e. leaky coefficients, gain and nonlinearity parameters), therefore it is not very helpful in our situation. The equation error approach, has better stability, however is prone to give a biased optimum in the presence of noise. Furthermore, none of the improved versions of the equation error family we have seen so far works for the delta-sigma modulator loops since the quantization noise, although white, is measured to be coloured noise at the output, and it is coloured by an IIR filter. For this reason, the convergence is always biased by using the above methods.

Therefore, the output error structure, despite its poor stability and limited performance with poles outside the unit circle, is the only valid method to be used on delta-sigma loop identification. It is the only method that will have a chance to provide an unbiased solution. An added benefit is that the algorithm works for both IIR filters with either repeated poles or split poles.

For coefficients update, the LMS method is chosen for its robustness and simplicity. A common limitation of the LMS family is that the input of the adaptation

has to be rich in frequency domain. This is satisfied by injecting a white noise into the delta-sigma loop as the input for characterization.

4.1.2 Selection of Dither Location

As mentioned above, LMS adaptation requires the input to be rich in the frequency domain. A pseudo-random sequence will suffice for this purpose and in a delta-sigma modulator, this is called random dither. There are several places that dither can be introduced into the system, but the trade-offs must be considered. All the possible locations for injecting dither are shown in Figure 4.1. $H(z)$ is assumed to be known.

Figure 4.1: Possible dither adding points on an delta-sigma loop.

$Q(z)$ stands for quantization error (or quantization noise). $k1$ and $k0$ are the non-linear factors. $D1$ to $D4$ are the dithering signals to be added. The following equations describe the resulting output signal Y due to each of the dithering signal input.

Dither added at the input. ($D1$):

$$Y = \frac{H(z) \cdot k_0 \cdot D1 + H(z) \cdot k_0 \cdot X + Q}{1 + k_0 \cdot k_1 \cdot H(z)} = \frac{H(z) \cdot k_0 \cdot D1}{1 + k_0 \cdot k_1 \cdot H(z)} + Noise \quad (4.1)$$

Dither added before the quantizer. (D2):

$$Y = \frac{k_0 \cdot D2 + H(z) \cdot k_0 \cdot X + Q}{1 + k_0 \cdot k_1 \cdot H(z)} = \frac{k_0 \cdot D2}{1 + k_0 \cdot k_1 \cdot H(z)} + Noise \quad (4.2)$$

Dither added from the digital side (D3):

$$Y = \frac{D3 + H(z) \cdot k_0 \cdot X + Q}{1 + k_0 \cdot k_1 \cdot H(z)} = \frac{D3}{1 + k_0 \cdot k_1 \cdot H(z)} + Noise \quad (4.3)$$

Dither added from the digital side (D4):

$$Y = \frac{k_0 \cdot k_1 \cdot H(z) \cdot D4 + H(z) \cdot k_0 \cdot X + Q}{1 + k_0 \cdot k_1 \cdot H(z)} = \frac{k_0 \cdot k_1 \cdot H(z) \cdot D4}{1 + k_0 \cdot k_1 \cdot H(z)} + Noise \quad (4.4)$$

As we care more about the relationship of dither versus the output Y in the following context, all the other terms are represented as “Noise”.

For the case where dither is added from the analog side, that is, from the input (D1) or before the quantizer(D2), the advantage is that k_0 and k_1 can be identified separately as shown in Equation(4.1) and Equation(4.2). However, it is hard to generate analog random sequences, and it is also hard to use analog dither in the digital domain for calculation directly. Besides, input and DC offset at the first op-amp are mingled with the dither signal and this can directly affect convergence of the algorithm and render biased coefficients. On the other hand, if dither is added from the digital side, it will be easily generated digitally as a quantized signal. A random sequence generator (or PN sequence

generator) can be used to conveniently generate pseudo-random noise. The length of the sequence is preferably as long as possible since longer sequences will eliminate more periodical effects of the sequence repetition. The other advantage is that digital dither can be loaded in the adaptive algorithm directly and dc offset can be considered irrelevant to the dither, thus, it will not affect the final convergence. The disadvantage is that k_0 and k_1 in Equation(4.3) and Equation(4.4) appears as product term only. It is therefore hard to identify them individually.

To run the system identification method on-line, we also care about what effects the dither has on the delta-sigma loop. As the loop filter $H(z)$ is usually designed to have a high gain, linearity factors k_0 and k_1 are ideally to be one. From Equation(4.1) to Equation(4.4) it can be seen that D_1 and D_4 have approximately unit gain while D_2 and D_3 have much lower gain. (i.e., D_2 and D_3 are noise shaped). Considering that $X(z)$ is the real input of the delta-sigma modulator, SNR (signal-to-noise ratio) is the power of input signal $X(z)$ divided by the total output noise power, which is made up of the total of the quantization noise and dither power, with the same input and dithering power, dither signals applied at D_2 or D_3 locations will result in higher SNR.

Table 1 lists all the benefit and shortcoming of each dither position. As shown, D3 provides the best overall performance.

Table 1: Comparison Between Dither Spots

	Analog side D1	Analog side D2	Digital side D3	Digital side D4
Usability in digital domain	No	No	Yes	Yes
Circuit complexity	Complex	Complex	Simple	Simple
Additional quantizer needed?	Yes	Yes	No	No
Susceptibility to interfer- ence	High	High	Low	Low
Parameters identification	Good	Good	Fair	Fair
SNR degradation	High	Low	Low	High
Total performance	Fair	Good	Best	Better

Although dithering seems to degrade the SNR compared to the situation where there is no dithering, there is actually a big advantage to using it. The reason is that the quantization noise from delta-sigma modulation is not white in a strict sense. It has periodic components related to the input signal frequency and amplitude. After going through the delta-sigma modulation, the periodic structure of the quantization error is worsened by inter-modulation. Small amount of dithering randomizes those patterns and

spreads energy around, which helps to reduce those aliasing effect without sacrificing much of the SNR.

While such dithering is very useful and practical, there are drawbacks as well. When an input is already large, adding an extra dither may overload the quantizer, clipped signals will be generated and this will introduce more nonlinearity. For higher order delta-sigma modulators, larger dither may also cause instability. To avoid these, a small amount of low frequency dithering, typically a few LSBs, is preferred [Bai95].

