DIGITAL FILTER DESIGN, ERROR MODELING AND ERROR CORRECTION IN SIGMA-DELTA MODULATORS

by

Necmiye GENÇ

B.S., Microelectronics Engineering, Sabancı University, 2004

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electronical Engineering

Boğaziçi University

2007

ACKNOWLEDGEMENTS

I dedicate this work to my mother Muteber, my father Mustafa and my younger brother Necmi. If my family was not with me all through my life, I would not be accomplishing any of my goals. I wish to thank to them very much, for their endless support, love and infinite patience.

My advisor Prof. Günhan Dündar inspired me with his scientific insight, intelligence and perseverance. Assisted with the invaluable comments of him, I was able to successfully conclude this study.

I would like to thank to Çağatay Akçadoğan, expressing my deep appreciation for his suggestions and encouragement. He always kept me on track with his love and care.

ABSTRACT

DIGITAL FILTER DESIGN, ERROR MODELING AND ERROR CORRECTION IN SIGMA-DELTA MODULATORS

Modern electronic systems that are used in computers, automotive, communication, etc., are mostly mixed-signal systems. To implement mixed-signal systems, it is necessary to convert analog signals into the digital domain with the help of analog to digital (A/D) converters. Sigma Delta ($\Sigma\Delta$) Modulators also called *Oversampling Converters* have found widespread applications in many electronic systems because of their high speed, high resolution and low analog circuitry cost more than other A/D converters.

Even though $\Sigma\Delta$ conversion is intrinsically less sensitive to non-idealities than other A/D conversion techniques, non-idealities in $\Sigma\Delta$ building blocks can directly limit the overall performance of these converters. To increase the performance, the non-idealities such as finite amplifier gain, reference voltage errors, component mismatches, charge injection errors etc., can be reduced in the analog domain by careful analog circuit design or in the digital domain by noise-cancellation and reduction blocks.

In this thesis, it is aimed to perform the implementation of digital error correction and calibration algorithms for $\Sigma\Delta$ Converters in VHDL-AMS, an extension to VHDL standardized by IEEE. To achieve this, 1) Linear Least Mean Square (LMS) based digital filter, of which the coefficients are adaptively determined on-line to correct finite amplifier gain, component mismatch, reference voltage offset errors and 2) A novel dynamic element matching technique as a digital block that can be easily incorporated with any unit element DAC to correct DAC nonlinearity or reference voltage errors are implemented. The results of these error correction and calibration circuits are evaluated on 3-bit Second Order $\Sigma\Delta$ Converter and as a result significant SNR improvement is achieved.

ÖZET

SIGMA-DELTA MODÜLATÖRLERDE DİJİTAL FİLTRE TASARIMI, HATA MODELLEMESİ VE DÜZELTİLMESİ

Bilgisayarlarda, otomotiv sanayinde ve haberleşme sistemlerinde kullanılan modern elektronik devrelerin büyük bir kısmında analog ve sayısal sinyaller bir arada kullanılmaktadır. Analog ve sayısal sinyallerin bir arada olduğu bu karma sinyal sistemlerinin gerçeklenebilmesi için, analog-sayısal sinyal dönüştücüler yardımıyla, analog sinyallerin sayısal sinyallere dönüştürülmesi gerekmektedir. Fazla örnekleme çeviricileri olarak da adlandırılan Sigma Delta($\Sigma\Delta$) çeviricileri, diğer A/D çeviricilere göre yüksek hızları, yüksek çözünürlükleri ve düşük analog devre gereksinimleri sebebiyle birçok karma elektronik sistemde yaygın bir kullanım alanı bulmuşlardır.

$\Sigma\Delta$ analog-sayısal sinyal çevirme tekniği diğer çevirme tekniklerine gore ideal durumdan sapmalara daha az duyarlı olsa da, bu çeviricilerin doğasından kaynaklanan hatalar genel performanslarını doğrudan sınırlandırabilmektedirler. $\Sigma\Delta$ çeviricilerde sınırlı yükseltici kazancı, referans gerilim hataları, devre elemanı uyuşmazlıkları, yük enjeksiyon hataları gibi ideal durumdan sapmalara karşı analog düzlemde dikkatli bir devre tasarımı yapmak ya da sayısal düzlemde gürültü azaltma ve yok etme blokları kullanmak gibi önlemler alınabilmektedir.

Bu tezde amaçlanan sayısal düzlemde $\Sigma\Delta$ çeviriciler için geliştirilmiş gürültü azaltıcı ve hata düzeltici algoritmaların VHDL-AMS ortamında uygulanmasının yapılmasıdır. Bu çalışmada: 1) En küçük karesel ortalama hata kestirim yöntemine dayanan, katsayıları on-line olarak kendinden uyarlamalı bir sayısal filtrenin ve 2) Multi-bit çıkışlı $\Sigma\Delta$ çeviricilerde DAC'tan kaynaklı referans gerilim hatalarını düzeltmek için dinamik eleman eşleme tekniği temelli sayısal bir bloğun tasarımları ve uygulamaları yapılmıştır. Yapılan bu LMS temelli sayısal filtrenin ve DAC'daki hataları düzelten sayısal bloğun performans

değerlendirmeleri ise 3-bit çıkışlı ikinci dereceli bir $\Sigma\Delta$ analog-sayısal çevirici üzerinde yapılmış ve çıkış sinyal-gürültü seviyesinde dikkate değer düzelmeler elde edilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVATIONS

ΣΔ	Sigma Delta
Δ	Delta
ΣΔM	Sigma Delta Modulator/Modulation
σ_ε	Resistor Standard Deviation

A/D	Analog/Digital
VHDL	Very High Speed Integrated Circuits Hardware Description Language
AMS	Analog Mixed Signal
LMS	Least Mean Square
DAC	Digital Analog Converter
SNR	Signal Noise Ratio
SQNR	Signal-Quantization Noise Ratio
ENOB	Effective number of bits
ADC	Analog Digital Converter
STF	Signal Transfer Function
NTF	Noise Transfer Function
OSR	Oversampling Ratio
D/A	Digital/Analog
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
CIC	Cascaded Integrator-Comb
LSB	Least Significant Bit
SR	Slew Rate
OL	Open Loop
BW	Bandwidth
DC	Direct Current
SC	Switched Capacitor

MSE	Mean Square Error
ADF	Adaptive Digital Filter
DEM	Dynamic Element Matching
AC	Alternating current
INL	Integral Nonlinearity
DNL	Differential Nonlinearity
ILA	Individual Level Averaging
DWA	Data Weighting Averaging

1. INTRODUCTION

Modern electronic circuits are mainly mixed-signal circuits. This situation necessitates the conversion of signals from A/D and D/A. Throughout the last two decades, various A/D and D/A conversion techniques have been explored and implemented. Regarding their conversion speeds and used bandwidths, these data converters can be classified as Nyquist-rate converters and oversampling converters. Oversampling converters have many advantages such as lower circuit complexity, insensitivity to analog circuit imperfections, simpler anti-aliasing filters and quantizer requirements. Section 2 includes the theoretical background for A/D Converters and oversampled ΣΔ Modulators.

Sigma Delta (ΣΔ) A/D converters, introduced by Inose in 1962, are examples of oversampling converters. They operate with redundant temporal data; in other words, they make use of the oversampling concept. The output of ΣΔ modulator is a coarse quantization of the analog input and these converters require digital filter circuitry to remove out-of-band quantization noise and to achieve sample rate reduction. In Section 3, VHDL implementation of low pass finite impulse response (FIR) and decimation filters for ΣΔ converters is discussed.

ΣΔ Conversion is intrinsically less sensitive to non-idealities than other A/D conversion techniques [1]; inherent non-idealities in ΣΔ building blocks limit the overall performance of these converters. This makes their design and optimization a difficult task and simulations play an important role in designing ΣΔ Converters. High-level languages such as C, Matlab have been extensively used for simulation purposes. VHDL-AMS supports the hierarchical description and simulation of discrete, continuous and mixed systems with conservative and non-conservative semantics. Section 4 involves behavioral and structural modeling of ΣΔ Converters in VHDL-AMS. This section also includes the modeling non-idealities of ΣΔ Converters.

Pipelined ΣΔ Converters usually suffer from both dynamic and static errors [2]. Finite bandwidth and slew rate of residue amplifiers introduce memory. Capacitor mismatch, finite op-amp gain and non-linearity, switch-induced charge injection, as well as

various sources of offset result static errors. The static errors highly distort the overall transfer function, resolution and linearity of ΣΔ Modulators. (In Section 4, the effects of these static non-ideality errors are also mentioned). Background self calibration techniques have been used to overcome static errors in ΣΔ Modulators without stopping input conversion. LMS based adaptive filters are mostly used for static error correction and calibration of ΣΔ Modulators. In Section 5, a general overview about adaptive filters and LMS filters is given. Section 6 includes a detailed description of LMS Based Adaptive Error Correction Filter and performance analysis of LMS filter for each static error on 1-bit second order ΣΔ Modulator.

Modulators with more than two quantization levels are called multi-bit modulators and multi-bit ΣΔ Modulators are sensitive to the feedback D/A converter (DAC) non-idealities. High DAC linearity requires precise matching of the DAC unit elements that are typically capacitors and current sources. To overcome this trade-off two signal processing strategies are used: 1) Dynamic Element Matching (DEM). 2) Background Calibration/Correction Schemes. Compared to DEM techniques background calibration schemes are more expensive in terms of design complexity, hardware requirements and power consumption [3]. In Section 7, 3 different DEM methods: 1) Randomization, 2) First Order DEM, 3) Second Order DEM are explained. Conclusion and Future Work part, Section 8, states the overall performance of LMS based adaptive filter and Second Order DEM method on 3-bit second order ΣΔ Modulator.

2. THEORETICAL BACKGROUND

2.1. Analog to Digital Converters

2.1.1. Sampling Process

Analog to digital conversion of a signal is performed in terms of two separate operations: Uniform sampling in time domain, and quantization in amplitude. Sampling process transforms a continuous-time signal $x(t)$ into a discrete time signal $x[n] = x(nT_s)$. The effect of the sampling process is to create periodically repeated versions of the signal spectrum at multiples of the sampling frequency $f_s = 1/T_s$ [4]. This relationship is expressed in 2.1, where $X_s(f)$ represents the spectrum of the sampled signal, and $X(f)$ is the spectrum of the original continuous time signal.

$$X_s(f) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(f - kf_s) \tag{2.1}$$

The ratio of the sampling rate f_s to the signal bandwidth differentiates two types of converte rs: Nyquist-rate and oversampling data converters. In Nyquist-rate data converters, the sampling frequency is twice the analog signal maximum frequency, $f_s = 2f_{\max}$ to allow accurate reproduction of the original signal [4]. Oversampling converters sample the analog signal at a greater rate than that of the Nyquist. Oversampling data converters have several advantages over Nyquist-rate converters in terms of the relaxation of the anti-alias filter requirements and quantization noise shaping.

2.1.2. Quantization Noise

Once sampled, the signal samples must be quantized in amplitude to a finite set of output values. Quantization is a non-invertible process, since an infinite number of input amplitude values are mapped to a finite number of output amplitude values. Quantization inevitaply introduces an error. Figure 2.1. shows the transfer function of a quantizer and the relationship between the quantization noise and the input signal.

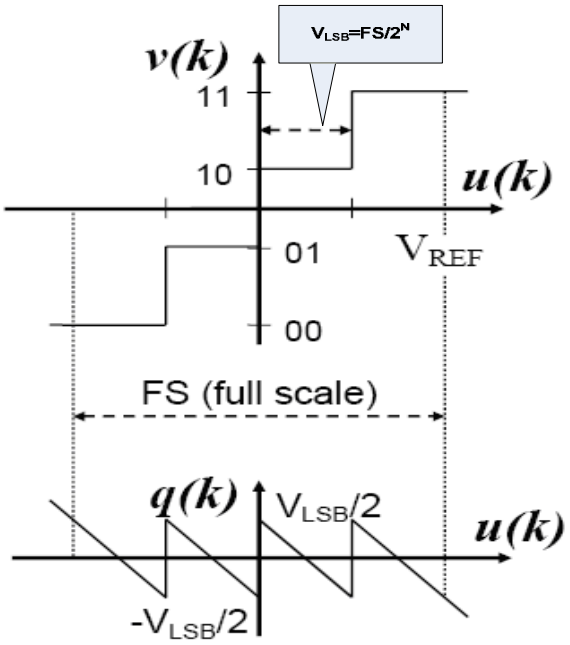

Figure 2.1. Quantization Noise

The quantization noise $q(k)$ is evenly distributed between $-V_{LSB}/2$ and $V_{LSB}/2$ as long as the sampled values are evenly distributed between $-V_{REF}$ and V_{REF}. V_{LSB} is the quantization step and V_{REF} is the reference voltage. The quantization noise $q(k)$ has a power of

$$\sigma^2(q) = \frac{1}{V_{LSB}} \int_{-VLSB/2}^{VLSB/2} q^2 dq = \frac{V_{LSB}{}^2}{12} \tag{2.2}$$

When a quantized signal is sampled at $f_s = 1/T_s$, all the power folds into the band $0 \le f \le f_s/2$. If the quantization noise is white, the power spectral density of the sampled noise is given by

$$S_E(f) = \frac{\sigma^2(q)}{fs/2} = \frac{{V_{LSB}}^2}{12 f_s/2} \tag{2.3}$$

If the dither is sufficiently large and busy to decorrelate the quantization error, the noise power that falls into the signal band f_b is given by

$$P_Q = \int_0^{f_b} S_E(f)df = \frac{{V_{LSB}}^2}{12} \tag{2.4}$$

The quantization noise affects the quality of the signal. This is expressed as the signal-to-noise ratio SNR or often called signal-to-quantization noise ratio $SQNR$. For a sine wave input with amplitude variation $-2^{N-1}V_{LSB}$ to $2^{N-1}V_{LSB}$, where N is quantizer number of bits.

$$SNR = \frac{2^{2N-3}{V_{LSB}}^2}{{V_{LSB}}^2/12} \cong 10\log\left(\frac{3.2^{'N}}{2}\right) = 6.02N + 1.76dB \tag{2.5}$$

The SNR determines the effective number of bits $(ENOB)$ that the converter can resolve, given by

$$ENOB = \frac{SNR_{dB} - 1.76}{6.02} \tag{2.6}$$

2.1.3. Nyquist –Rate A/D Converters

A Nyquist-rate A/D converter generates a series of digital output levels that have a one-to-one correspondence to a particular input value. The sampling frequency is twice the maximum input frequency $f_s = 2 f_{\max}$. Nyquist-rate converters are classified as the number of clock cycles required to complete a single conversion like flash, sub-ranging, successive approximation and pipelined. Nyquist-rate converters have two significant limitations: the anti-alias filter and quantization noise.

If out-of-band interferers are very strong and very near the pass frequency of the actual input signal, the requirements for filter stop-band and narrowness of the transition band will be quite important. Topologies satisfying these requirements are high-order filters like Chebychev and introduce phase distortion at the edge of the passband. In addition to this, quantization noise is another limitation for Nyquist-rate A/D converters. They have to use the minimal required sampling frequency, that allows a maximal quantization error as given by (2.4) and this error decreases the signal quality as mentioned in (2.5). Also in real applications, quantizers used with high resolution Nyquist-rate A/D converters are in order of micro-volts. It is difficult to implement them since they are prone to circuit non-idealities.

2.1.4. Oversampled A/D Converters

Oversampled data converters have several advantages over Nyquist-rate converters including the anti-alias filter requirements and quantization noise [1]. These advantages are achieved by having a sampling frequency much larger than the Nyquist-rate converters. However, an extra decimation filter is needed at the output to filter and downsample the digital stream as seen in Figure 2.2.

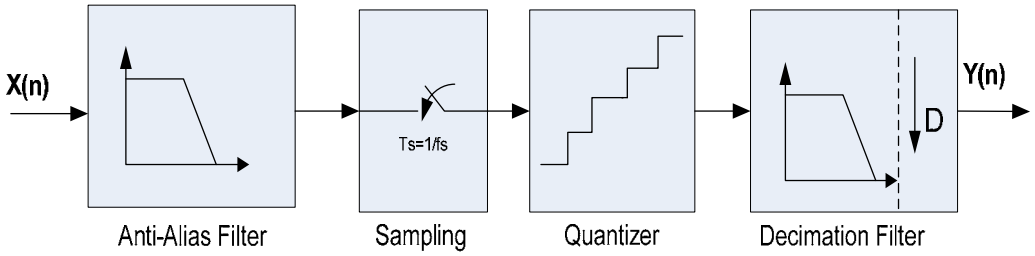

Figure 2.2. Conventional Oversampled A/D converters

In oversampled converters, quantization noise can be reduced or shaped by using a feedback loop which is implemented by Δ modulation and $\Sigma\Delta$ modulation respectively.