4.1.3 Simulation Examples

A number of system level simulations have been carried out. Example one is given in Figure 4.2. A second order low-pass delta-sigma modulator is shown. Coefficients α and β are the leaky coefficients of the op-amps. They are used to model a

Figure 4.2: An 2nd-order low-pass sigma-delta modulator.

non-ideal op-amp. Coefficients k_0 and k_1 are quantizer and DAC gains, respectively. A 3-bit quantizer is used and a dither with an amplitude of the LSB is used. If the dither is added at the digital side after the output, the system equation can be derived as:

$$Y = noise + \frac{(-2) \cdot k_0 \cdot k_1 \cdot Z^{-1} + \alpha \cdot k_0 \cdot k_1 \cdot Z^{-2}}{1 - [(\alpha + \beta) - 2 \cdot k_0 \cdot k_1] \cdot Z^{-1} - (\alpha \cdot k_0 \cdot k_1 - \alpha \times \beta) \cdot Z^{-2}} \times D \quad (4.5)$$

$$\text{or: } Y = noise + \frac{A_1 \cdot Z^{-1} + A_2 \cdot Z^{-2}}{1 - B_1 \cdot Z^{-1} - B_2 \cdot Z^{-2}} \times D \quad (4.6)$$

From Equation(4.5), we can see that the coefficients of the Z terms can be obtained and thus the product of $k_0 \cdot k_1$, α and β can be determined if the system identification converges. Simulation results are given in Figure 4.3 using the output error method. In an ideal case, there should be no leakage in the op-amps and no gain offsets. hence $\alpha=\beta=k_0=k_1=1$, and coefficients A_1 , A_2 , B_1 , B_2 converge to -2, 1, 0, 0 respectively as shown in figure (a). In the case where there is some non-unity gain inside the quantizer, for example, $k_1=0.9$, the coefficients A_1 , A_2 , B_1 , B_2 converge to -1.8, 0.9, 0.2, and -0.1 as in Figure (b). If the dither is added to other spots as mentioned in Table 1,

similar convergence is obtained. This example shows us that the output error method can give an un-biased solution in the presence of quantization noise.

Figure 4.3: Simulation results for the second order low-pass sigma-delta modulator.

(a) Ideal case, step size 0.05. (b) $k_1=0.9$, step size 0.05.

Example two is a fourth-order band-pass sigma-delta modulator. (see Figure 4.4).

Figure 4.4: Simulation model of a fourth-order band-pass sigma-delta modulator.

A sine wave of amplitude 0.1 at an in-band frequency was applied to the input of the sigma-delta modulator. The constant 0.04 was used to model the dc off-set errors at the inputs of the op-amps in a real circuit. A gain of “10/8” was used to unify the random input at the LSB level. A repeating sequence may also be used as dithering input in simulation to model the hardware limitation on the implementation of a pseudo random sequence generator and see how the frequency of dithering would affect the convergence of the adaptive algorithm. This is because a pseudo random sequence is usually

implemented as a chain of flip-flops (F.F.) as shown in Figure 4.5. A maximum length sequence so generated is always periodic with a period of $N=2^m-1$.

Figure 4.5: Pseudo noise generator.

To verify the level “SNR degradation” as identified in Table 1, simulations were done for different dithering locations. The results are shown in Figure 4.6. Figure 4.6 (a) shows the result of the case when there is no dither. It can be seen that many unwanted intermodulation tones are generated. Figure 4.6 (b) shows the output of the delta-sigma modulator with noise-shaped dither added. (i.e. Equation(4.2). Equation(4.3)). The tones are smoothed out by the random dither although with an extra cost of a few dBs SNR degradation. However, in some digital audio applications, a smoothed spectrum is preferred to a higher SNR with lots of aliasing tones in the band of interest. In Figure 4.6 (c), the noise level is elevated because the dither added to the delta-sigma loop is not

noise-shaped and modulates with the out-of-band noise around $f_s/2$ back into the band of interest.

Figure 4.6: Output comparison for the 4th-order bandpass delta-sigma modulator.

Simulated with dither applied at various location: (a) No dithering. (b) Dithering before the output y . (D2 or D3). (c) Dithering after the output y . (D1 or D4).

By comparison, the best place to add dither is point D3, which is after the quantizer and before the output Y . The output versus dither transfer function is given as follows:

$$Y = Noise + \frac{1 + (\alpha + \beta) \cdot Z^{-2} + \alpha \cdot \beta \cdot Z^{-4}}{1 - (2 \cdot K_0 \cdot K_1 - (\alpha + \beta)) \cdot Z^{-2} - (\alpha \cdot K_0 \cdot K_1 - \alpha \cdot \beta) \cdot Z^{-4}} \times D \quad (4.7)$$

$$\text{or: } Y = \text{Noise} + \frac{A_0 + A_2 \cdot Z^{-2} + A_4 \cdot Z^{-4}}{1 - B_2 \cdot Z^{-2} - B_4 \cdot Z^{-4}} \times D \quad (4.8)$$

The algorithm converges to the same set of values with or without the input sin-wave and dc-offsets. The difference is that when the loop is presented with the input and dc-offsets, the plot of the coefficient convergence is slightly noisier. This is because input and dc-offsets are noise term only, they will not affect the optimum solution to the adaptive LMS algorithm. Figure 4.7 shows the noisy version of the convergence curves (i.e. with the sin-input and dc-offsets) with iteration step size of 0.001. Again, in an ideal case, the coefficients should converge to 1, 2, 1, 0, 0 as in Figure 4.7 (a). If the op-amps are leaky, say both have a leaky coefficient of 0.9, then the coefficients should converge to 1, 1.8, 0.81, 0.2, and 0.09 respectively, which is verified in Figure 4.7(b).

Simulations also show that with this specific structure, a minimum length of 28 is required for the repeating sequence to guarantee convergence. This information tells us what is the minimum random generator length we should use and therefore how many D flip-flops we need for implementation. In this case, we can calculate that 5 D flip-flops are required to cover a non-repeating sequence of 28. Simulations also show that with a lower magnitude of dithering, convergence can still be maintained, but with a much lower speed.

Figure 4.7: Simulation results for the 4th-order bandpass delta-sigma modulator.

(a) Ideal case, step size 0.001. (b) $\alpha=\beta=0.9$, step size 0.001.

A few limitations were noted during the simulation as well that we need to be aware of to ensure that the system identification method is working properly. The general rule is that we do not want to lose too much information in the system or add too much noise to the system during adaptation. In a delta-sigma loop, the quantizer, which is the link between the analog signal and the digital signal, plays a vital part. There are two critical factors in the quantizer. One is the choice of quantization step. It should be chosen small since larger quantization steps bring in larger quantization noise and this can interfere the algorithm. The simulation results show that a multi-bit quantizer with three

or more bits is good enough to have little impact on coefficient convergence. The other critical factor is the saturation level of the quantizer. The saturation level is preferably chosen to be high enough so that the input analog signal will not be hard-limited, otherwise, we are losing some information by reaching the limit. Note that the term “saturation level” is a misnomer since it comes from a simulation point of view only. From the circuit point of view, we do not want the quantizer to be overloaded. This is not only important for system identification mentioned above, but also important for delta-sigma loop stability. Low amplitude and low frequency dither is preferable for stability reasons, although this will certainly lower the speed of the algorithm convergence and also makes it more susceptible to noise interference. Here low frequency means longer period of random sequence, i.e., slower repetition rate.

4.2 Digital Correction of a Crystal Semiconductor Circuit

4.2.1 Theory Analysis

As mentioned in Chapter 2, most known calibrating techniques for achieving high linearity require some kind of highly linear calibration reference. The precision of the calibrated DAC is limited by the precision of the linearity reference, and it is difficult to obtain a good linearity reference. The approach mentioned in a Crystal patent [Tho94],

eliminates the need for a calibration reference. The remainder of this section, will address this technique.

The family of switched-capacitor structure delta-sigma analog-to-digital converters with 3-level internal quantizers is discussed in the paper [Tho96]. The thermal-noise current of a switched capacitor exhibits the same thermal noise as an actual continuous-time resistor in the same bandwidth. As the capacitor is repeatedly charged and discharged during each clock cycle, the thermal-noise advantage gained by using a tri-level quantizer, namely +1, 0, -1, comes from the fact that the middle quantizer level is a “do-nothing” state. Maximizing the density of zeros around the modulator loop results in an effective thermal noise reduction. This noise reduction is realized simply by dumping charges to ground for the middle quantizer state.

The noise of the feed-back DAC dominates the total performance of the converter since it is directly connected to the input, whereas the multi-bit quantizer error is greatly reduced by noise-shaping. Noise introduced by the DAC can be modelled as linearity error, gain error and dc-offset error. Calibration can easily correct for gain error. DC-offset error, which will be addressed later, also turns out to be unimportant. Hence our main concern is the linearity error.

The output stream of the converter is a combination of +1, 0, and -1. Suppose the charge on the -1 side is k times less than the charge on the +1 side in the DAC as shown in Figure 4.8. After the negative feedback, the charge on the +1 side will be k times less than

the charge on the -1 side. We can either multiply the -1 side by k or multiply the +1 side by $1/k$ to bring both sides on the same line again and eliminate the linearity error.

The main problem is finding the correct value of the compensation coefficient W

Figure 4.8: Illustration of DAC output versus modulator output.

as shown in Figure 4.9. The structure in Figure 4.9 provides a viable solution. The output data stream is split into two data paths. Quantizer positive 1's are routed to one decimation filter while negative 1's are routed to a second decimation filter. Zeros are applied to both filters. W is updated using least mean square adaptation.

The theoretical model of the system is depicted in Figure 4.10. The quantizer is represented as a gain K_0 plus white quantization noise. The DAC is modelled with a non-linear gain K_1 . We can derive the equation of the output $Y(z)$ as:

Figure 4.9: Block diagram of correcting and decimator chip.

Figure 4.10: Theoretical model for the system.

$$Y(z) = [X(z) - (y_1 + K_1 y_2)] H(z) K_0 + Q(z)$$

$$= [X(z) - (y_1 + y_2 - (K_1 - 1) y_2)] H(z) K_0 + Q(z)$$

$$Y(z) = y_1 + y_2 = \frac{H(z) \times K_0 \times X(z) + Q}{1 + H(z) \times K_0} + \frac{(1 - K_1) \times H(z) \times K_0 \times y_2}{1 + H(z) \times K_0} \quad (4.9)$$

In the ideal case, K_1 should be one and the second term disappears. When nonlinearity exists, K_1 is no longer one and the second term appears as in-band noise.

In the non-ideal case, W is used to compensate K_1 . The final $Y_0(z)$ is represented as follows:

$$Y_o(z) = y1 + (W \times y2) \text{ so:}$$

$$Y_o(z) = \frac{H(z) \times K_o \times X(z) + Q}{1 + H(z) \times K_o} + \left[\frac{(1 - K_1) \times H(z) \times K_o}{1 + H(z) \times K_o} - (1 - W) \right] \times y2 \quad (4.10)$$

The first term is equal to that of Y in Equation(4.9) which in an ideal case, consists of a signal and quantization noise. The second term is additive noise. Proper choice of W can minimize passband quantization noise. The loop filter $H(z)$ is often designed with high gain, K_0 is approximately one so that $\frac{H(z) \times K_o}{1 + H(z) \times K_o}$ is near unity. When W is chosen to be nearly equal to K_1 , the second term is greatly reduced and the corrected output Y_0 is nearly equal to the ideal output Y .

Least mean square adaptation is used to minimize the total power of Y_0 in the frequency domain. To insure that Y_0 is minimized, input X has to be zero or out of the band of interest so that it can be removed later by the decimation filter. This insures the adaptation is concentrated on minimizing in-band noise only, otherwise the optimum solution of W will be biased.

If this input has a DC component, which is very common in analog circuit, it will

Figure 4.11: The general block diagram of the calibration system.

also affect the convergence of W . This can be solved by adding a high-pass filters on paths y_1 and y_2 . The complete calibration model is illustrated in Figure 4.11.

A control circuit controls the overall operation, with the calibration operation initiated in response to either an external signal or an internally generated signal. In calibration mode, the input MUX selects the calibration signal, which is often ground or an out of band signal. The calibration signal passes through the delta-sigma modulator with its energy concentrated in the passband of the decimation filter and with power proportional to the variance among capacitor values. The switches are connected to the high-pass filters to eliminate possible dc-offset and the LMS adaptive engine begins operating. Once W converges within a given tolerance, the calibration is over and W is

frozen and stored. The system is then ready for operation. The MUX selects the analog input, high-pass filters and the adaptation engine are bypassed and the final corrected output is Y_0 .

4.2.2 Simulation Examples

A Simulink simulation example was made on a 4th-order band-pass delta-sigma modulator as shown in Figure 4.12. To simplify the procedure, the sampling frequency was normalized to 1 Hz. The centre frequency was at $f_s/4$ for a band-pass delta-sigma modulator, which is 0.25 Hz. Using an oversampling ratio of 32, the band of interest is 1/64 Hz. As an out-of-band input signal is required, a sin-wave with input amplitude of 0.2 and a frequency of 1/32 Hz was sent into the delta-sigma loop. DC offset input errors were modelled as 0.04. The quantizer step was chosen to be the same as the saturation limit so that the output Y would be +1, 0, and -1. A “Sep” component was constructed to model the nonlinearity of feed-back DAC. It divides the +1, 0, and -1 output into two paths so the balance of each path can be easily changed.

The post-processing algorithm is given in Figure 4.13. The output of the delta-sigma modulator is loaded in and then divided into two paths. A cosine-wave of frequency 0.25 Hz is used to demodulate the signal to base-band, then followed by a decimator before going into the adaptive engine. This assures the out-of band noise is removed before adaptation.