2.2. $\Sigma\Delta$ Modulators

A conventional $\Sigma\Delta$ A/D converter is shown in Figure 2.3 Oversampling converters reduce the in-band quantization noise by spreading it uniformly across the frequency band. Moreover, oversampling $\Sigma\Delta$ ADCs reduce the quantization noise power inside the signal band by pushing most of the in-band noise outside the signal band. The output of the $\Sigma\Delta$ Modulator is then low-pass filtered and finally down sampled to or near to the Nyquist rate like conventional oversampled converters as shown in Figure 2.3.

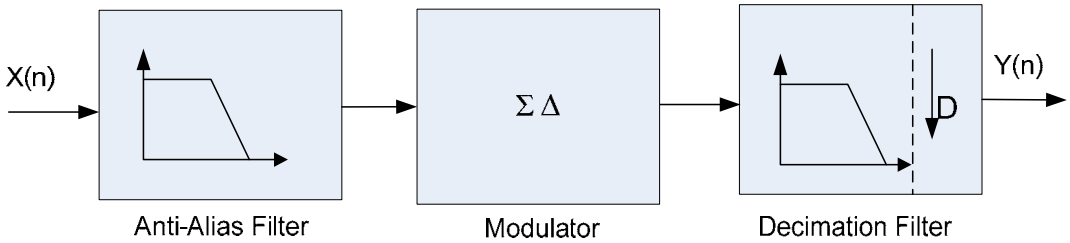

Figure 2.3. Conventional $\Sigma\Delta$ Modulators

The quantization noise spreading and shaping characteristics are present in various types of $\Sigma\Delta$Ms. The noise shaping properties of the modulator define its order and properties. The following sections introduce the fundamental $\Sigma\Delta$ Modulator topologies and their base analysis.

2.2.1. First-Order ΣΔ Modulators

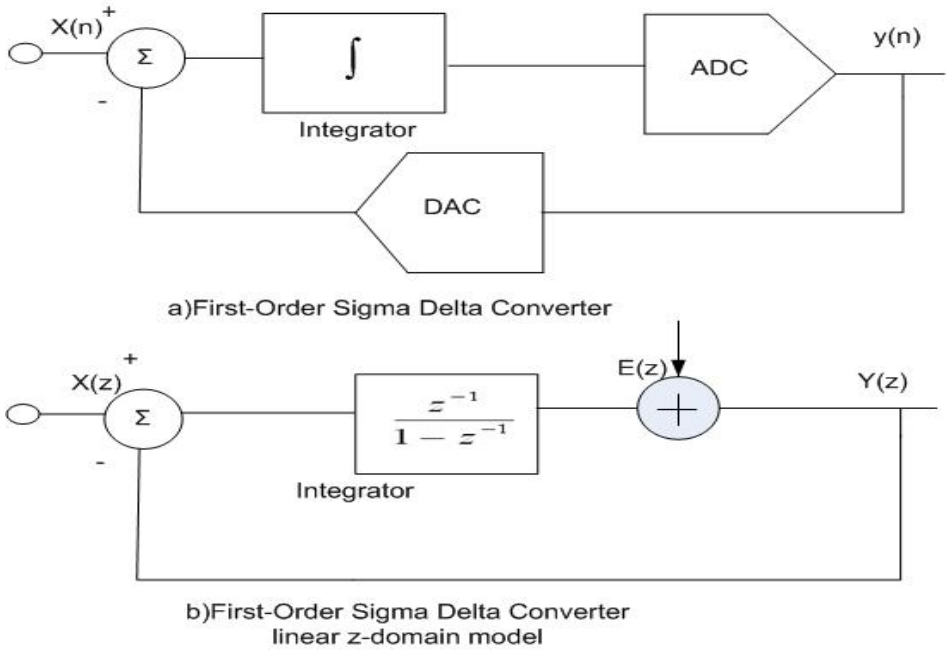

Figure 2.4. First Order ΣΔ block and linear z-domain model

Figure 2.4. shows the basic block diagram of a ΣΔ modulator. The main components are: a summing node at the input, an integrator, an ADC and a DAC inside the feedback loop. Its output $y(n)$ is subtracted from its input signal, x, which is sampled at a rate much larger than Nyquist rate. The result after passing through a discrete-time filter which is the integrator in this case serves as an input to the quantizer. If the gain of the filter is high in the interval of the frequency of interest, the quantization error is attenuated in signal band due to the feedback loop.

The first-order ΣΔ modulator linear model is shown in Figure 2.4(b). The output $Y(z)$ can be represented by

$$Y(z) = z^{-1}X(z) + (1 - z^{-1})E(z) \tag{2.7}$$

where $X(z)$ and $E(z)$ are the z-transform of the input signal and the quantization noise respectively. $STF(z)$ and $NTF(z)$ are the respective transfer function of the input signal and quantization error. From (2.7) we can obtain $STF(z)$ and $NTF(z)$

$$STF(z) = z^{-1} \tag{2.8}$$

$$NTF(z) = 1 - z^{-1} \tag{2.9}$$

Assuming that the quantization noise is uncorrelated and uniformly spread in the signal band, the power spectral density (PSD) of the modulation noise is

$$S_Q(f) = S_E(f)(1 - \exp^{-j2\pi fTs})^2 = 4S_E(f)\sin^2(\frac{2\pi f}{2f_s}) \tag{2.10}$$

The total noise power in the signal band is

$$P_Q = \int_0^{fb} S_Q(f)df \approx \frac{V_{LSB}{}^2}{12}\frac{\pi^2}{3OSR^3}, f_s^2 >> f_b^2 \tag{2.11}$$

As it is seen from 2.11 each doubling of OSR (Over Sampling Ratio) reduces the noise by 9dB and provides 1.5 bits of more resolution. However, a digital filter is required at the output of the modulator, to decimate the modulated signal to Nyquist rate. Otherwise, the noise pushed toward high-frequency range will degrade the resolution when it is sampled at the Nyquist rate [1].

2.2.2. Second-Order ΣΔ Modulators

Including one more integrator in the modulator loop increases the noise transfer function order to two. The resulting architecture is the single-loop second-order ΣΔ modulator. (Figure 2.5.)

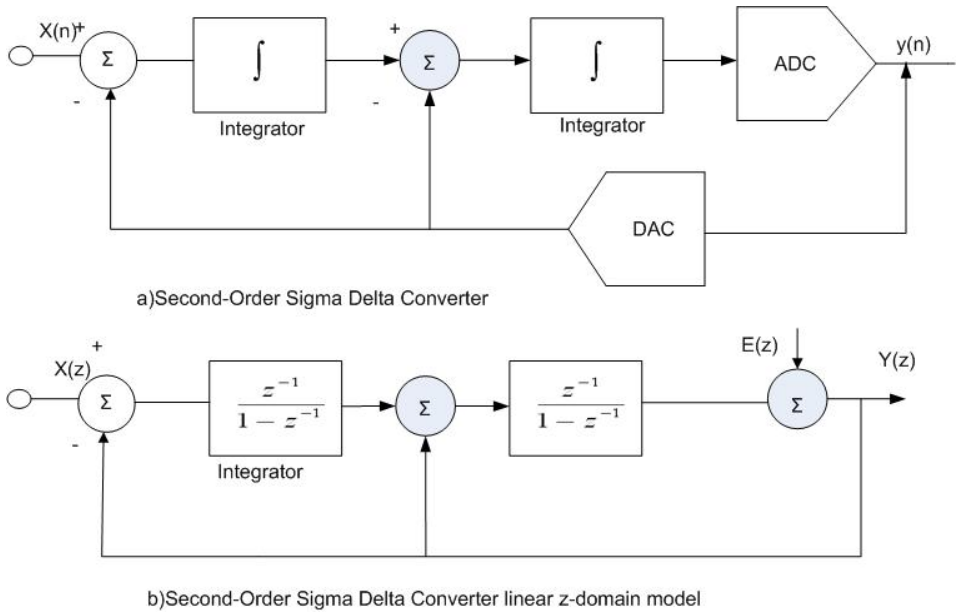

Figure 2.5. Second Order ΣΔ Modulator and linear z-domain model

The linear model of second-order ΣΔ Modulator is given by

$$Y(z) = z^{-2}X(z) + (1 - z^{-1})^2 E(z) \tag{2.12}$$

Consequently, the power spectral density of the quantization noise is as follows:

$$S_Q(f) = S_E(f)\left|1 - \exp\left(-j2\pi\frac{f}{fs}\right)\right|^4 = S_E(f)\cdot 16\sin 4(\pi\frac{f}{f_s}) \tag{2.13}$$

Moreover, for busy signals the noise is given by

$$P_Q = \int_0^{fb} S_Q(f)df \approx \frac{V_{LSB}{}^2}{12}\frac{\pi^4}{5OSR^5}, \quad f_s{}^2 >> f_b{}^2 \tag{2.14}$$

For this architecture the noise falls by 15dB for each doubling of the *OSR*, giving 2.5 extra bits of resolution [1].

2.2.3. Higher-Order ΣΔ Modulators

Higher order modulators are classified as single-loop higher order modulators and cascade modulators. Higher order modulators are obtained by adding more feedback loops to the system. In general, an *L*-order modulator contains *L* feedback loops. An alternative to the single-loop or interpolative modulators are the cascade architectures like multi-stage or MASH. Their functioning is based on the cascade connection of low-order modulators (0, 1 or 2), whose stapility is guaranteed by design [5], [6], [7].

2.2.4. Multi-bit Quantization ΣΔ Modulators

Another way for increasing the effective resolution of ΣΔ modulators is to increase the number of levels of the internal quantization [1]. These converters have significant advantages: Large SNR for lower OSR, better stability and smaller noise patterns. Figure 2.6. below exhibits that increasing the bit number used in quantization, enhances SNR of ΣΔ Modulator. However, the linearity of the multi-bit ΣΔ modulator is limited by the accuracy of D/A converter and they require more complex digital and analog circuitry. In Section 4.3.5 Multi-bit Quantization Non-Linearity modeling will be mentioned and in Section 7 linearity Enhancement of Multi-bit DAC will be explained in detail.

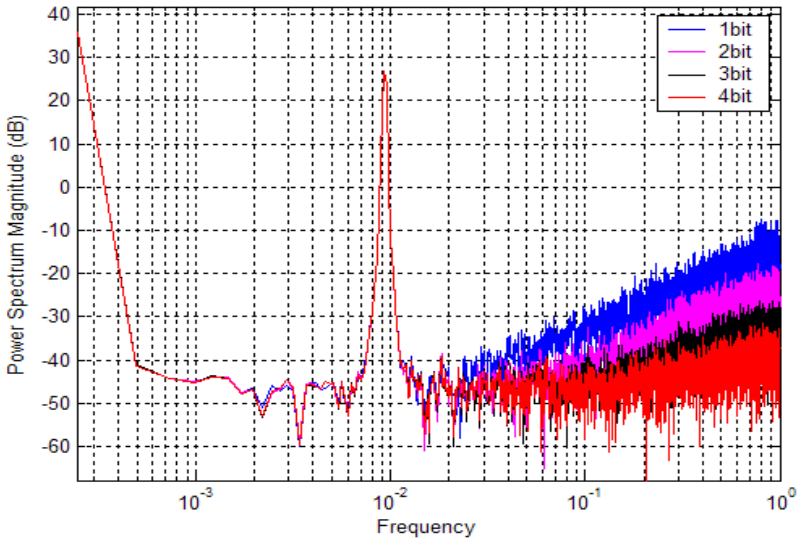

Figure 2.6. PSD comparison of different multi-bit ΣΔ Modulators

3. DIGITAL FIR DECIMATION FILTER DESIGN

The output of the ΣΔ Modulator is a stream of bits, 0s and 1s, synchronous with the sampling clock. This data stream contains all the data required about the analog input to the ADC, but it also contains a lot of high frequency noise. A digital low-pass filter FIR removes those high-frequency components, and a data decimator removes the oversampled data.

3.1. Conventional FIR Decimation Filter

Finite Impulse Response filters (FIR) are commonly preferable to infinite impulse response filters (IIR) as a digital low-pass filter for ΣΔ Converters, because they are inherently stable. They require no feedback and can have linear phase [4].

FIR filters can be expressed as a difference equation which defines how the input signal is related to the output signal.

$$y[n] = b_0 x[n] + b_1 x[n-1] + \ldots\ldots\ldots\ldots + b_p x[n-P] \tag{3.1}$$

$$y[n] = \sum_{i=0}^{P} b_i x[n-i] \tag{3.2}$$

where P is the filter order, $x[n]$ is the input signal, $y[n]$ is the output signal and b_i are the filter coefficients.

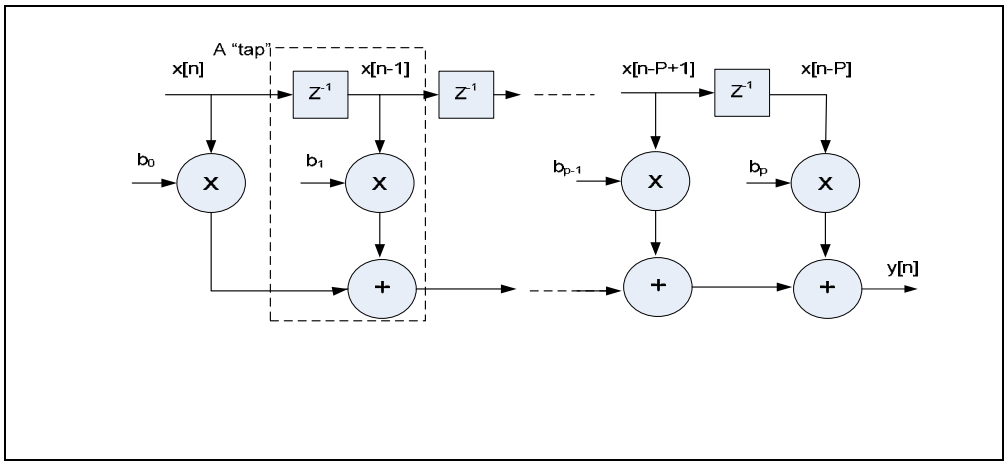

Figure 3.1. Finite Impulse Response Filter (FIR)

In the most straightforward implementation of Figure 3.1, the low-pass FIR filter computes an output sample at each value of *n,* but then only one of every M output points is retained. Intuitively, we might expect that it should be possible to obtain a more efficient implementation which does not compute the samples that are thrown away. To obtain a more efficient implementation, a polyphase decomposition of the filter is used for both decimation and low-pass filtering [7].

3.2. Polyphase Decimation Filters

The polyphase decomposition of a sequence is obtained by representing it as a superposition of M sequences, each consisting of every *M*th value of successively delayed versions of the sequence. When this decomposition is applied to a filter impulse response, it can lead to efficient implementation structures for linear filters in several contexts. Specially, consider an impulse response $h[n]$ that we decompose into M subsequences $h_k[n]$ as follows:

$$h_k[n]=\begin{cases} h[n+k] & \text{n = integer multiple of M} \\ 0 & \text{otherwise} \end{cases} . \tag{3.3}$$

By successively delaying these subsequences, we can construct the original impulse response $h[n]$;

$$h[n] = \sum_{k=0}^{M-1} h_k[n-k] \tag{3.4}$$

The sequences $e_k[n]$ are

$$e_k[n] = h[nM + k] = h_k[nM] \tag{3.5}$$

It is referred to in general as the polyphase components of $h[n]$. In frequency or z-transform domain, the polyphase representation corresponds to expressing $H(z)$ as

$$H(z) = \sum_{k=0}^{M-1} E_k(z^m) z^{-k} \tag{3.6}$$

Equation (3.6) expresses the system $H(z)$ as a sum of delayed polyphase component filters.

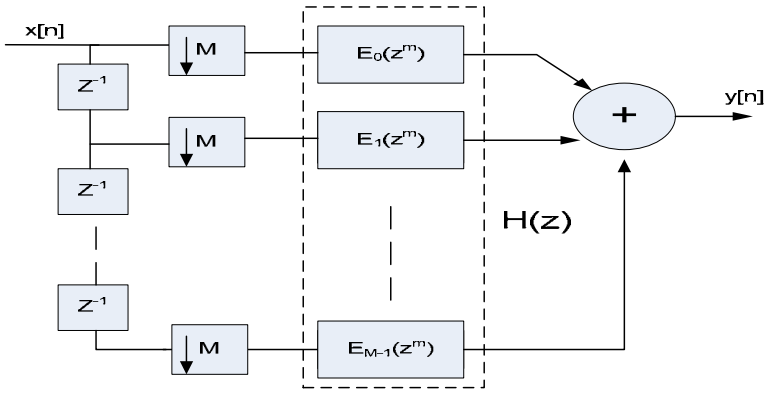

Figure 3.2. Polyphase decomposition Decimation Filtering.

The input $x[n]$ is clocked at a rate of 1 sample per unit time and $H(z)$ is a P-point FIR filters. In the straightforward implementation of Figure 3.1., we require P multiplications and (P-1) additions per unit time. In the system of Figure 3.2., each of the filters $E_k(z)$ is

of length P/M, and their inputs are clocked at a rate of 1 per M units of time. Consequently, each filter requires $\frac{1}{M}\left(\frac{P}{M}-1\right)$ additions per unit time, and the entire system than requires *(P/M)* multiplications and $\left(\frac{P}{M}-1\right)+(M-1)$ additions per unit time. Thus, we can achieve a significant savings for some values M and P.

3.2.1. Design and Implementation

It is aimed to design a generic polyphase FIR decimation filter for ΣΔMs that can easily be used later. To achieve this, filter input and output vector dimensions; number of taps and phase are defined as generic type in VHDL.

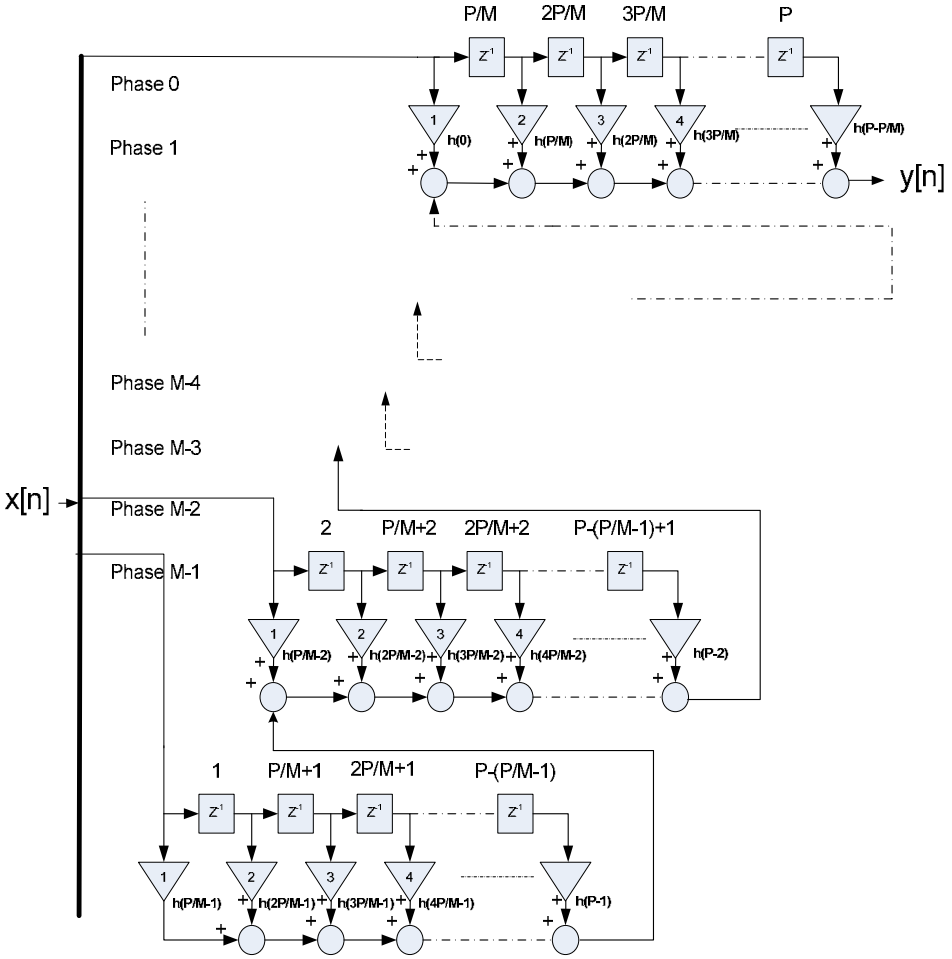

Figure 3.3. M Phase P order Polyphase FIR Decimation Filter

Figure 3.3. describes M Phase P order Polyphase FIR Decimation Filter signal flow diagram. The order of the states that corresponds to filter taps appears above each delay element. Filter coefficients (from h (0) to h (P-1)) for each tap are indicated with triangles. According to the filter flow below, state 1 applies to the first delay element in phase M-1; state 2 applies to the first delay element in phase M-2; state 3 applies to the first delay element in phase M-3 and so on to have the output y [n]. For decimation, there is a commutator that operates counterclockwise at the input of the filter. It moves from position 0 to position M-1, position 1, and back to position M-2, position 3, and back to position M-3 and so on as input samples enter the filter. The implementation cost of this topology is P+1 multipliers, P adders and P states. To decrease the complexity of FIR filter CSD or Factored-CSD type multipliers and tree type adders can be used.

Design of the polyphase FIR decimation filter is performed with the help of Matlab Filter Design and Analysis Tool (FDATool). FDATool enables the user to design digital FIR or IIR filters by setting filter specifications, by importing filters from your MATLAB workspace, or by adding, moving or deleting poles and zeros. FDATool also provides tools for analyzing filters, such as magnitude and phase response and pole-zero plots.

In our application, the sampling frequency of our ΣΔM is 4MHz with OSR 32 and we need a polyphase FIR decimation filter with decimation factor of 32. To have a better magnitude response, it is decided to use 64 taps which is a multiple of phase number 32. Frequency specifications are the following for our application: Fpass =0.025 MHz and fstop is 0.0625 MHz. The coefficients satisfying these decimation and frequency requirements are calculated with the help of FDATool and applied to each corresponding tap. As it is mentioned before, this FIR polyphase decimation filter is aimed to be generic and input/output bit numbers, filter phase and order numbers, tap coefficients are defined as generic type in VHDL code. In Figure 3.4 magnitude response of 32-Phase 64-Order polyphase FIR decimation filter can be seen.

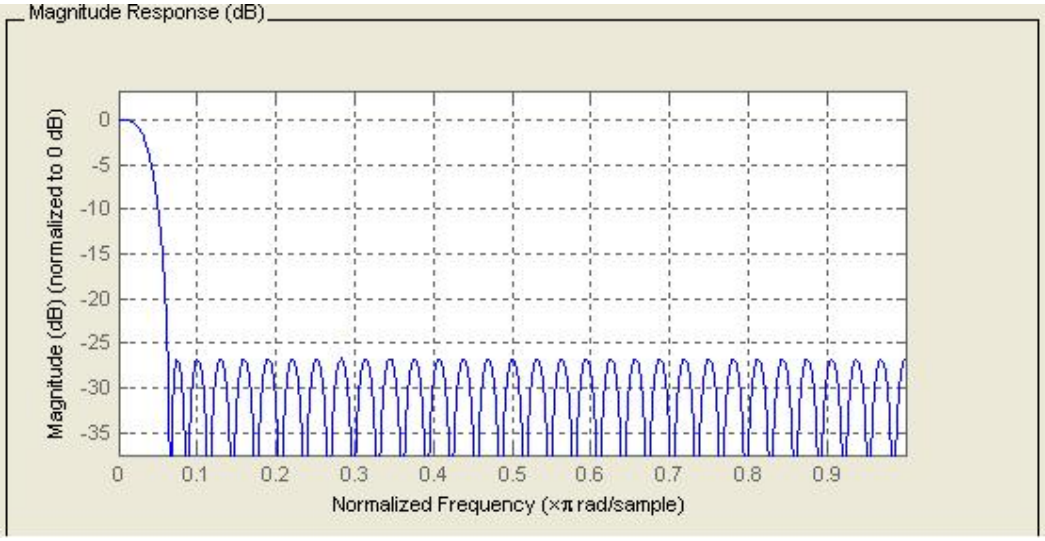

Figure 3.4. Magnitude Response of Polyphase FIR Decimation Filter

3.3. Cascaded Integrator-Comb Decimator

CIC is a class of linear phase FIR filters for decimation and interpolation. These filters require no multipliers and use limited storage thereby leading to more economical hardware implementation. They are designated as cascaded integrator-comb (CIC) filters because their structure consists of an integrator section operating at the high sampling rate and a comb section operating at the low sampling rate. Using CIC filters, the amount of passband aliasing or imaging error can be brought within prescribed bounds by increasing the number of stages in the filter [8]. However, the width of the pass-band and the frequency characteristics outside the pass-band are severely limited. For critical applications these limitations can be overcome by using CIC filters to make the transition between high and low sampling rate, and to use conventional filters at the low sampling rate to "shape" or "clean up" the frequency response [8]. In this manner, CIC filters are used at high sampling rates where economy is critical, and conventional filters are used at low sampling rates where the number of multipliers per second is low [8].

Figure 3.6 shows the basic structure of the CIC decimation filter. The integrator section of CIC filters consists of N ideal digital integrator stages operating at the high

sampling rate, f_s. Each stage is implemented as a one-pole filter with a unity feedback coefficient. The system function for a single integrator is

$$H_I(z) = \frac{1}{1 - z^{-1}} \tag{3.6}$$

The comb section operates at the low sampling rate fs/R where R is the integer rate change factor. This section consists of N comb stages with a differential delay of M samples per stage. The differential delay is a filter design parameter used to control the filter's frequency response. In practice, the differential delay is usually held to M=1 or 2. The system function for a single comb stage referenced to high sampling rate is

$$H_C(z) = 1 - z^{-RM} \tag{3.7}$$

There is a rate change switch between the two filter sections. For decimation, the switch subsamples the output of the last integrator stage, reducing the sampling rate from fs to fs/R; and for interpolation, the switch causes a rate increase by a factor of R by inserting R-1 zero valued samples between consecutive samples of the comb section output. It follows from (3.7) and (3.8) that the system function for the composite CIC filter referenced to the high sampling rate, fs, is

$$H(z) = H_I{}^N(z) H_C{}^N(z) = \frac{(1 - z^{-RM})^N}{(1 - z^{-1})^N} = \left[\sum_{k=0}^{RM-1} z^{-k}\right]^N \tag{3.8}$$

It is implicit from the last form of the system function that the CIC filter is functionally equivalent to a cascade of N uniform FIR filter stages. A conventional implementation consists of a cascade of N stages each requiring RM storage registers and one accumulator. Taking advantage of the rate change factor, one of the N stages can be simplified to use only M storage registers.

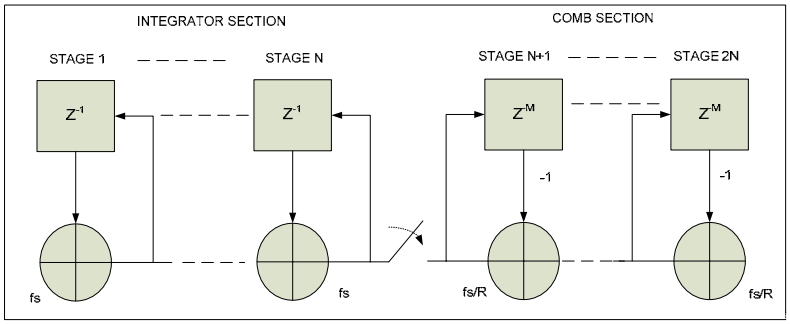

Figure 3.5. CIC Decimation Filter [11]

3.3.1. Design and Implementation

This section presents design considerations for CIC decimation filters. We wish to design a decimation filter to reduce the sampling rate from 4 MHz to 62.5 kHz. The most important issue in designing CIC Decimation Filter is to decide number of differential delay M and sections N.

Table 3.1. Passband Attenuation for Large Rate Change Factor [11]

Relative Bandwidth-Differential Delay Product (Mf_c)	Passband Atteneation at f_c (dB) As a Function of Number of Stages (N)					
	1	2	3	4	5	6
1/128	0.00	0.00	0.00	0.00	0.01	0.01
1/64	0.00	0.01	0.01	0.02	0.02	0.02
1/32	0.01	0.03	0.04	0.07	0.08	0.08
1/16	0.06	0.11	0.17	0.28	0.34	0.34
1/8	0.22	0.45	0.67	1.12	1.35	1.35
¼	0.91	1.82	2.74	4.56	5.47	5.47

Table 3.1. describes passband attenuation for large rate change factors. According to this table, if relative bandwidth-differential delay product (Mf_c) is 1/32, passband

attenuations are: 0.01 for 1 Stage, 0.03 for 2 Stages, 0.04 for 3 Stages, 0.06 for 4 Stages, 0.07 for 5 Stages and 0.08 for 6 Stages.

Another issue is deciding filter input and output word lengths. The most significant bit (MSB) of these filters is determined as function if the overall register growth [11]. The number of bits in the input data stream is B_{in} the rate of change factor R and M is differential delay. B_{max} the most significant bit at the filter output is given by:

$$B_{\max} = (N \log_2 RM + B_{in} - 1) \tag{3.9}$$

B_{max} is large for many practical cases and can cause in large register widths and methods of truncation and rounding may be used at each stage reducing register widths significantly [8].

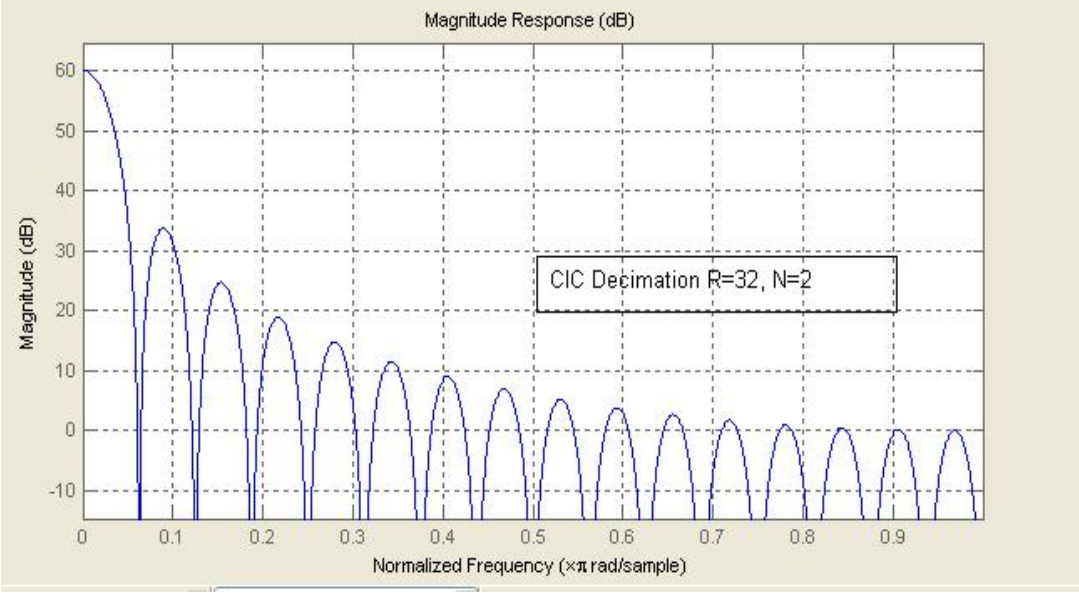

Figure 3.6. CIC Decimation Filter for R=32, N=2.

In our application, the design parameters are decided as the following: differential delay M=1, number of sections N=4, the most significant bit at output is B_{max} is 34 with input word length 12. It is aimed to design CIC Decimation filter as generic as possible to implement into further applications just like polyphase FIR decimation and input word lengths, integrator and comb section numbers are defined as generic type in VHDL code. On the other hand, output word length is assigned to input word length as in (3.10). The

magnitude response of our CIC decimation (N=4, R=32) can be seen in Figure 3.7. In addition to this, passband attenuation for different number of stages can be seen in Figure 3.6. and 3.8. to have an idea.