Results are shown in Figure 4.14. If in the system diagram of Figure 4.12, the “Sep” component was set to model an unbalanced feedback DAC with the -1 path being 0.95 less than +1 path, The LMS adaptive engine will eventually make W converge to 0.95 despite the presence of the DC offsets. (See Figure 4.14(a)). Figure 4.14 (b) shows the power spectrum density of the output under normal operation before correction. The notch can only go down to -45dB, while after digital correction as shown in Figure 4.14 (c) the notch can reach -65 dB.

Simulations show that the algorithm also works for low-pass delta-sigma modulation. In that case, it is even simpler since no demodulation is needed.

Figure 4.12: A 4th order band-pass delta-sigma modulator with a 3-level quantizer.

Figure 4.13: General algorithm structure for band-pass delta-sigma modulator nonlinearity cancellation

Figure 4.14: Simulation example for nonlinearity cancellation.

(a) Convergence of W. (b) Before Correction. (c) After correction.

Simulink simulations of the digital correction technique were also made on a model of a sixth-order band-pass delta-sigma modulator as shown in Figure 4.15. The coefficients were carefully chosen such that the sixth-order band-pass delta-sigma modulator was stable. A single-tone sine-wave was fed as the input signal. A constant of 0.03 was fed into the input to imitate dc-offset. The output bit-stream was captured into variable y . The Sep component was used to separate +1 and -1 branches and multiply it with scaling factors so that it could simulate both linear and non-linear circumstances.

The convergence results are shown in Figure 4.16. If the Sep component in the system diagram of Figure 4.15 is toggled to model an unbalanced feedback DAC (with

the -1 path being 0.9 less than +1 path), the LMS adaptive engine will eventually make W converge to 0.9 despite the presence of the dc-offset. Figure 4.16 (b) shows the significant improvement using a corrected output versus an uncorrected output.

Figure 4.15: A 6th order band-pass delta-sigma modulator with a 3-level quantizer.

Figure 4.16: Simulation example: (a) Convergence of W . (b) Corrected vs. uncorrected output.

Hardware testing was also performed on this example. The circuit structure and hardware implementation of this sixth-order band-pass modulator chip can be found in “High-order Band-pass Sigma-Delta Modulators” [Ste98]. Unfortunately, this chip, designed by a former Carleton graduate student, was not operational. This made any testing impossible.

In general, digital correction methods provide an effective solution to system nonlinearity. Most known calibrating techniques for achieving high linearity rely on some kind of highly linear calibration reference, which is hard to obtain. The Crystal

Semiconductor approach eliminates the need for precise component matching or precise calibration references at the expense of increased digital complexity. The technique is particularly attractive as it can be generalized to multi-level (≥ 3) low-pass and band-pass cases with no dc-offset problems. As the number of quantization level increases, the number of taps should also be increased correspondingly. One disadvantage of the Crystal technique is that the adaptation is off-line. The value of W is frozen after each calibration period, which prevents it from continuous tracking variation due to factors such as temperature and supply voltage changes.

4.3 Summary

In this chapter, we first looked at the innovative delta-sigma loop identification technique. It is very useful in determining the actual gain of each op-amp, to spot leaky op-amps, or to determine the nonlinearity coefficients. The output error method was chosen. Simulations have shown that the output error system identification method may work for both low-pass and band pass second or higher order delta-sigma modulators. However, it does not work for delta-sigma modulators with poles on or outside the unit circle. Note that there are some delta-sigma modulators that are designed to work with poles outside the unit circle. Simulations also show that the convergence of the identification method is sensitive to the number of bits used in the quantizers and DACs. Although the use of dither is necessary for the adaptive LMS algorithm to converge, the advantages of reducing aliasing and improving the SNR for the final outputs are

significant. As for dither, low amplitude and low frequency dither is preferable for stability reasons, although this will certainly lower the speed of the algorithm convergence and also makes it more susceptible to noise interference. In the second part of the chapter, we looked at the principle analysis of the Crystal Semiconductor digital correction patent. Simulation results were provided to support this analysis.

Both system identification and the Crystal Semiconductor techniques are useful to characterize the linearity factors in the delta-sigma loop. The system identification method is more suitable for multi-bit modulators with more than three bits, while the Crystal method is most effective for nonlinearity corrections in systems with fewer bits. The system identification method works on-line, while the Crystal method has to be used off-line.

Both techniques have been implemented in Simulink and in C. The C version requires less memory and has faster speed. The source code is attached in Appendix A and B. As well, both techniques were taken to the hardware testing stage, which will be addressed in more details in the next chapter.

Chapter 5. Hardware Testing

5.1 System Identification

5.1.1 Simulation Result

A second-order band-pass delta-sigma modulator is chosen on which to perform hardware tests as the second order delta-sigma is known to be stable. The system level block diagram is shown in Figure 5.1.

Figure 5.1: Block diagram of a second-order band-pass delta-sigma modulator.

The transfer function can be derived as:

$$Y = \frac{(1 + Z^{-2}) \cdot (D + Q)}{(1 + 0.5 \cdot Z^{-2})} + \frac{X}{(1 + 0.5 \cdot Z^{-2})}$$

$$\text{or: } Y = \frac{(A0 + A1 \cdot Z^{-1} + A2 \cdot Z^{-2}) \cdot (D + Q)}{1 - (B1 \cdot Z^{-1} + B2 \cdot Z^{-2})} + \frac{X}{1 - (B1 \cdot Z^{-1} + B2 \cdot Z^{-2})}$$

where: X stands for input signal

Y stands for output signal Yout

D stands for Dither

Q stands for Quantization Noise

A0, A1, A2 stand for coefficients on the nominator

B1, B2 stand for coefficients on the denominator

The system was first simulated. The input signal was a single sine wave tone at a frequency of 259.77e3 Rads/sec with amplitude of 1/8. The saturation level was set at plus and minus 0.5 and a 3-bit quantizer was used giving a quantizer interval of 1/8. The amplitude of the dither input was half of that of the input signal. An adaptation step size of 0.01 was chosen and DC offset was added at the input. The convergence curves are

quite similar with or without the dc-offset. (See Figure 5.2). Again, as discussed in the

Figure 5.2: Simulation result of coefficients convergence.

last chapter, this confirms that this configuration is less sensitive to interference at the input of delta-sigma modulator than the other configurations.

The convergence curves at the same conditions without any input or dc-offset are much smoother as expected. (See Figure 5.3 (a)). As can be seen from Figure 5.3 (b), a quantizer with lower number of bits has more quantization noise, thus the coefficients

convergence is noisier and biased. As the lower power of Z terms were adapted first, it

Figure 5.3: Simulation results: (a) With a 5-bit quantizer. (b) With a 2-bit quantizer.

can be seen that the relative tolerance of A2 and B2 (i.e. the percentage drift of final

convergence value to the expected value) was increased. The average tolerance versus number of bits is plotted in Figure 5.4.