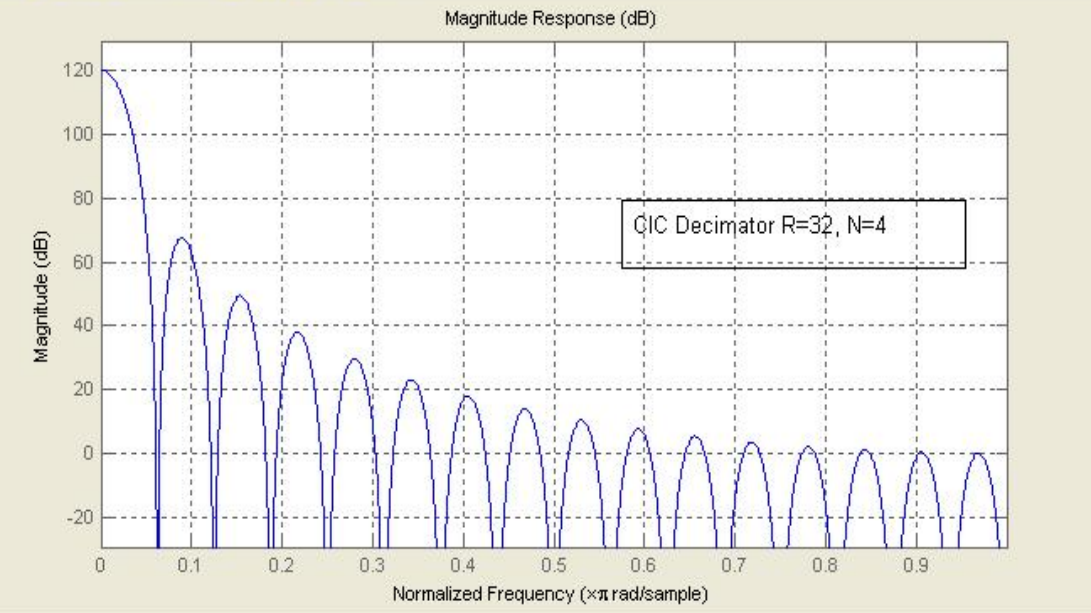

Figure 3.7. CIC Decimation Filter for R=32, N=4.

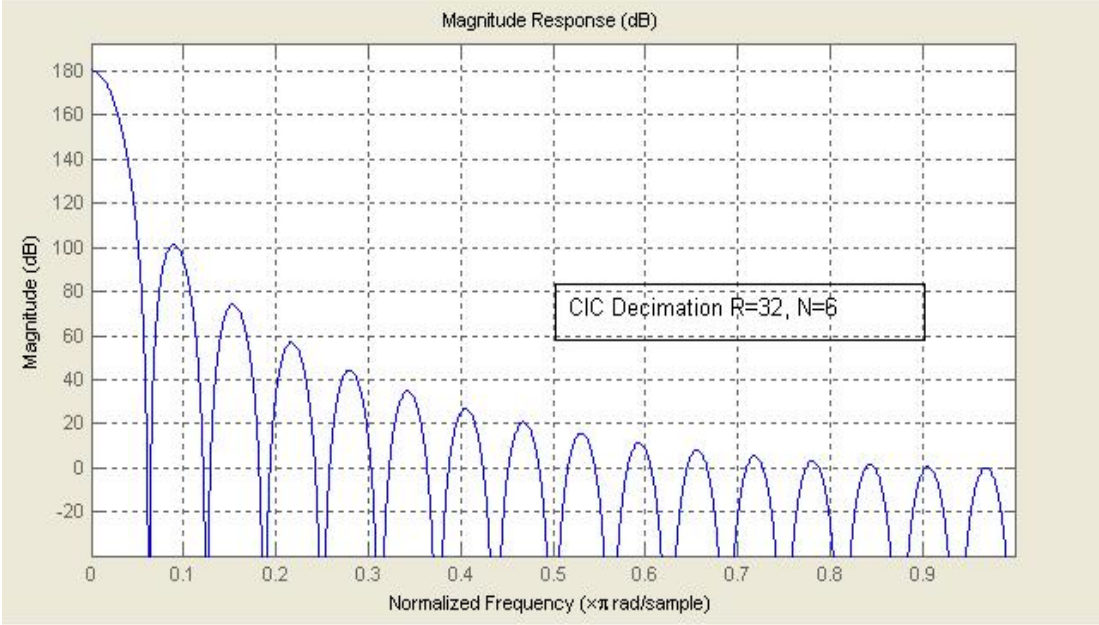

Figure 3.8. CIC Decimation Filter for R=32, N=6.

4. MODELING ΣΔ MODULATOR

4.1. Behavioral Modeling

The simplest oversampling ΣΔ modulator consists of a first-order discrete-time filter connected in series with a comparator (1-bit ADC); with the output of the comparator subtracted from the input signal as in Figure 2.4 before.

From (2.7), the transfer function of an integrator of first order-sigma delta modulator is

$$H(z) = \frac{z^{-1}}{1 - z^{-1}} \tag{4.1}$$

In VHDL-AMS, the 'ZTF property which allows us to write any discrete-time transfer function directly is available as:

vmed==viin'ztf(num,den,tsamp,tsamp);

where num and den are vectors, whose values represent the coefficients of $1, z^{-1}, z^{-2}$, etc. The sampling frequency can be defined through the parameter "tsamp", sampling period. The non-idealities of the integrator can be modeled by modifying these coefficients.

1-bit ΣΔ Modulators include 1-bit comparator and 1-bit DAC blocks; whereas a multi-bit ΣΔ Modulator requires multi-bit comparator and DAC blocks and these blocks can be coded parametrically. In addition to these, the subtraction at the input node is accomplished using an arithmetic subtractor block and the non-idealities of jitter noise and thermal noise that will be mentioned in the following sections are added through this block.

4.2. Structural Modeling

The structural model contains the same comparator, DAC and subtractor blocks for 1 bit ΣΔ Modulators. The main difference is the integrator block which is implemented structurally as it is seen in Figure 4.1.

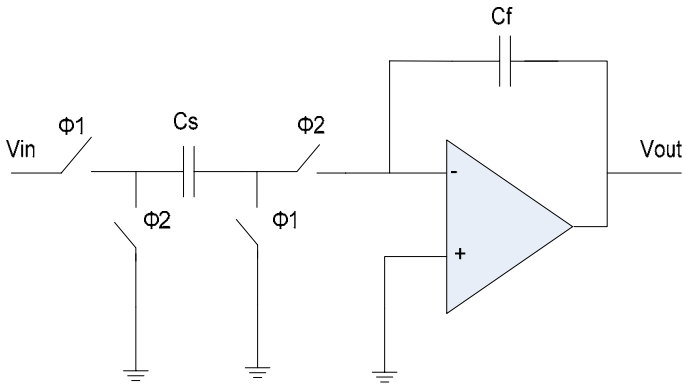

Figure 4.1. Switched Capacitor Integrator

Each pair of switches works from a two-phase clock. Cf is feedback capacitor and Cs is sampling capacitor. To implement the structural model, op amp macro model from MGC CommLib that contains various op amp non-idealities such as SR, finite OL gain, finite BW, saturation voltage, etc is used.

Comparator and subtractor blocks remain same, but integrator and DAC blocks are changed for multi-bit ΣΔ Modulators' structural modeling. In Figure 4.2, 3-bit unit element DAC is seen. For N bit digital input DAC, 2^N equivalent resistors and 1 N-2^N thermometer code decoder are used. Resistors and switches are same with the ones that are used in the structural modeling of the integrator.

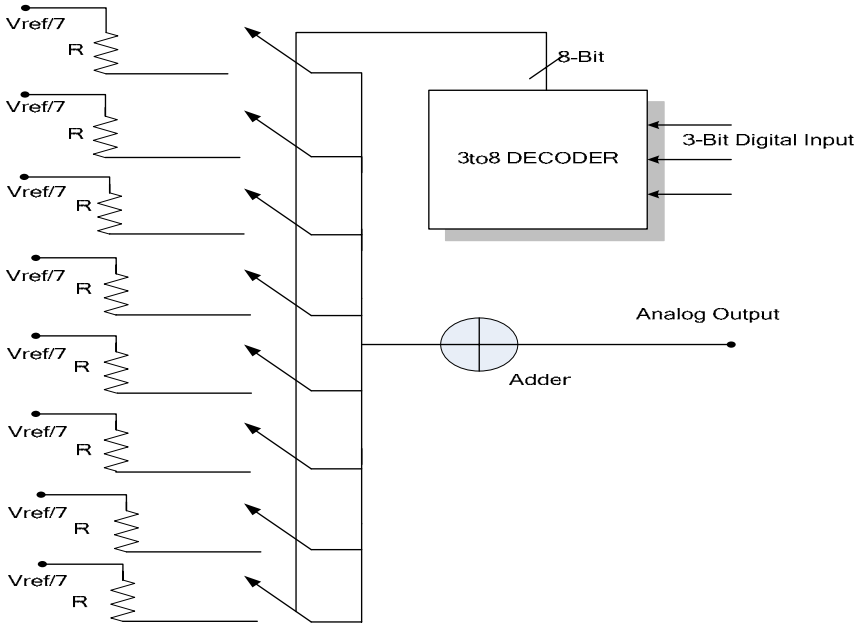

Figure 4.2. 3-bit Unit Element DAC

4.3. Modeling Non-Idealities of ΣΔ Converters

In literature [1], some errors such as finite DC gain error, capacitor mismatch produce changes in STF (z) and NTF (z). The impact of this type of errors depends on the modulator's order and architecture. The other group consists of errors whose effect can be modeled as an error source at the integrator input such as clock jitter, thermal noise, and op-amp noise. ΣΔ Modulators' non- idealities can be evaluated both structurally and behaviorally. In the following sections, these non-idealities will be mentioned and their non-ideality effects will be evaluated seperately. In this section, circuit non-idealities in ΣΔ Modulators will be encountered for both 1- bit Second Order ΣΔ Modulator and 3-bit Second Order ΣΔ Modulator. In all VHDL-AMS simulations in the following sections, the input sinusoidal signal amplitude is 0.2V (dynamic range set by the DAC is +/-0.5V), sampling frequency is 4 MHz and oversampling ratio is OSR=32.

4.3.1. Amplifier Finite DC-gain

The DC gain of the integrator described by [1] is infinite. However, the gain in practical circuits is limited by the open loop gain A_0. The consequence of the finite DC gain can be interpreted as "leakage" α that means only a fraction of the delayed output of the integrator is added to the new input sample [9].

$$\alpha = \frac{1 + \frac{1}{A_0}}{1 + \frac{1}{A_0}(1 + \frac{Cs}{Cf})} \tag{4.2}$$

where A_0 is the operational amplifier open-loop gain, Cf is feedback and Cs is sampling capacitor as it is seen in Figure 3.8.

The signal transfer function of the integrator for first order $\Sigma\Delta$ converter with "leakage" α which was (z^{-1}) in 2.8 becomes [9]

$$H(z) = \frac{z-1}{1-\alpha z^{-1}} \tag{4.3}$$

The DC gain of the integrator H_0 becomes

$$H_0(z) = H(1) = \frac{1}{1-\alpha} \tag{4.4}$$

The noise transfer function which was $(1 - z^{-1})$ in (2.9) becomes

$$NTF(z) = \frac{1-\alpha z^{-1}}{1+(1-\alpha)z^{-1}} \tag{4.5}$$

To include the finite DC gain in the structural modeling, the open loop gain parameter of the CommLib op amp macro model is changed from 1G to 2185 V/V.

In behavioral modeling, if the open loop gain A_0 is taken as 2185 V/V, leakage factor α is calculated as 0.999 from (4.2) where Cf and Cs are taken as 1 pF. This leakage factor α is added as a coefficient for integrator denominator of z^{-1} as it is mentioned in Section 4.1.

Structural and behavioral models for amplifier finite DC gain error give similar results in terms of PSD; we will prefer to use structural modeling as a reference. Figure 4.3. shows PSD of ideal $\Sigma\Delta$M (blue curve) and PSD of non-ideal $\Sigma\Delta$M with finite DC gain leakage α=0.999.

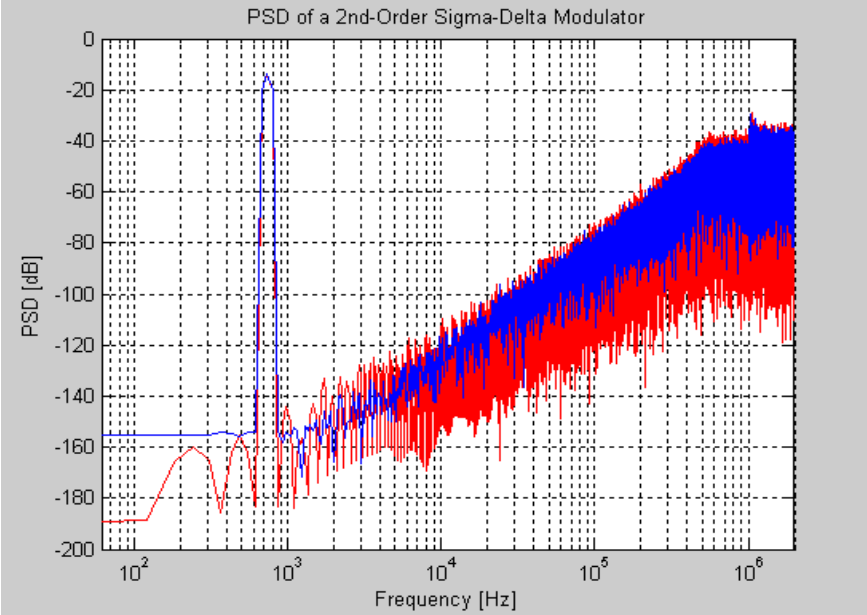

Figure 4.3. PSD of Second Order Sigma Delta Converter with α=0.999

4.3.2. Capacitor Mismatch

In a SC circuit, the gain factor is given by

$$g_i = \frac{Cf}{Cs} \tag{4.6}$$

The capacitors mismatch due to the fabrication process in Cs and Cs affect the gain g_i directly. Although today's fabrication processes can produce matching with an error as low as 0.1%, still the gains differ from their nominal values affecting the performance of the integrator. Assuming that the gain error in the integrator for each stage is given by [10]

$$g_k = g_{k,ideal}(1-\varepsilon_{gk}) \quad k = 1,2 \tag{4.7}$$

Then, the *STF* and *NTF* of the second-order modulator are given by

$$\left|STF(z)\right| \cong (1-\varepsilon_{g1})/(1-\varepsilon_{g1}') \tag{4.8}$$

$$\left|NTF(z)\right| \cong \frac{(1-z^{-1})^2}{(1-\varepsilon_{g1}')(1-\varepsilon_{g2})} \tag{4.9}$$

where it has been assumed that $g_1' = g_1$ and $g_2' = 2g_1g_2$ [10].

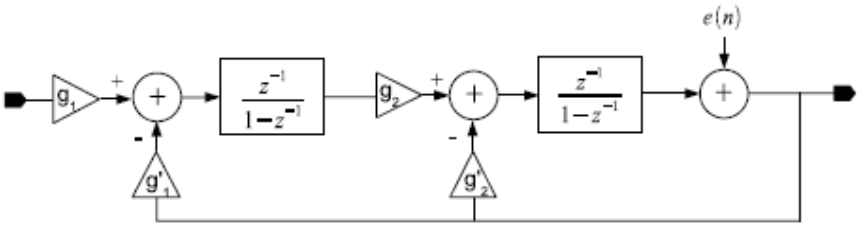

Figure 4.4. Second Order 1-bit $\Sigma\Delta$ Modulator with finite-gain error

If capacitor mismatch non-ideality factor ε_{gk} is taken as 10% and ideal case gain factors g_1 and g_2 become 0.45 or 055. If it is taken as 1%, ideal case the gain factors g_1 and g_2 become 0.495 or 0505. Figure 4.5 shows the effect capacitance mismatch on circuit performance. (Blue curve is PSD of ideal case; red curve is PSD of non-ideal $\Sigma\Delta$ M with capacitor mismatch of 10%.

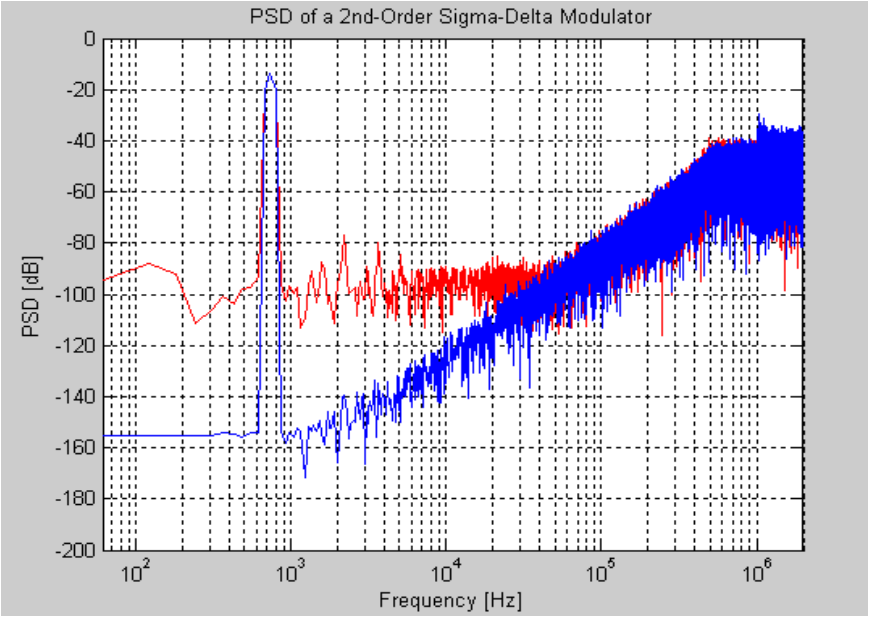

Figure 4.5. PSD of Second Order **ΣΔ** Converter with Capacitor Mismatch

4.3.3. Slew Rate and Finite GB

The effect of finite BW and the SR are related to each other, and may be interpreted as a nonlinear gain [11]. With reference to SC integrator shown in Figure 4.1., the evaluation of the output node during the *n*th integration period is:

$$v_0(t) = v_0(nT - T) + \alpha V_s\left(1 - e^{\frac{-t}{\tau}}\right), nT - \frac{T}{2} < t < nT \tag{4.10}$$

where $V_s = V_{in}(nT - T/2)$, α is the integrator leakage and $\tau = 1/(2\pi GBW)$ is the time constant of the integrator. The slope of the this curve reaches its maximum value when $t = 0$, resulting in

$$\frac{d}{dt}v_0(t)\bigg|_{\max} = \frac{\alpha V_s}{\tau} \tag{4.11}$$

If the value specified by 4.11 is lower than the op-amp slew-rate, SR, there is no slew-rate limitation and the evaluation of v_0 fits 4.10.

If the value specified by 4.11 is larger than *SR*, the op-amp is in slewing and the first part of the temporal evolution v_0 ($t_0 < T$) is linear with slope *SR*. The following equations hold (by assuming $t_0 < T$):

$$t \le t_0 \; v_0(t) = v_0(nT - T) + SRt \tag{4.12}$$

$$t > t_0 \; v_0(t) = v_0(t_0) + (\alpha V_s - SRt_0)\left(1 - e^{-\frac{t-t0}{\tau}}\right) \tag{4.13}$$

Imposing the condition for the continuity of the derivatives of 4.12 and 4.13 in t_0, we get

$$t_0 = \frac{\alpha V_s}{SR} - \tau \tag{4.14}$$

If $t_0 \ge T$, only (4.12) holds.

To model the finite BW and SR effects at behavioral model, one has to pass the input signal through a function before feeding it to the integrator. This piecewise-continuous function evaluates 2 cases: If the time derivative of the input signal is lower than the SR, no SR effect appears. Otherwise, the op-amp is in slewing and the transient response consists of a linear and an exponential part.

Modeling the slew rate non-ideality in the structural model is easier and the built-in op amp circuit in CommLib package of the Advance MS tool includes both effects as parameters. Figure 4.6 shows the effect of SR on circuit performance. For ideal case (blue curve), SR=10V/us and BW=1 kHz as we go from SR=10V/us to SR=1V/us (red curve), the noise floor increases dramatically-the slewing error deteriorates the noise shaping function (noise leakage).

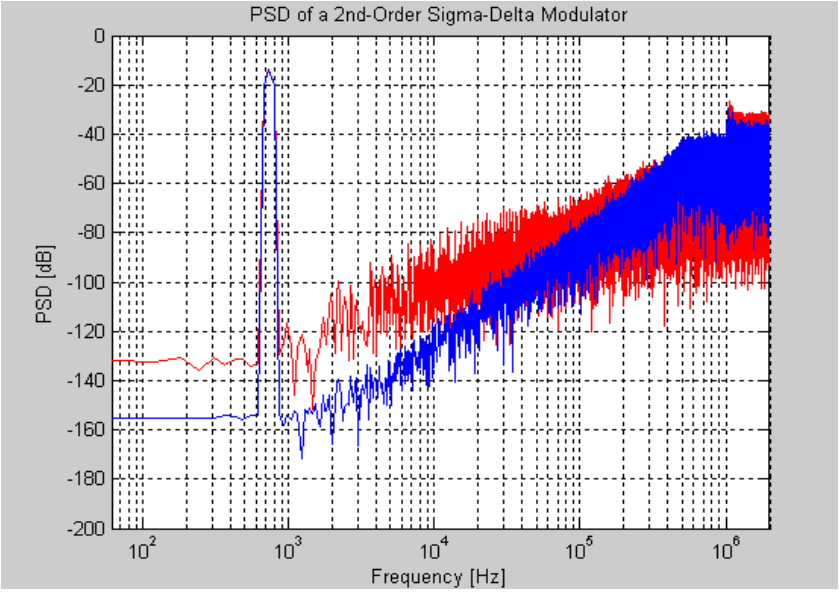

Figure 4.6. PSD of Second Order ΣΔM with Slew Rate Effect

4.3.4. Switches Thermal Noise

Thermal noise is caused by the random fluctuation of carriers due to thermal energy and is present even at equilibrium [11]. Thermal noise has a white spectrum and wide band limited only by the time constant of the switched capacitors or the bandwidths of op-amps. The sampling capacitor Cs in the single-ended SC integrator has a finite resistance Ron that periodically opens and it samples the thermal noise voltage onto Cs Figure 4.7.

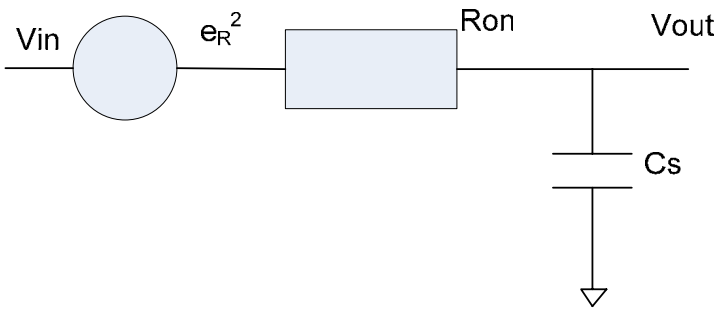

Figure 4.7. Switches thermal noise model

The total power can be found evaluating the integral [11].

$$e_R{}^2 = \int_0^\infty \frac{4kTRon}{1+(2\pi fRonCs)^2} df = \frac{kT}{Cs} \tag{4.13}$$

where k is the Boltzman constant and T is the absolute temperature in Kelvin.

The switch thermal noise is modeled as an additive white noise source of variance kT/Cs. To model a white noise source with zero mean and unity standard deviation in VHDL-AMS, the Box-Muller transformation is used. $n(t)$ is the Gaussian random process with unity standard deviation, $kT/Cs \cdot n(t)$ models the switch thermal noise, since its variance is kT/Cs. This noise is coded at the input of the subtractor block and this effect has been integrated to the structural and behavioral models using the same method. Figure 4.8 shows the effect of thermal noise on the PSD of the $\Sigma\Delta$ Modulator output. (Blue curve is PSD of ideal case; red curve is PSD of the kT/Cs model).

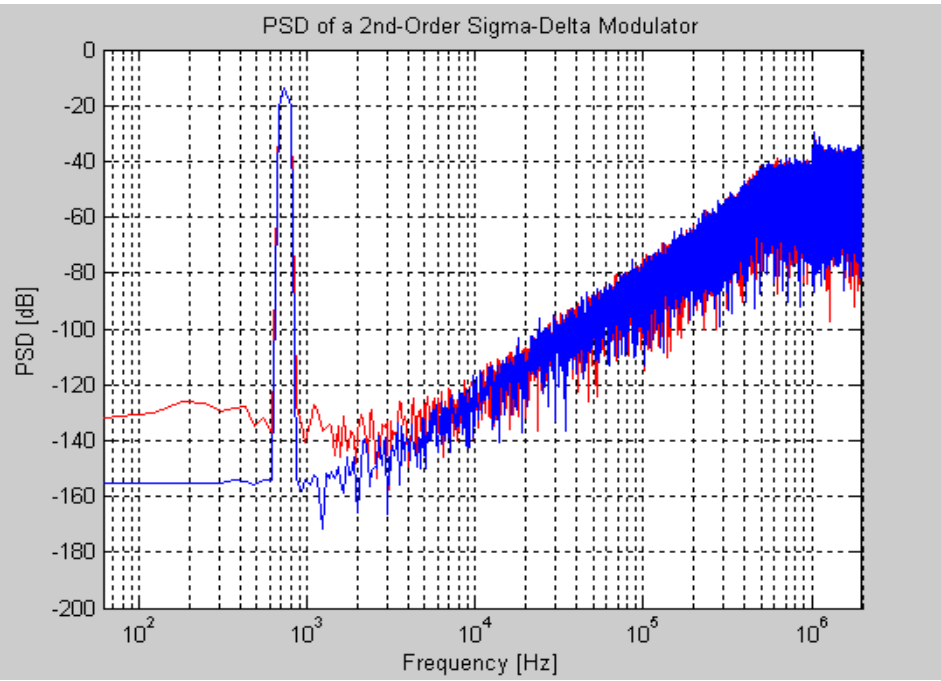

Figure 4.8. PSD of Second Order $\Sigma\Delta$ M with Thermal Noise

4.3.5. Clock Jitter

Clock jitter at the input sampling capacitor results in a non-uniform sampling time sequence, and produces an error which increases the total power at the quantizer output [12]. The magnitude of this error is a function of both the statistical properties of the jitter and the modulator input signal [9]. The error introduced when a sinusoidal signal $x(t)$ with amplitude A and frequency f_{in} is sampled at an instant which is in error by an amount δ is given by

$$x(t+\delta) - x(t) \approx 2\pi f_{in}\delta A\cos(2\pi f_{in}t) = \delta\frac{d}{dt}x(t) \tag{4.14}$$

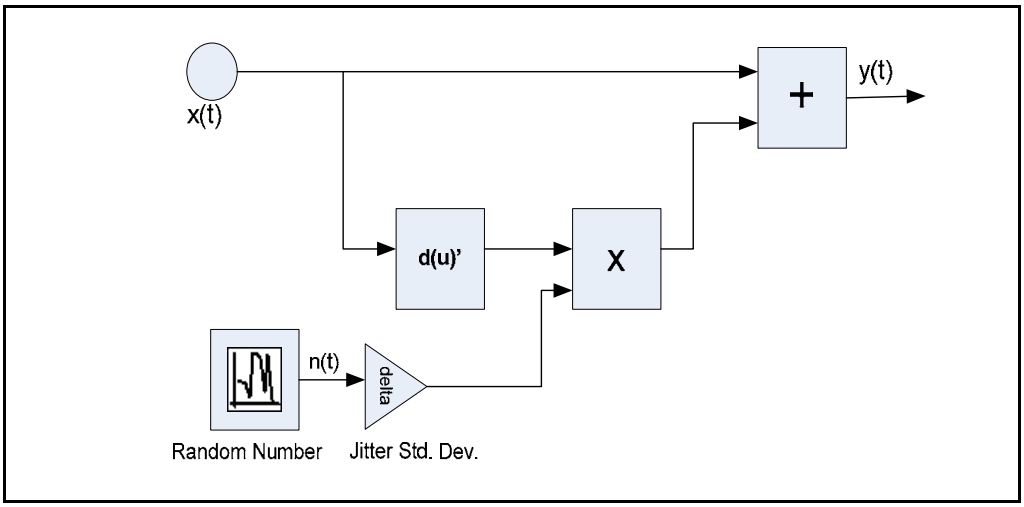

Figure 4.9. Random sampling jitter

The signal $n(t)$ is Gaussian random number with mean value zero and standard deviation unity. In VHDL-AMS this sequence is created by the white noise generator code. Multiplying the number with delta (Δ) changes the standard deviation of the distribution and then this number is added to the input signal as it is seen in Figure 4.9. Clock jitter do not have a direct effect on the modulator behavior, it has an effect on the sampling of the input signal. Due to this, clock jitter model is defined at the input of the subtractor block just like switches thermal noise. Figure 4.10. shows the effect of non-uniform sampling

with sampling uncertainty. (Blue curve belongs to PSD of ideal ΣΔM, red curve belong to non-ideal ΣΔM with sampling jitter Δτ, of 10 ns)

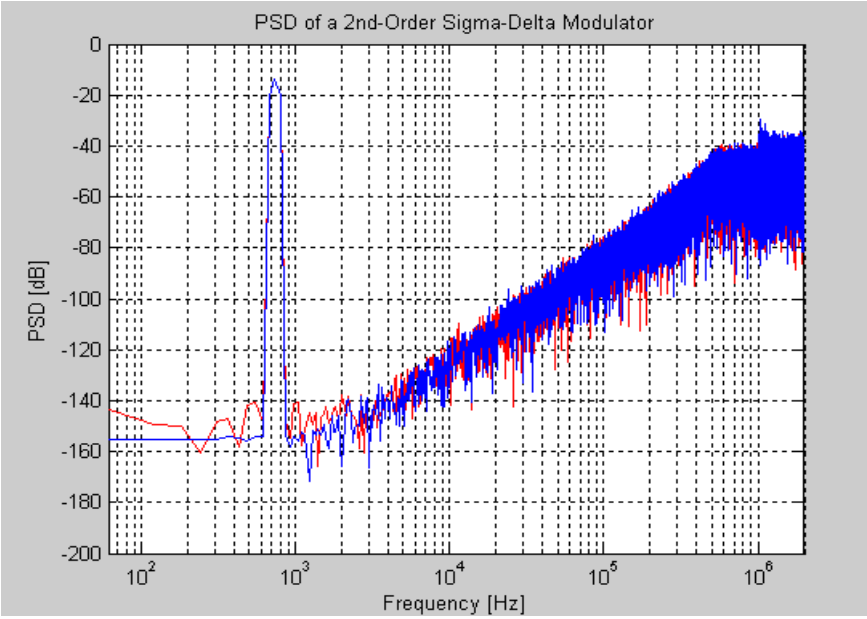

Figure 4.10. PSD of Second Order ΣΔM with Clock Jitter

4.3.6. Multi-bit Quantizer Non-Linearity

The modulators with more than two internal quantization levels called multi-bit modulators are insensitive to the quantizer non-idealities. The main reason of multi-bit quantizer non-ideality is the component mismatch in the DAC. References in a DAC are typically voltages and currents generated by matched components such as resistors, transistors and capacitors. Fabrication process variations, temperature gradients across the circuit, component aging and noise can cause circuit component values to differ from their design value.

Figure 4.11. shows a linear model of a second-order multi-bit ΣΔM. Noise sources from the quantization noise, ADC threshold errors in the comparators and the DAC mismatch are represented by $e(n)$, $c(n)$ and $d(n)$, respectively. It can be observed that unlike $e(n)$ and $c(n)$, the error $d(n)$ lies inside the feedback path. Hence, it is not shaped by

the NTF in the feedback of the modulator. Consequently, the linearity of the modulator is no better than the linearity of the *N*-bit internal DAC. (See Section 7)

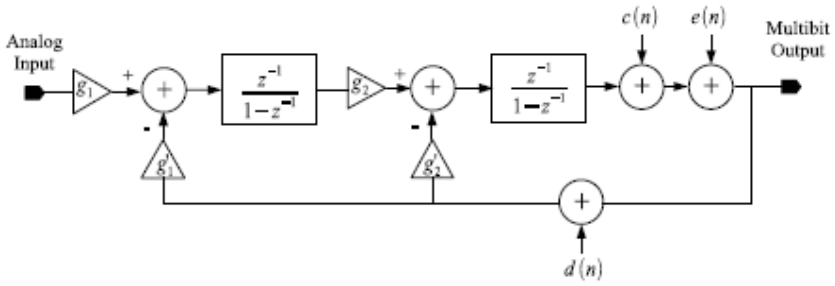

Figure 4.11. Multi-bit ΣΔM linear model with noise from elements mismatch.