Figure 5.4: Tolerance to number of quantization bits ratio.

5.1.2 Circuit Structure and Tested Results

Due to the time limit before the fabrication deadline, layout level design of the complete modulator was not possible. Instead, the testing circuit was built at the board level. A diagram of the complete system is shown in Figure 5.5. The integrator part of $H(z)$ in Figure 2.1 is contained on chip as shown in Figure 5.5. It is modified from an existing double-sampling delta-sigma modulator. The top-level schematic of the chip is provided in Figure 5.6. It has differential inputs, clocks and differential outputs. The

layout view of the chip is shown in Figure 5.7. It is implemented in a 0.35 μm CMOS process. Other components necessary to build the delta-sigma loop are multi-bit A/D and multi-bit D/A. An ADS8xxE analog-to-digital converter unit from Burr-Brown was used as the multi-bit quantizer. An AD9764 D/A board from Analog Devices was used as the multi-bit feedback D/A. An ALTERA FPGA board was used as the link between the A/D and D/A and also serves as a digital processing tool.

Figure 5.5: Block diagram of the testing circuit.

Figure 5.6: Top level schematics of the chip.

Figure 5.7: Layout view of the chip in Cadence.

The first step in testing was to verify whether the chip was functioning. To

Figure 5.8: Block diagram of simplified testing circuit.

simplify the test, the loop structure of Figure 5.8 was used. Using this structure, the D/A was omitted and directly by-passed by the Altera board. This structure of the loop is similar to a one-bit double-sampling delta-sigma modulator. Correct biases and power supplies were provided to the chip, A/D converter and Altera board. The Altera board was programmed such that the MSB of the A/D board was sampled at both input clock edges and fed back to fb1p, fb2p of the chip while the compliments of the MSB were sampled at

both input clock edges and fed back to fb1n, fb2n. The circuit diagram of the block

Figure 5.9: Altera board configuration.

function inside the FPGA is shown in Figure 5.9. A digital logic analyser was used to capture the output bit-stream of the delta-sigma loop. The bit-stream was then processed in Matlab. Using an input sampling frequency of 1.2 MHz, the double sampling technique provides an effective sampling rate of 2.4 MHz. A differential pair of sine

waves with peak-to-peak amplitude of 400 mv and frequency of 530 kHz were fed into

Figure 5.10: Experimental result of the delta-sigma chip.

the input of the system. The power spectrum of the output signal is shown in Figure 5.10. We can see noise-shaping in the plot. Although the noise shaping performance is not great, it does demonstrate that the filter chip is working. The major reason for the deteriorated noise-shaping is believed to be noise picked up from the external A/D interface of the integrated loop. Also, the dc voltage levels of each board are different and require careful adjustment of extra bias supplies.

The second step of testing was to generalize the structure to a multi-bit level. In Figure 5.5, we can see the feedback pins fb1p, fb1n, fb2p, and fb2n. These signals are transistor switch inputs used to control the current flow from Vrefp and Vrefn, the reference voltage inside the chip. For multi-bit operation, fb1p and fb2p are tied high, while fb1n and fb2n are tied low. The multi-bit D/A outputs are fed directly into Vrefp and Vrefn. However, because of overloading and mismatch problem between the modulator chip and A/D interface, noise ends up filling in the notch. Although an output buffer may help to improve the noise-shaping, it will introduce an extra gain factor into the whole loop function and as a result it will be more difficult for the system identification to distinguish between the parameters. Another option is to feed the multi-bit D/A outputs directly into fb1p, fb1n and fb2p, fb2n and to use them to control the reference voltage. Unfortunately, the transistor switches are not linear and even if they were linear, it introduces an extra scaling factor. This scaling factor again makes the system identification not practical in this case.

5.2 Summary

In this chapter, we have presented experimental results for a board level integrated double-sampling delta-sigma loop. The filtering part of the modulator loop is realized on a chip implemented in a 0.35 μm CMOS process. The hardware difficulties as well as the simulation results are shown and discussed. Due to circuit noise and accuracy limitation, the experimental results can not be used to test the system identification algorithm.

Chapter 6. Conclusions and Possible Future Work

6.1 Contribution and Conclusions

There are two main contributions made in this thesis. First, a novel system identification approach is presented. It was implemented first in Simulink and then in C. This approach can help characterize a multi-bit sigma-delta modulator. The practical side of this method is that if a delta-sigma modulator is fabricated and does not work as well as predicted, then by applying this approach the gain of each stage can be calculated and compared to its original design to determine which parts inside the delta-sigma loop are not operating correctly. Examples of such undesirable operation might be a leaky switch or that the op-amp gain is too low. The second contribution is a theoretical explanation and analysis are given for the Crystal Semiconductor Digital Correction Patent.

Other contributions of this thesis includes a detailed comparison between the relevant existing techniques to solve the non-linear problem in a delta-sigma modulators. A detailed implementation of the Crystal Semiconductor patent on digital correcting the multi-bit nonlinearity error is given in both Simulink and C. Hardware testing was performed although not successful. Finally, simulation and hardware testing has been done on the system identification method to prove its practicability.

It can be concluded that both system identification and the Crystal Semiconductor digital correction techniques are useful to characterize the linearity factors in the delta-sigma loop. The system identification method is presently more suitable for multi-bit modulators with more than three bits, while the Crystal method is most effective for nonlinearity corrections in systems with fewer bits.

6.2 Future Work

A CMOS integrated multi-bit delta-sigma modulator chip can be made to further test the characterization methods mentioned in this thesis and FPGA could be made to implement the system identification algorithm. With further effort, the system identification method could be extended to multi-bit delta-sigma converters with less than 2 bits so that gains can be adjusted to minimize the nonlinearity.

Improvement can be investigated for the Crystal Semiconductor digital correction method to make it work on line and improve its accuracy and tracking ability.

References

- [Nor97] Steven R Norsworthy, Richard Schreier, Gabor C. Temes, "Delta-Sigma Data Converters-Theory, Design, and Simulation", 1997.
- [Can92] James C. Candy, Gabor C. Temes, "Oversampling Delta-sigma Data Converters", 1992.
- [Car89] L. R. Carley, "Trimming Analog Circuits Using Floating-gate Analog MOS Memory," IEEE J. Solid-state Circuits, Vol. SC-24, pp. 1569-1575, Dec, 1989
- [Sho96] Omid Shoaee, "Continuous-time Delta-sigma A/D Converters for High Speed Applications", Ph.D. Thesis, Carleton University, 1996.
- [Sne95] Martin Snelgrove, "Delta -sigma course notes", 1995.
- [Tho96] Charles D. Thompson, "Method and Apparatus For Calibrating a Multi-bit Delta-Sigma Modulators", United States Patent 5257026, 1996.
- [Sak93] Y. Sakina, "Multi-bit sigma-delta Analog-to-Digital Converters with Nonlinearity Correction using Dynamic Barrel shifting," Electronics Research Laboratory, College of Engineering, University of California, Berkeley CA, Memorandum No. UCB/ERLM93/63, 1993.
- [Leu92] B. H. Leung and S. Sutarja, "Multi-bit Sigma-delta A/D Converter Incorporating a Novel Class of Dynamic Element Matching", IEEE Trans. Circuits Syst.II, Vol.39, pp.35-51, Jan.1992.
- [Che95] F.Chen and B.H. Leung, "A High Resolution Multi-bit Sigma-delta Modulator With Individual Level Averaging," IEEE J. Solid-State Circuits, Vol. SC-

- 30, No.4, pp. 453-460, April 1995.
- [Bai95] R. T. Baird and Terri S. Fiez, "Linearity Enhancement of Multi-bit Delta-sigma A/D and D/A Converters Using Data Weighted Averaging", IEEE Trans. Circuits Syst.II, Vol. 42, pp.753-762, Dec.1995.
- [Sch95] R. Schreier and B. Zhang, "Noise-shaped Multi-bit D/A Converter Employing Unit Elements," Electron. Lett., Vol.31, No.20, pp.1712-1713, Sept.1995.
- [Gal97] Ian Galton, "Spectral Shaping of Circuit Errors in Digital-to-analog Converters" IEEE Trans. on Circuits and Systems II, Vol.44, pp. 808-817, Oct. 1997.
- [Hai94] A. Hairapetian and G.C. Temes, "A Dual-quantization Multi-bit Sigma-delta A/D Converter", IEEE Proc. ISCAS 94, Vol. 5, pp.437-440, May 1994.
- [Bra91] B. P. Brandt and B.A.Wooley, "A 50-MHz Multi-bit Sigma-delta Modulator For 12-62-MHz A/D Conversion," IEEE J. Solid-State Circuits, Vol.26, pp. 1746-1756, Dec. 1991.
- [Les90] T.C. Leslie and B. Singh, "An Improved Sigma-delta Modulator Architecture", IEEE Proc. ISCAS 89, Vol. 1, pp. 372-375, May 1990.
- [Lar88] L.E. Larson, T. Cataltepe, and G.C. Temes, "Multibit Oversampled Sigma-delta A/D Converter With Digital Error Correction", Electron letters, Vol. 24, pp. 1051-1052, August 1988.
- [Tho94] Charles D. Thompson, Salvador R. Bernadas, "A Digitally-Corrected 20b Delta-Sigma Modulator", ISSCC94, Session 11, Oversampling Data Conversion, Paper TP 11.5.
- [Opp75] A. V. Oppenheim and R.W. Schaffer, "Digital Signal Processing", Englewood Cliffs, Ch.5, 1975.
- [Wid85] Bernard Widrow, Samuel D. Stearns, "Adaptive Signal Processing", prentice-
-

hall publication, 1985.

- [Hay96] Simon Haykin, "Adaptive Filter Theory", 3rd edition, Prentice Hall Information and System Sciences Series, 1996.
 - [Ste81] S.D.Stearns, "Error Surfaces of Recursive Adaptive Filters", IEEE Trans. on Acoustics, Speech, and Signal processing, Vol. ASSP_29, no.3. pp. 763-766, June 1981.
 - [Shy89] John J. Shynk, "Adaptive IIR Filtering Using Parallel-form Realizations", IEEE Trans. on Acoustics, Speech, and Signal processing, Vol. 37, No.4, April 1989.
 - [Hon86] Hong Fan, W.Kenneth Jenkins, "A New Adaptive IIR Filter", IEEE Trans. on Circuits and Systems, Vol. cas-33, No.10, October 1986.
 - [Nan92] Ji-Nan Lin, Rolf Unbehauen, "Bias-Remedy Least Mean Square Equation Error Algorithm for IIR Parameter Recursive Estimation", IEEE Trans. on Signal Processing, Vol.40, No.1, January 1992.
 - [Ken89] J.B. Kenney and C.E.Rohrs, "Bias Analysis of A Combined Output Error-equation Error Algorithm", in Proc. 1989 Int. Conf. Acoust., Speech, Signal Processing, PP.2081-2084, May 1989.
 - [Ho96] K.C.Ho, Y.T.Chan, "Bias Removal in Equation-Error Adaptive IIR Filters", IEEE Trans. on Signal Processing, Vol.43, No.1, January 1996.
 - [Sou93] Soura Dasgupta, C. Richard Johnson Jr., and A. Mayalar Baksho, "Characterizing Persistent Excitation for The Sign-sign Equation Error Identifier". Automatica, Vol.29, No.6, pp. 1473-1489, 1993.
 - [Ste98] Stelian Mocanita, "High-Order Bandpass Sigma-delta Modulators", M. Eng. Thesis, Carleton University, 1998.
-
-

Appendix A: System Identification in C

```
/*Delta-sigma modulator modelling */

#include <stdio.h>
#include <math.h>
#include <stdlib.h>

#define Iteration 50000
#define SL      5.0      /* Saturation Level + & - */
#define QL      SL/4.0    /* Quantization Level */
#define Omega    2000     /* 2*Pi*frequency , rads/sec */
#define Amplitude 0      /* Input Sin-wave amplitude*/
#define A1      0.9      /* Leaky coefficient for first op-amp */
#define A2      0.9      /* Leaky coefficient for second op-amp */

main()
( FILE *outfileY, *outfileD;
  char dataout[50];
  float input,dco1,dco2,fb1,S1,S3;
  float dither,y,yf;
  int i;
  static float S2[Iteration],S4[Iteration];
  float random(),quantize();
  float DAC();

  printf("Enter output filename for Y:\n");
  scanf("%s", dataout);
  outfileY=fopen(dataout, "w");
  printf("Enter output filename for Dither:\n");
```

```
scanf("%s", dataout);
outfileD=fopen(dataout, "w");

/* Loop initialization */

dco1=0.04;      /* DC-offset of the first op-amp */
dco2=0.04;      /* DC-offset of the second op-amp */

dither= random();
fprintf(outfileD,"%f\n",dither);
S2[0]=0.0;
S3=dco2;
S4[0]=0.0;
y=quantize(S4[0]+dither);
fprintf(outfileY,"%f\n",y);
fb1=DAC(y);
S1=fb1+dco1;

dither= random();
fprintf(outfileD,"%f\n",dither);
S2[1]=S1;
S3=dco2+S2[1]+fb1;
S4[1]=dco2;
y=quantize(S4[1]+dither);
fprintf(outfileY,"%f\n",y);
fb1=DAC(y);
S1=fb1+dco1+Amplitude*sin(Omega*1);
```

```

/* Loop operation */

for(i=2;i<Iteration;i++)
{
    input=Amplitude*sin(Omega*i);

    S2[i]=S1-A1*S2[i-2];
    S4[i]=S3-A2*S4[i-2];
    dither=random();
    fprintf(outfileD,"%f\n",dither);
    y=quantize(S4[i]+dither);
    fprintf(outfileY,"%f\n", y);
    yf=y;
    S3=S2[i]+dco2+fb1;
    fb1=DAC(yf);
    S1=input+dco1+fb1;

}

fclose(outfileY);
fclose(outfileD);
}

float random()
/* generate random 1 or 0 */
{ float temp;
  double drand48();
  temp=drand48();
  if (temp>=0.5) return(1*QL);
  else return((-1)*QL);
}