In Table 4.1., 3-bit DAC [see Figure 4.2.] ideal output and output with mismatch can be seen. In our application to model 3-bit quantizer non-linearity effect behaviorally, ε_k values for each corresponding digital input are added to DAC outputs. If unit element mismatch in the system is 1%, the values of each ε_k will be 0.01/7 in behavioral modeling.

Table 4.1. DAC input/output map including mismatched units.

Digital Input	Thermometer Code	Ideal Output	Output with mismatch
“000”	“00000000”	0	0
“001”	“00000001”	1/7	$1/7+\varepsilon 1$
“010”	“00000011”	2/7	$2/7+\varepsilon 1+\varepsilon 2$
“011”	“00000111”	3/7	$3/7+\varepsilon 1+\varepsilon 2+\varepsilon 3$
“100”	“00001111”	4/7	$4/7+\varepsilon 1+\varepsilon 2+\varepsilon 3+\varepsilon 4$
“101”	“00011111”	5/7	$5/7+\varepsilon 1+\varepsilon 2+\varepsilon 3+\varepsilon 4+\varepsilon 5$
“110”	“00111111”	6/7	$6/7+\varepsilon 1+\varepsilon 2+\varepsilon 3+\varepsilon 4+\varepsilon 5+\varepsilon 6$
“111”	“01111111”	1	$1+\varepsilon 1+\varepsilon 2+\varepsilon 3+\varepsilon 4+\varepsilon 5+\varepsilon 6+\varepsilon 7$

For simplicity in structural modeling, each nonlinearity factor ε_k is added to resistor values used in multi-bit DAC as (R+/-ε_k) (See Section 7.3 for further information for modeling σ_ε in VHDL-AMS). Figure 4.12. shows PSD of Second Order two different 3-bit ΣΔMs. Blue curve corresponds to PSD of ideal ΣΔM, while red curve corresponds to non-ideal ΣΔM with 1% Resistor Mismatch in multi-bit DAC.

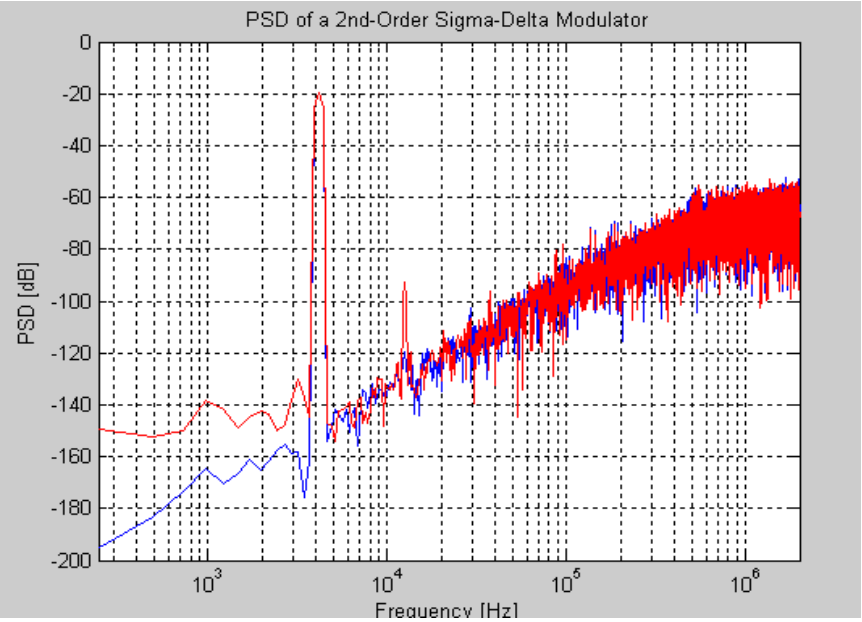

Figure 4.12. PSD of Second Order 3-bit ΣΔM with 1% Mismatch

5. ADAPTIVE FILTER OVERVIEW

The goal of any filter is to extract useful information from noisy data. Whereas an ordinary fixed filter is designed in advance with knowledge of the statistics of both the signal and the unwanted noise, the adaptive filter continuously adjusts to a changing environment through the use of recursive algorithms. This is useful when either the statistics of the signals are not known beforehand or change with time. The discrete adaptive filter (see Figure 5.1.) accepts an input $u(n)$ and produces an output $y(n)$ by a convolution with the filter's weights $w(k)$. A desired reference signal $d(n)$ is compared to the output to obtain an estimation error $e(n)$. This error signal is used to incrementally adjust the filter's weights for the next time instant.

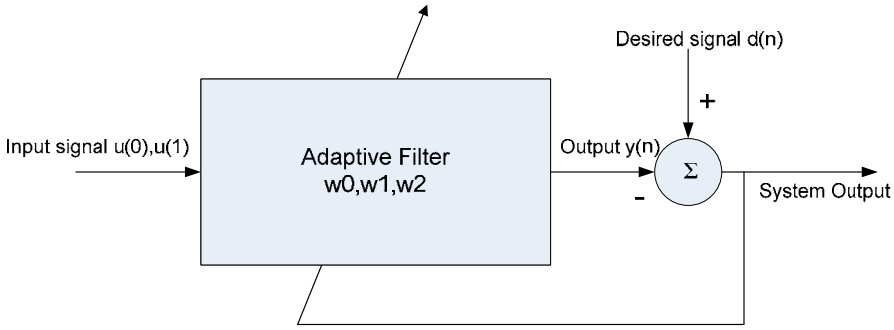

Figure 5.1. A discrete adaptive filter

5.1. Adaptive Algorithms

There are numerous methods for the performing weight update of an adaptive filter. There is the Wiener filter, which is the optimum linear filter in the terms of mean squared error, and several algorithms that attempt to approximate it, such as the method of steepest descent. There is also least mean square algorithm, developed by Widrow and Hoff originally for use in artificial neural networks. Finally, there are other techniques such as the recursive least squares algorithm and the Kalman filter. The choice of algorithm is

highly dependent on the signals of interest and the operating environment, as well as the convergence time required and computation power available.

5.2. Wiener Filters

The Wiener filter, so named after its inventor, was developed in 1949. It is the optimum linear filter in the sense that the output signal is as close to the desired signal as possible. Although not often implemented in practice due to computational complexity, the Wiener filter is studied as a frame of reference for the linear filtering of stochastic signals [13] to which other algorithms can be compared. To formulate the Wiener filter and other adaptive algorithms, the mean squared error (MSE) is used. If the input signal $u(n)$ to a filter with M taps is given as

$$u(n) = \left[u(n), u(n-1), ..., u(n-M+1)\right]^T \tag{5.1}$$

and the coefficients or weight vector is given as

$$w(n) = \left[w(0), w(1), ..., w(M-1)\right]^T \tag{5.2}$$

then the square of the output error can be formulated as

$$e_n{}^2 = d_n{}^2 - 2d_n u_n{}^T w + w^T u_n u_n{}^T w \tag{5.3}$$

The mean square error, J, is obtained by taking the expectations of both sides:

$$\begin{aligned} J &= E\left[e_n{}^2\right] = E\left[d_n{}^2\right] - 2E\left[d_n u_n{}^T w + w^T u_n u_n{}^T w\right] \\ &= \sigma^2 + 2p^T w + w^T R w \end{aligned} \tag{5.4}$$

Here, σ is the variance of the desired output, p is the cross-correlation vector and R is the autocorrelation matrix of u. A plot of the MSE against the weights is a non-negative bowl shaped surface with the minimum point being the optimal weights. This is referred to as the error performance surface [13], whose gradient is given by

$$\nabla = \frac{dJ}{dw} = -2p + 2Rw \tag{5.5}$$

To determine the optimal Wiener filter for a given signal requires solving the Wiener-Hopf equations. First, let the matrix R can denote the M-by-M correlation matrix of u. That is,

$$R = E\left[u(n)u^{H}(n)\right] \tag{5.6}$$

where the superscript H denotes the Hermitian transpose. In expanded form this is

$$R = \begin{bmatrix} r(0) & r(1) & .. & r(M-1) \\ r^{*}(1) & r(0) & .. & r(M-2) \\ .. & .. & .. & .. \\ r^{*}(M-1) & r^{*}(M-2) & .. & r(0) \end{bmatrix} \tag{5.7}$$

Also, let p represent the cross-correlation vector between the tap inputs and the desired response $d(n)$:

$$p = E\left[u(n)d^{*}(n)\right] \tag{5.8}$$

which expanded as:

$$p = \left[p(0), p(-1), ..., p(1-M)\right]^{T} \tag{5.9}$$

Since the lags in the definition of p are either zero or negative, the Wiener-Hopf equation may be written in compact matrix form:,

$$Rw_{o} = p \tag{5.10}$$

with w_o stands for the M-by-1 *optimum tap-weight vector* [13], for the transversal filter. That is, the optimum filter's coefficients will be:

$$wo = \left[w_{o0}, w_{o1}, ..., w_{oM-1}\right]^T \tag{5.11}$$

This produces the optimum output in terms of the mean-square-error, however if the signals statistics change with time then the Wiener-Hopf equation must be recalculated. This would require calculating two matrices, inverting one of them and then multiplying them together. This computation cannot be feasibly calculated in real time, so other algorithms that approximate the Wiener filter must be used.

5.3. Method of Steepest Descent

With the error-performance surface defined previously, one can use the method of steepest-descent to converge to the optimal filter weights for a given problem. Since the gradient of a surface (or hypersurface) points in the direction of maximum increase, then the direction opposite the gradient ($-\nabla$) will point towards the minimum point of the surface. One can adaptively reach the minimum by updating the weights at each time step by using the equation

$$w_{n+1} = w_n + \mu(-\nabla_n) \tag{5.12}$$

where the constant μ is the step size parameter. The step size parameter determines how fast the algorithm converges to the optimal weights. A necessary and sufficient condition for the convergence or stapility of the steepest descent algorithm [13] is for μ to satisfy:

$$0 < \mu < \frac{2}{\lambda_{\max}} \tag{5.13}$$

where $\lambda_{\max}$ the largest eigen value of the correlation matrix R.

5.4. Least Mean Square Algorithm

The least mean square (LMS) algorithm is similar to the method of steepest-descent in that it adapts the weights by iteratively approaching the MSE minimum. Widrow and Hoff invented this technique in 1960 for use in training neural networks. The key is that instead of calculating the gradient at every time step, the LMS algorithm uses a rough approximation to the gradient.

The error at the output of the filter can be expressed as

$$e_n = d_n - w_n^T u_n \tag{5.14}$$

which is simply the difference of the desired output and the actual filter output. Using this definition for the error an approximation of the gradient is found by

$$\nabla = -2e_n u_n \tag{5.15}$$

Substituting this expression for the gradient into the weight update equation from the method of steepest-descent gives

$$wn+1 = wn + 2\mu \cdot e_n u_n \tag{5.16}$$

which is the Widrow-Hoff LMS algorithm. As with the steepest-descent algorithm, it can be shown to converge [13] for values of μ less than the reciprocal of λ_{max}, but λ_{max} may be time-varying, and it another criterion can be used to avoid computing.

This is:

$$0 < \mu < \frac{2}{MS_{\max}} \tag{5.17}$$

where M is the number of filter taps and S_{max} is the maximum value of the power spectral density of the tap inputs u.

The relatively good performance of the LMS algorithm given its simplicity has caused it to be the most widely implemented in practice. For an N tap filter, the number of operations has been reduced to 2*N multiplications and N additions per coefficient update. This is suitable for real-time applications, and is the reason for the popularity of the LMS algorithm.

6. LMS BASED ADAPTIVE DIGITAL ERROR CORRECTION

The resolution of high speed $\Sigma\Delta$ Modulators is limited by reference voltage errors, component mismatches, finite amplifier gain, finite settling times, comparator offsets, amplifier offsets, charge injection errors and component non-linearity effects [14]. Background self calibration techniques have been used to improve the resolution and linearity of the ADC without stopping the input conversion. One of the main ideas proposed in this thesis is to use a LMS based adaptation algorithm to correct for all static errors in the digital domain.

6.1. Design Objective

A block diagram of a 1-bit per stage pipelined ADC is illustrated in Figure 6.1. For switched capacitor implementation of the pipeline stages, the output residue voltage of pipeline stage I, $Vout(i)$ can be expressed in (6.1). Finite op-amp gain coefficient α, the capacitor ratio error ε, charge injection δ and non-ideal reference voltage $Vref(i)$ are also included.

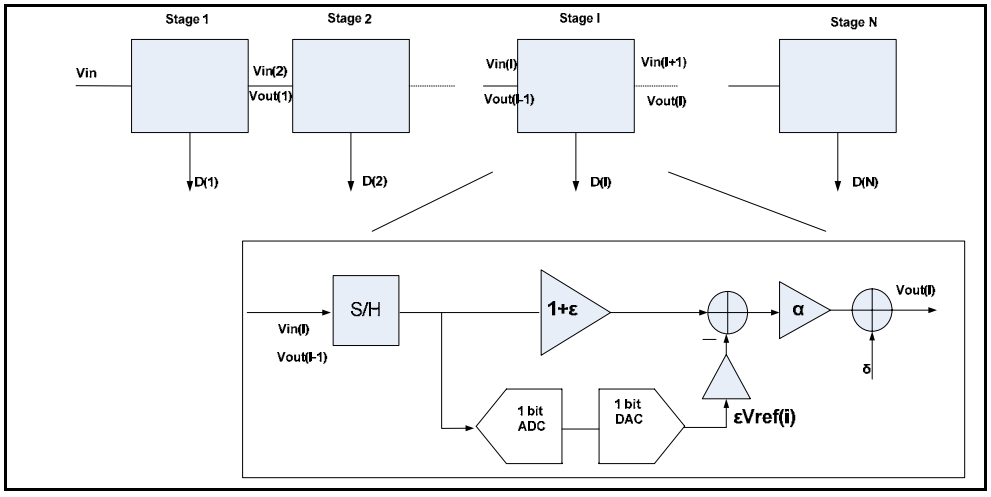

Figure 6.1. Pipelined 1-bit per stage ADC [14]

$$Vout(I) = \alpha(1+\varepsilon)Vout(i-1) - \alpha\varepsilon D(I)Vref(I) + \delta \tag{6.1}$$

For N stage pipeline ADC, (6.2) will be obtained.

$$Vin = \frac{V_N}{\alpha^N (1+\varepsilon)^N} + \sum_{i=1}^{N}\left(\frac{\varepsilon Vref(i)}{\alpha^{i-1}(1+\varepsilon)^i} D(i) - \frac{\delta}{\alpha^i (1+\varepsilon)^i} \right) \tag{6.2}$$

Neglecting the first term, the expression for the input signal in terms of the digital output bits:

$$Vin \cong \sum_{i=1}^{N} \frac{\Delta Vref(i)}{\alpha^{i-1}(1+\varepsilon)^i} D(i) - \delta \tag{6.3}$$

By setting $\alpha = 1, \varepsilon = 1,\ \delta = 0$ and $Vref(i) = Vref$ for all *i*, the ideal expression for the input in terms of digital output bits is obtained:

$$Vin = \sum_{i=1}^{N} \frac{Vref}{2^i} D(i) \tag{6.4}$$

As it is seen in (6.4) the major sources of errors in pipelined ADCs are of finite amplifier gain leakage α, capacitor ratio mismatch ε and charge injection δ.