```

```
}
```

```
float quantize(x)
```

```
float x;
```

```
{
```

```
    if (x>=3.5*QL) return(SL);
```

```
    else if (x>=2.5*QL) return(3*QL);
```

```
    else if (x>=1.5*QL) return(2*QL);
```

```
    else if (x>=0.5*QL) return(QL);
```

```
    else if (x>=(-0.5)*QL) return(0.0);
```

```
    else if (x>=(-1.5)*QL) return(-1*QL);
```

```
    else if (x>=(-2.5)*QL) return(-2*QL);
```

```
    else if (x>=(-3.5)*QL) return(-3*QL);
```

```
    else return(-SL);
```

```
}
```

```
float DAC(x)
```

```
/* feedback DAC */
```

```
float x;
```

```
/* {
```

```
    switch(x)
```

```
    { case -4: return((-3.5)*QL); break;
```

```
      case -3: return((-2.5)*QL); break;
```

```
      case -2: return((-1.5)*QL); break;
```

```
      case -1: return((-0.5)*QL); break;
```

```
      case 1: return((0.5)*QL); break;
```

```
      case 2: return((1.5)*QL); break;
```

```
      case 3: return((2.5)*QL); break;
```

```
      case 4: return((3.5)*QL); break;
```

```

    }
}  */

/* {
    if (x>=3.5*SL-0.01) return(SL);
    else if (x>=2.5*QL-0.01) return(3*QL);
    else if (x>=1.5*QL-0.01) return(2*QL);
    else if (x>=0.5*QL-0.01) return(QL);
    else if (x>=(-0.5)*QL-0.01) return(0.0);
    else if (x>=(-1.5)*QL-0.01) return(-1*QL);
    else if (x>=(-2.5)*QL-0.01) return(-2*QL);
    else if (x>=(-3.5)*QL-0.01) return(-3*QL);
    else return(-SL);
} */
{ return (x);}

/* System Identification Adaptive Algorithm */

#include <stdio.h>
#include <stdlib.h>

#define TRUE 1
#define FALSE 0

#define Iteration 50000 /* Total Iteration */
#define iteration 36000 /* Iteration# when to change gear*/

#define STEP1 0.003 /* Gear 1 */
#define STEP2 0.003 /* Gear 2, supposedly smaller than Gear 1 */

main()

```

```
{  
    FILE *infileY, *infileD, *outA0, *outA1, *outA2, *outB1, *outB2;  
    char datain[50];  
    int i, flag=FALSE;  
    float e, step;  
    static float inY[Iteration], inD[Iteration];  
    static float Y[Iteration], A0[Iteration];  
    static float A1[Iteration], A2[Iteration], B1[Iteration], B2[Iteration];  
  
    while(!flag)  
    {  
        printf("Enter input filename for y:\n");  
        scanf("%s", datain);  
        if((infileY=fopen(datain, "r")) == NULL)  
            printf("File does not exist.\n");  
        else  
            flag=TRUE;  
    }  
  
    flag=FALSE;  
    while(!flag)  
    {  
        printf("Enter input filename for Dither:\n");  
        scanf("%s", datain);  
        if((infileD=fopen(datain, "r")) == NULL)  
            printf("File does not exist.\n");  
        else  
            flag=TRUE;  
    }  
}
```

```
printf("Enter output filename for A0:\n");
scanf("%s", datain);
outA0=fopen(datain, "w");
printf("Enter output filename for A1:\n");
scanf("%s", datain);
outA1=fopen(datain, "w");
printf("Enter output filename for A2:\n");
scanf("%s", datain);
outA2=fopen(datain, "w");
printf("Enter output filename for B1:\n");
scanf("%s", datain);
outB1=fopen(datain, "w");
printf("Enter output filename for B2:\n");
scanf("%s", datain);
outB2=fopen(datain, "w");

for(i=0; i<Iteration;i++)      /* Read in data from files */
{
    fscanf(infileY,"%f",&inY[i]);
    fscanf(infileD,"%f",&inD[i]);
}
fclose(infileY);
fclose(infileD);

step=STEP1;                    /* Loop Initialization */
A0[0]=step*inY[0]*inD[0];
```

```

Y[1]=A0[0]*inD[1];
e=inY[1]-Y[1];
A0[1]=A0[0]+step*e*inD[1];

```

```

Y[2]=A0[1]*inD[2];
e=inY[2]-Y[2];
A0[2]=A0[1]+step*e*inD[2];
A1[2]=step*e*inD[0];

```

```

Y[3]=A0[2]*inD[3]+inD[1]*A1[2];
e=inY[3]-Y[3];
A0[3]=A0[2]+step*e*inD[3];
A1[3]=A1[2]+step*e*inD[1];

```

```

for(i=4;i<Iteration;i++)          /* Loop Operation */
{
    Y[i]=A0[i-1]*inD[i-0]+A1[i-1]*inD[i-2]+A2[i-1]*inD[i-4]+B1[i-1]*Y[i-2]+B2[i-1]*Y[i-
4];

    e=inY[i]-Y[i];
    A0[i]=A0[i-1]+step*e*inD[i-0];
    A1[i]=A1[i-1]+step*e*inD[i-2];
    A2[i]=A2[i-1]+step*e*inD[i-4];
    B1[i]=B1[i-1]+step*e*Y[i-2];
    B2[i]=B2[i-1]+step*e*Y[i-4];
    if (i>=iteration) step=STEP2;
}

```

```

for(i=0; i<Iteration;i++)

```

```
    { fprintf(outA0,"%f\n",A0[i]);  
      fprintf(outA1,"%f\n",A1[i]);  
      fprintf(outA2,"%f\n",A2[i]);  
      fprintf(outB1,"%f\n",B1[i]);  
      fprintf(outB2,"%f\n",B2[i]);  
    }  
    fclose(outA0);  
    fclose(outA1);  
    fclose(outA2);  
    fclose(outB1);  
    fclose(outB2);  
}
```