(6.5) is normalized by the reference voltage to get the digital equivalent of the input.

$$Vin = \sum_{i=1}^{N} \frac{D(i)}{2^i} \tag{6.5}$$

If (6.3) and (6.4) are compared, we find that the digital output deviates from the ideal due to non-idealities and it must be modified to correct errors. The problem of calibration is how to estimate $b(i)$ in the following expression:

$$Vin = \sum_{i=1}^{N} \frac{b(i)D(i)}{2^i} \tag{6.6}$$

The digital output of the main ADC is processed in digital domain to incorporate the estimation parameters $b(i)$. The slow, low power ADC connected in parallel is used to generate the ideal digital output D_{ideal} for the same input. The following Least Mean Squares (LMS) algorithm is used to estimate the coefficient $b(i)$.

$$b(i)new = b(i)old - \mu \frac{\partial e^2}{\partial b(i)} \tag{6.7}$$

where $e = D_{ideal} - D_{out}$.

The basic idea behind the error correction scheme in Figure 6.2 is to correct for the residue errors in the non-ideal pipeline stages by using a suitable set of parameters which are determined by comparing the digital output of the ADC with the digital output of a slow, low power high resolution reference ADC.

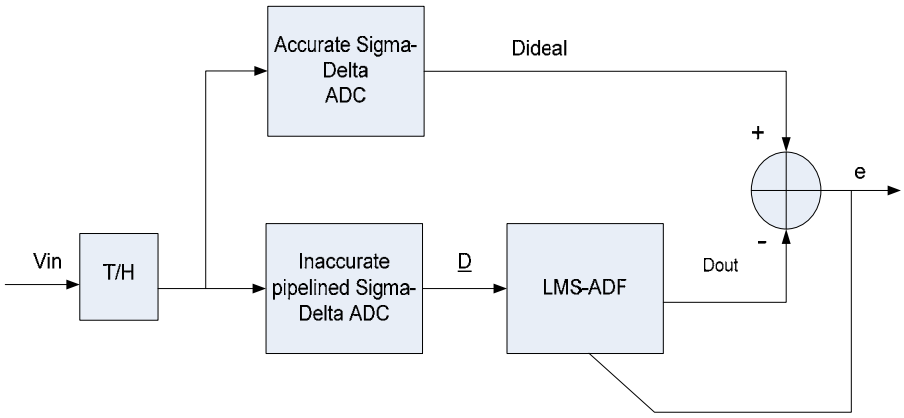

Figure 6.2. Error correction of pipelined **ΣΔ** ADC: Adaptive LMS Method

D is output vector of a high speed, inaccurate pipelined ADC and it is applied to an adaptive filter digital filter (ADF). Slow-but-Accurate ADC is used to obtain $D_{ideal.}$ The ADF tap values are updated using a least mean square (LMS) algorithm driven by the error signal, $e = D_{ideal} - D_{out}$.

6.2. Design Approach

6.2.1. Matlab Simulink Implementation

Initial implemention of LMS Adaptive Filter for digital error correction and calibration of Sigma Delta modulators is performed with Matlab Simulink. In the design of this LMS Filter, Sigma Delta toolbox by S. Brigati, F. Francesconi, P. Malcovati, D. Tonietto, A. Baschirotto and F. Maloberti is used [11]. Sigma Delta toolbox includes many set of models taking into account most of the Sigma Delta modulator non-idealities, such as sampling jitter, kT/C noise and operational amplifier parameters (noise, finite gain, finite bandwidth, SR and saturation voltages). All simulations in Section 6.2.1 and 6.2.2 are performed with following settings: the input sinusoidal signal amplitude is 0.2V (dynamic range set by the DAC is +/-0.5V), sampling frequency is 4 MHz and oversampling ratio is OSR=32.

Figure 6.3. shows the Matlab Simulink Model of our adaptive LMS Filter. The main purpose of this filter is to extract a desired signal from a noise corrupted signal by canceling the noise. The input signal for our LMS filter is 1-bit second order ΣΔ Modulator with finite gain bandwidth, capacitor mismatch, reference voltage offset error non-idealities. The desired signal is the output of an ideal 1-bit Second Order ΣΔ Modulator that can be seen at the bottom left side of Figure 6.3. All non-idealities such as jitter effect, thermal noise, finite gain error, SR, capacitor mismatch are disabled on ideal 1-bit Second Order ΣΔ M. Similar to Polyphase FIR decimation and CIC decimation filter design, the input vector size of LMS filter is defined as generic type and input word length is adapted with respect to input and desired signals. It should be noted that input signal's type and size should be same with desired signal's type and size.

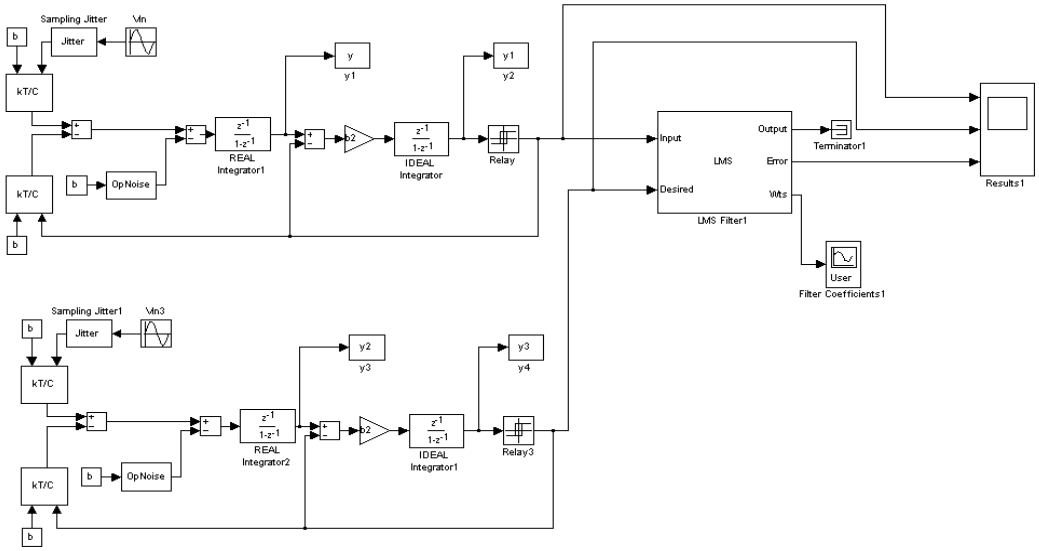

Figure 6.3. Matlab Simulink Model of LMS Adaptive Filtering

Table 6.1. shows the circuit non-idealities that are added to input signal [2]. It should be noted Sigma Delta toolbox does not include reference voltage offset error non-ideality and capacitor mismatch models. Sampling capacitor mismatch non-ideality is added to system by changing finite gain parameters g_1 and g_2 like in Section 4.3.2. Op-amp input-referred offset, comparator offset and comparator noise are also added to Sigma Delta toolbox by attaching DC offset voltages mentioned in Table 6.1 to corresponding nodes.

Table 6.1. Non-ideality parameters

Non-Idealities	**Value**
Sampling Cap. Mismatch	20%
Op-Amp DC gain leakage	0.99
Op-amp input-referred offset	10% Vref
Comparator offset	10% Vref
Comparator Noise	1% Vref

The convergence speed of LMS method depends on two factors: the eigenvalue distribution of the input autocorrelation matrix R, and the chosen initial condition. Modes corresponding to small eigenvalues converge much more slowly than those corresponding to large eigen values, and the initialization of the algorithm determines how much excitation is received by each of these modes [13]. As it is indicated in (5.17) adaptation step size μ should be bigger than 0 and smaller than $2/MS_{max}$ (M is the number of filter taps and S_{max} is the maximum value of the power spectral density of the tap inputs). After some iterations and taking into account the practical reasons prevent extra iterations, adaptation step size μ is chosen as 0.02 for 20 taps.

In Figure 6.4. blue curve shows ideal case (SNR 55.2 dB), red curve shows non-ideal case (SNR 45 dB) and green curve shows enhanced ΣΔ Modulator (with LMS Adaptive Filtering (SNR 54.1 dB). It means that with LMS Adaptive filter 9 dB SNR improvement is achieved. Mean Square Error (MSE) of input signal versus number of iterations can also be seen Figure 6.5. (There is a very smooth and accurate convergence in Matlab Model)

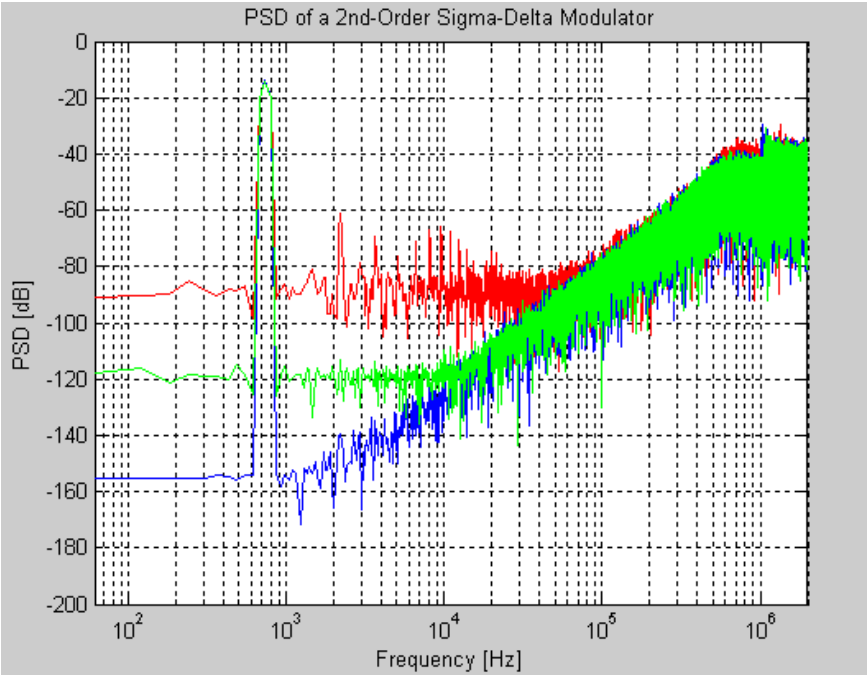

Figure 6.4. Ideal case (blue), non-ideal case (red) and enhanced ΣΔ Modulator

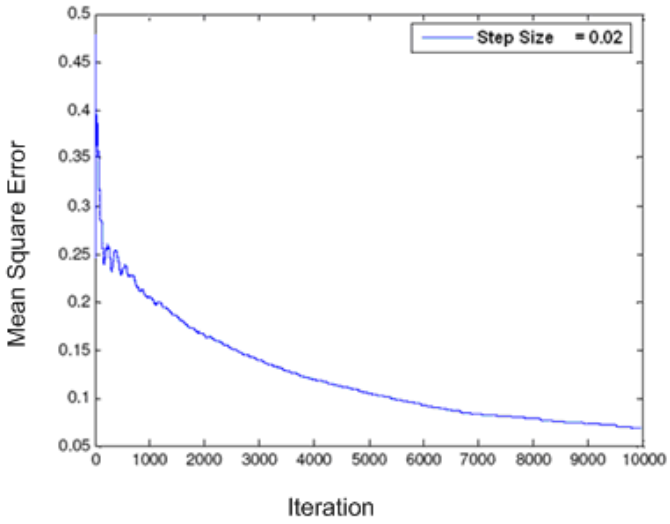

Figure 6.5. Mean Square Error (MSE) for Simulink Model

6.2.2. VHDL-AMS Implementation

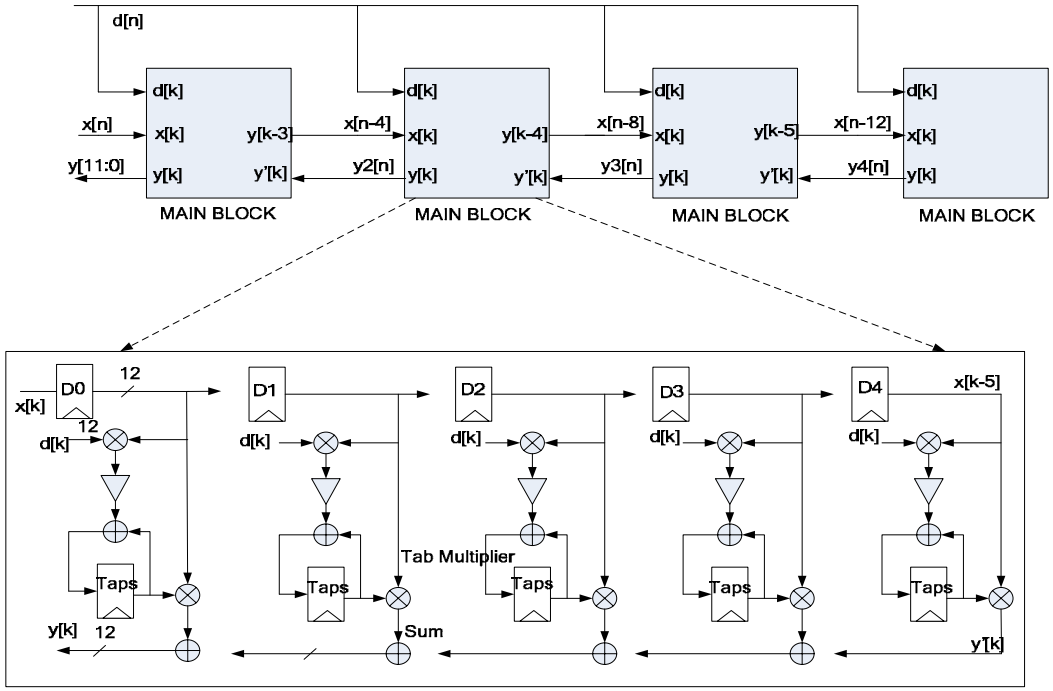

Figure 6.6. Block diagram of 20 taps LMS adaptive filter

As it is seen in Figure 6.6., top level of LMS adaptive filter consists of four 5-tap basic LMS blocks. The basic LMS blocks consist of: 1. Four 12 bit registers labeled “D0”

to “D4” for the delay line; 2. Five 12-bit registers labeled “Taps” for the taps; 3. Five multipliers labeled as “Tap Multiplier” to form the tap products; 4.Four adders shown as labeled “Sum” to form the sum of the products; 5. Five multipliers labeled “Tap Update Multiplier” for to compute the coefficient update.

Each filter block as in it seen above accepts three different inputs. In our application, 12-bit $x[k]$ is the output of inaccurate pipelined $\Sigma\Delta$ Converter as seen in Figure 6.3 , 12-bit $d[k]$ is the output of accurate pipelined $\Sigma\Delta$ Converter and 12-bit $y[k]$ is the output of the previous stages. The output $y[k]$ is the sum of $y'[k]$ and the convolution computed in that block. Each filter block also outputs a delayed $x[k]$ for the next block.

It should be noted that 20-tap filter is divided into 4 sections. The four sections are not identical because the first section contains only 4 register banks for the delay line since the first tap does not need a register. The second, third and fourth all have 5 register banks. The final filter block does not require an input $y'[k]$ from another block, so it saves one adder as well.

The circuit non-idealities that are added to input signal are same with Table 6.1 and adaptation step size μ is 0.02. In Figure 6.7 blue curve shows ideal case (SNR 55.2 dB), red curve shows non-ideal case (SNR 45 dB) and green curve shows enhanced $\Sigma\Delta$ Modulator (with LMS Adaptive Filtering (SNR 52.1 dB). It means that 7 dB SNR improvement is achieved with LMS Adaptive filtering.

Figure 6.8. exhibits MSE of input signal. As it is seen in this figure, mean square error converges to 0.05 very fast at first 1000 iterations. At iteration number 4500, there is a divergence to 0.1. However, LMS scheme starts to converge to 0 after 5000^{th} iteration again and finally mean square error is decreased to desired level at the end of iteration steps.