Appendix B: Digital Correction in C

```
#include <stdio.h>
#include <stdlib.h>

#define TRUE 1
#define FALSE 0
#define SIZE 819200/5
#define NO_BITS 131072 /* 18 bit accuracy */
#define RATE 50
#define u 0.1

main()
{
    FILE *infile,*outfileW,*outfileJ;
    char datain[50],outW[50];
    int i,j,k,flag=FALSE;
    float yre,inchar,old,tem;
    long mixre;
    long accum1re,accum2re,accum3re;
    long diff1re,diff2re,outre,tempre;
    static float pbranch[SIZE],nbranch[SIZE],temp[SIZE];
    static float W[SIZE],in[SIZE],tempp[SIZE],tempn[SIZE];
    /* static float J[SIZE];
    char outJ[50];
    */
    W[0]=-0.95;

    while(!flag)
    {
```

```
        printf("Enter input filename:\n");
        scanf("%s", datain);
        if((infile=fopen(datain, "r")) == NULL)
            printf("File does not exist.\n");
        else
            flag=TRUE;
    }

    fscanf(infile,"%f",&inchar);
    for(i=0;i<SIZE;i++)
    {    switch((long)inchar)
        { case 1:
            pbranch[i]=inchar;
            nbranch[i]=0.0;
            break;
        case -1:
            pbranch[i]=0.0;
            nbranch[i]=inchar;
            break;
        default :
            pbranch[i]=0.0;
            nbranch[i]=0.0;
        }
    }
    fscanf(infile,"%f",&inchar);
}

fclose(infile);
```

```
for(i=0;i<SIZE;i++)
{ inchar=pbranch[i];

switch(i%4)
{ case 0:
    temp[i]= inchar;
    break;
case 2:
    temp[i]=-1*inchar;
    break;
default :
    temp[i]=0.0;
}
}

accum1re=0;accum2re=0;accum3re=0;
outre=0;diff2re=0;diff1re=0;
tempre=0;

for(j=0;j<(SIZE/RATE);j++)
{
    for(i=0;i<RATE;i++)
    { yre=temp[RATE*j+i];
      accum1re +=(long)yre;
      if(accum1re > NO_BITS - 1)
          accum1re -= 2*NO_BITS;
      if(accum1re < -1*NO_BITS)
          accum1re += 2*NO_BITS;
      accum2re += accum1re;
      if(accum2re > NO_BITS - 1)
```

```
        accum2re -= 2*NO_BITS;
        if(accum2re < -1*NO_BITS)
            accum2re += 2*NO_BITS;
        accum3re += accum2re;
        if(accum3re > NO_BITS - 1)
            accum3re -= 2*NO_BITS;
        if(accum3re < -1*NO_BITS)
            accum3re += 2*NO_BITS;
    }

    outre = -1*diff2re;
    diff2re = -1*diff1re;
    diff1re = -1*tempre;
    tempre = accum3re;
    diff1re += accum3re;
    if(diff1re > NO_BITS - 1)
        diff1re -= 2*NO_BITS;
    if(diff1re < -1*NO_BITS)
        diff1re += 2*NO_BITS;
    diff2re += diff1re;
    if(diff2re > NO_BITS - 1)
        diff2re -= 2*NO_BITS;
    if(diff2re < -1*NO_BITS)
        diff2re += 2*NO_BITS;
    outre += diff2re;
    if(outre > NO_BITS - 1)
        outre -= 2*NO_BITS;
    if(outre < -1*NO_BITS)
        outre += 2*NO_BITS;
    for(k=0;k<50;k++)
```

```
{
    pbranch[RATE*j+k]=outre;
}
}

tem=0.0;
old=0.0;

for(i=0;i<SIZE;i++)
{ inchar=pbranch[i];
  tem=0.99*tem+inchar/(2.0*NO_BITS)-old;
  temp[i]=tem;
  old=inchar/(2.0*NO_BITS);
}

for(i=0;i<SIZE;i++)
{ inchar=nbranch[i];
  switch(i%4)
  { case 0:
      temp[i]= inchar;
      break;
    case 2:
      temp[i]=-1*inchar;
      break;
    default :
      temp[i]=0.0;
  }
}
```

```
accum1re=0;accum2re=0;accum3re=0;
outre=0;diff2re=0;diff1re=0;
tempre=0;

for(j=0;j<(SIZE/RATE);j++)
{
    for(i=0;i<RATE;i++)
    {
        yre=temp[RATE*j+i];
        accum1re +=(long)yre;
        if(accum1re > NO_BITS - 1)
            accum1re -= 2*NO_BITS;
        if(accum1re < -1*NO_BITS)
            accum1re += 2*NO_BITS;
        accum2re += accum1re;
        if(accum2re > NO_BITS - 1)
            accum2re -= 2*NO_BITS;
        if(accum2re < -1*NO_BITS)
            accum2re += 2*NO_BITS;
        accum3re += accum2re;
        if(accum3re > NO_BITS - 1)
            accum3re -= 2*NO_BITS;
        if(accum3re < -1*NO_BITS)
            accum3re += 2*NO_BITS;
    }

    outre = -1*diff2re;
    diff2re = -1*diff1re;
    diff1re = -1*tempre;
```

```

    tempre = accum3re;
    diff1re += accum3re;
    if(diff1re > NO_BITS - 1)
        diff1re -= 2*NO_BITS;
    if(diff1re < -1*NO_BITS)
        diff1re += 2*NO_BITS;
    diff2re += diff1re;
    if(diff2re > NO_BITS - 1)
        diff2re -= 2*NO_BITS;
    if(diff2re < -1*NO_BITS)
        diff2re += 2*NO_BITS;
    outre += diff2re;
    if(outre > NO_BITS - 1)
        outre -= 2*NO_BITS;
    if(outre < -1*NO_BITS)
        outre += 2*NO_BITS;
    for(k=0;k<50;k++)
    {
        nbranch[RATE*j+k]=outre;
    }
}

tem=0.0;
old=0.0;

for(i=0;i<SIZE;i++)
{ inchar=nbranch[i];
  tem=0.99*tem+inchar/(2.0*NO_BITS)-old;
  tempn[i]=tem;
  old=inchar/(2.0*NO_BITS);
}

```

```
    }

    printf("Enter output W filename:\n");
        scanf("%s", outW);
/*    printf("Enter output J filename:\n");
        scanf("%s", outJ);    */

/*    outfileJ=fopen(outJ, "w");        */
        outfileW=fopen(outW, "w");

    in[0]=tempp[0];
    for( i=0;i<SIZE;i++)
    { W[i+1]=W[i]+u*tempn[i]*in[i];
        in[i+1]=tempp[i+1]-tempn[i+1]*W[i+1];
/*    J[i]=in[i]*in[i];        */
        fprintf(outfileW,"%f\n",W[i]);
/*    fprintf(outfileJ,"%f\n",J[i]);        */
    }

    fclose(outfileW);
/*    fclose(outfileJ);        */
}
```