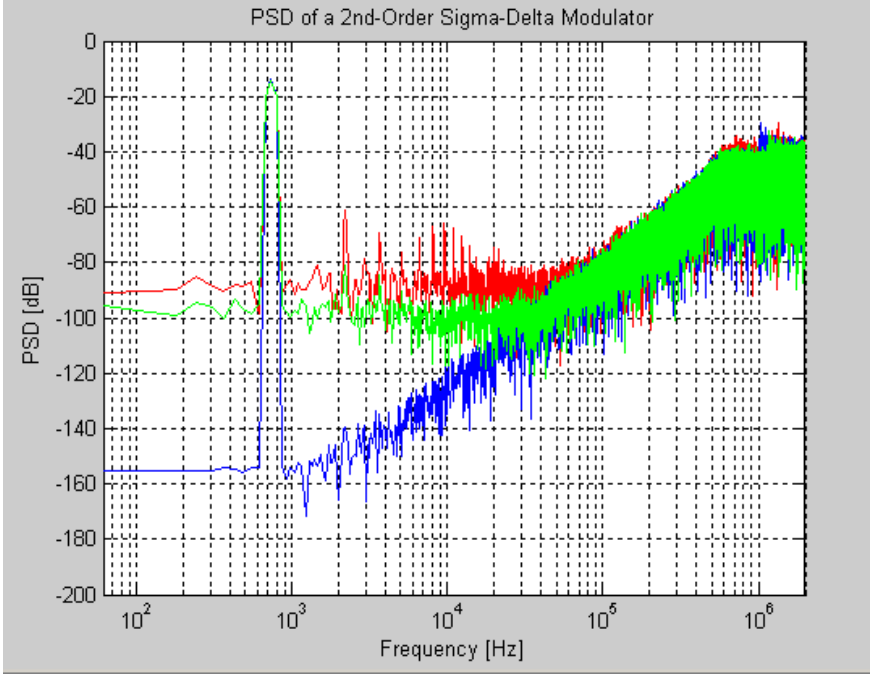

Figure 6.7. Ideal case (blue), non-ideal case (red) and enhanced $\Sigma\Delta$ Modulator

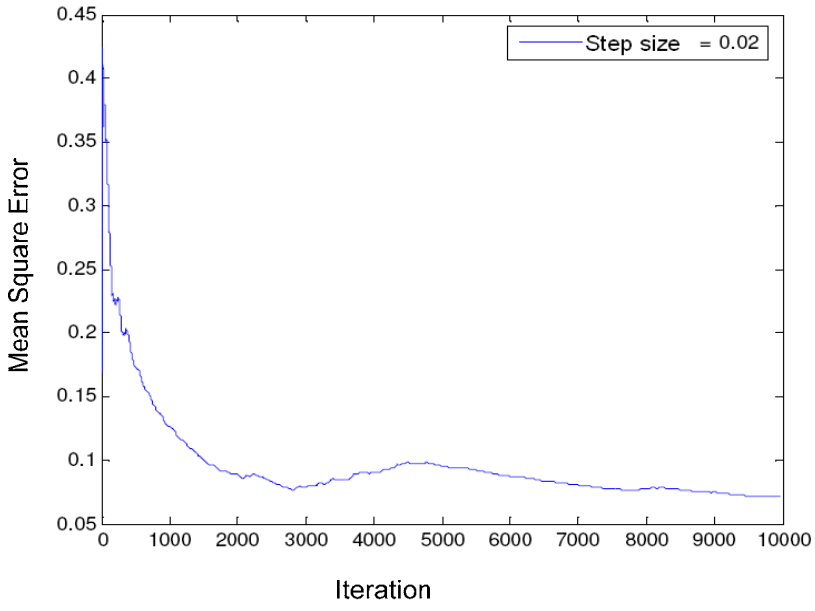

Figure 6.8. Mean Square Error (MSE) for VHDL-AMS Model

7. LINEARITY ENHANCEMENT OF MULTIBIT DAC

Oversampled multi-bit ΣΔ Converters are well known for their ability to achieve a high-resolution of medium to low frequency signals. However, the linearity of a multi-bit ΣΔ modulator is limited by the linearity of its multi-bit internal DAC. High DAC linearity requires precise matching of the DAC unit elements. Two signal processing strategies have been developed to correct for the DAC nonlinearity due to static element mismatch errors.

1. Digital Self Calibration: Compared to off-line calibration techniques [15-16], background (on-line) calibration schemes [17-18] have the advantage of being robust to environmental changes. These schemes use both off-line and on-line calibration schemes to improve the effective DAC linearity.
2. Dynamic Element Matching (DEM): An algorithm, implemented using the element selection logic selects different DAC unit elements to represent a given digital code at different times. This shaped mismatch noise can be filtered out by the decimation filter. DEM techniques differ in their effectiveness in shaping the spectrum of the DAC mismatch errors and in the complexity of the element selection logic. Different types of DEM method and their implementation will be mentioned in Section 7.2 in detail.

7.1. Digital Self Calibration

A digital self calibration scheme described in [18] is for multi-bit ΣΔ ADCs with resistor-string and switched-capacitor DACs. As shown in Figure 7.1., the internal DAC has two analog outputs: $v1$ is input to the loop filter, and v_T is input to the calibration ADC (DS_2).

The calibration ADC is used to derive a digital estimate $\hat{e_D}$ of the DAC errors e_D for all output levels, and stores these in a RAM. During conversion, internal DAC errors are corrected by filtering the RAM outputs via the FIR filter $NLFd(z)$ and subtracting the result from the digital output $d1$ of DS_1.

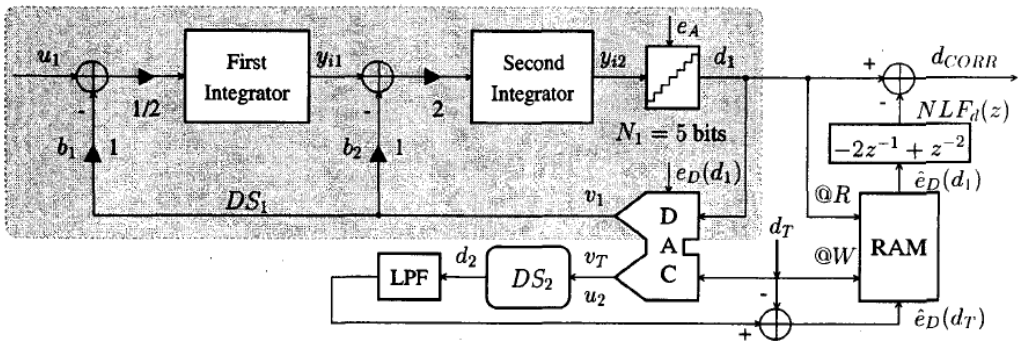

Figure 7.1. Multibit $\Sigma\Delta$ Modulator with mixed-mode error correction.

However, background calibration schemes are expensive to implement in terms of system design complexity, hardware requirements, and power consumption.

7.2. Dynamic Element Matching (DEM)

A DAC converts a digital number $d[n]$ into an analog voltage $v[k]$ or current $i[k]$. It also usually holds this voltage or current until the next sample, since it converts the signal to continuous tine as a first-order hold function.

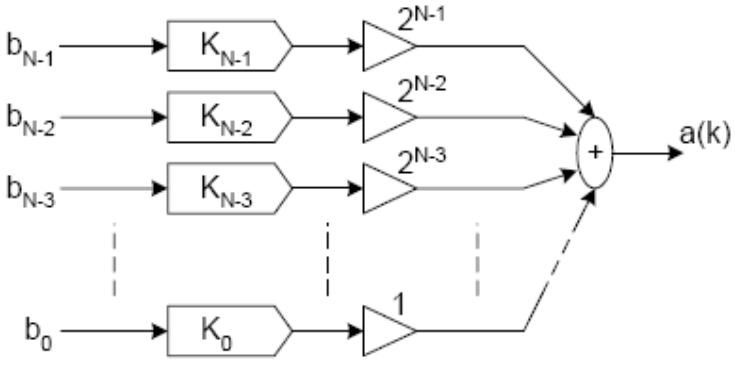

Figure 7.2. Binary weighted element DAC block diagram

They are typically implemented using DAC elements these being voltage sources or current sources. The simplest way to implement a binary DAC is to use binary weighted elements or unit elements methods, Figure 7.2. and Figure 7.3.

In binary weighted element DAC, most significant bits (MSB) elements will be more critical than the others and they are not suitable for high resolution applications.

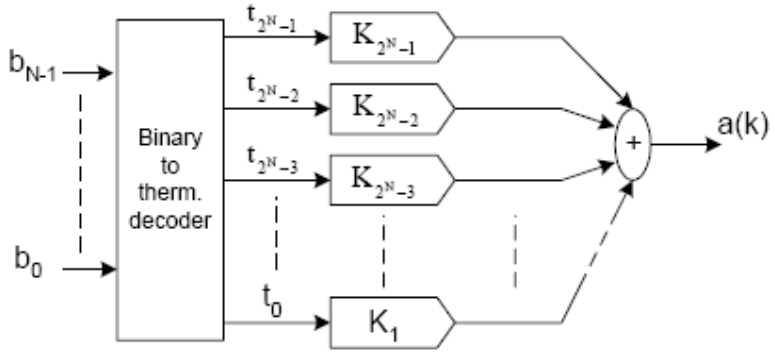

Figure 7.3. Unit Element DAC block diagram

In unit element DAC, the binary number is decoded to thermometer code representation and is passed through $2^N - 1$ equal unit elements. All elements are of equal (LSB) size and the nominal gain of the DAC will be given as:

$$\hat{K} = \frac{\sum_{i=0}^{2^{N-1}} K_i}{2^N} \tag{7.1}$$

The Integral of Non-linearity (INL) for any output value $a(k)$ will be given by:

$$INL(k) = \sum_{i=0}^{a(k)} K_i - a(k)\hat{K} = \sum_{i=0}^{2^{N-1}} t_i (K_i - \hat{K}) \tag{7.2}$$

7.2.1. Randomization

The simplest method of dynamic element matching is randomization of the element selection. Randomization is implemented by choosing different elements to represent the nth level as a function of time. The aim of the approach is to convert the error due to element mismatch from a DC offset into an AC signal of equivalent power. In an oversampling converter this AC noise can be partially removed by filtering.

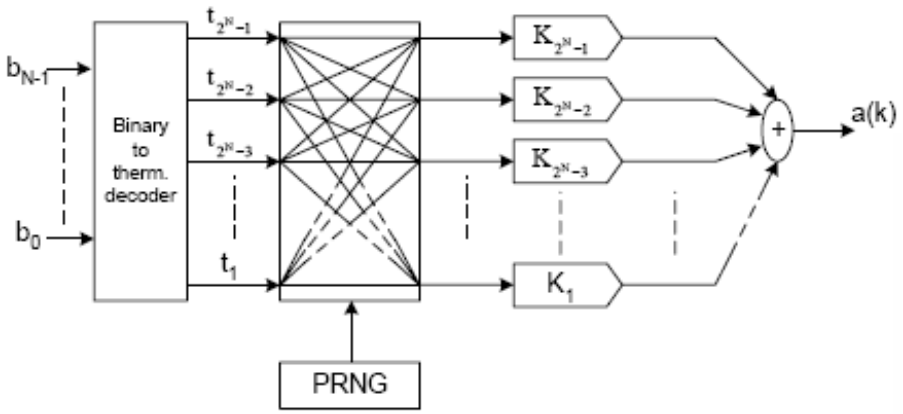

Figure 7.4. Element randomizer

A random selector can be implemented through a butterfly-network and a pseudorandom number generator which is implemented with linear feedback shift register in VHDL-AMS as seen in Figure 7.4.

The elements are randomly selected and the output will be a white noise source given by:

$$\begin{aligned} \mathrm{E}\left\{\varepsilon^2\right\} &= E\left\{(\sum_{i=0}^{d} K_i - d\cdot\hat{K})^2\right\} \\ &= E\left\{(\sum_{i=1}^{d} K_i - d\cdot\frac{\sum_{i=0}^{2^N-1} K_i}{2^N})^2\right\} \end{aligned} \tag{7.3}$$

$$= 2^N (\frac{d}{2^N}) \cdot (1 - \frac{d}{2^N}) \sigma_k{}^2$$

$$= d \cdot (1 - \frac{d}{2^N}) \sigma_k{}^2$$

If the noise is white and $\frac{1}{2 \cdot OSR}$ of the noise power falls within the base-band, a maximum signal swing of $-2^{N-1} V_{LSB}$ to $2^{N-1} V_{LSB}$ will result maximum SNR

$$SNR = \log_2 (\frac{\sqrt{\frac{2^N \cdot OSR}{3}}}{\sigma_\varepsilon}) \tag{7.4}$$

where σ_ε is standard deviation of unit elements used in thermometer code DAC.

In Table 7.2. element randomization outputs can be seen. On the left side of the Table, corresponding digital inputs for thermometer code decoder are also seen. Figure 7.5. exhibits PSD plots of ideal 3-bit DAC, thermometer code 3-bit DAC with 1% mismatch and randomization DEM 3-bit DAC. As it seen in Figure 7.5., there is a significant SNR improvement with the use of randomization DEM 3-bit DAC.

Table 7.1. Thermometer Code Decoder Outputs

d(0)=4	t0	t1	t2	t3	t4	t5	t6	t7
d(1)=4	t0	t1	t2	t3	t4	t5	t6	t7
d(2)=5	t0	t1	t2	t3	t4	t5	t6	t7
d(3)=5	t0	t1	t2	t3	t4	t5	t6	t7
d(4)=4	t0	t1	t2	t3	t4	t5	t6	t7
d(5)=5	t0	t1	t2	t3	t4	t5	t6	t7
d(6)=6	t0	t1	t2	t3	t4	t5	t6	t7
d(7)=6	t0	t1	t2	t3	t4	t5	t6	t7
d(8)=6	t0	t1	t2	t3	t4	t5	t6	t7
d(9)=5	t0	t1	t2	t3	t4	t5	t6	t7

Table 7.2. Element Randomization Outputs

d(0)=4	t0	t1	t2	t3	t4	t5	t6	t7
d(1)=4	t0	t1	t2	t3	t4	t5	t6	t7
d(2)=5	t0	t1	t2	t3	t4	t5	t6	t7
d(3)=5	t0	t1	t2	t3	t4	t5	t6	t7
d(4)=4	t0	t1	t2	t3	t4	t5	t6	t7
d(5)=5	t0	t1	t2	t3	t4	t5	t6	t7
d(6)=6	t0	t1	t2	t3	t4	t5	t6	t7
d(7)=6	t0	t1	t2	t3	t4	t5	t6	t7
d(8)=6	t0	t1	t2	t3	t4	t5	t6	t7
d(9)=5	t0	t1	t2	t3	t4	t5	t6	t7

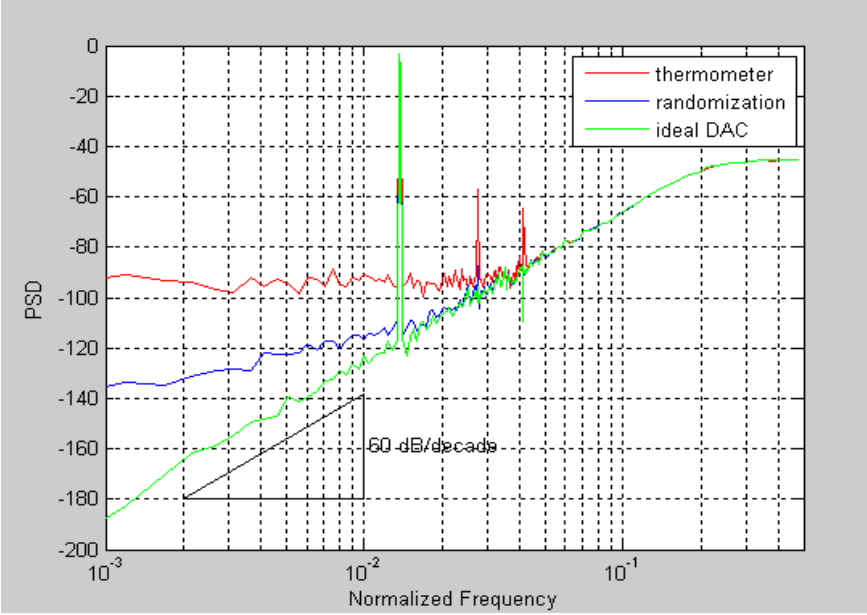

Figure 7.5. PSD of ideal, thermometer code, randomization DAC.

7.2.2. First Order Data Weighting Averaging

An improvement of element randomization came with the introduction of mismatch-shaping techniques in the early 90s, the Individual Level Averaging (ILA) algorithm from 1992 [19] and the Data Weighted Averaging (DWA) algorithm from 1995 [20] representing major breakthroughs. They were both based on the idea that if all elements

contribute equally, the INL would be cancelled, since the total error would then always have the same expectation value.

The principle of DWA is shown in Figure 7.6. If the first value $d(0)=2$, elements $t0..1$ are selected, if $d(1)=3$, elements $t3..4$ are selected next. As can be seen from the 3-bit example to the right in the figure, each element is used twice over the period. Of course, it will normally converge much slower.

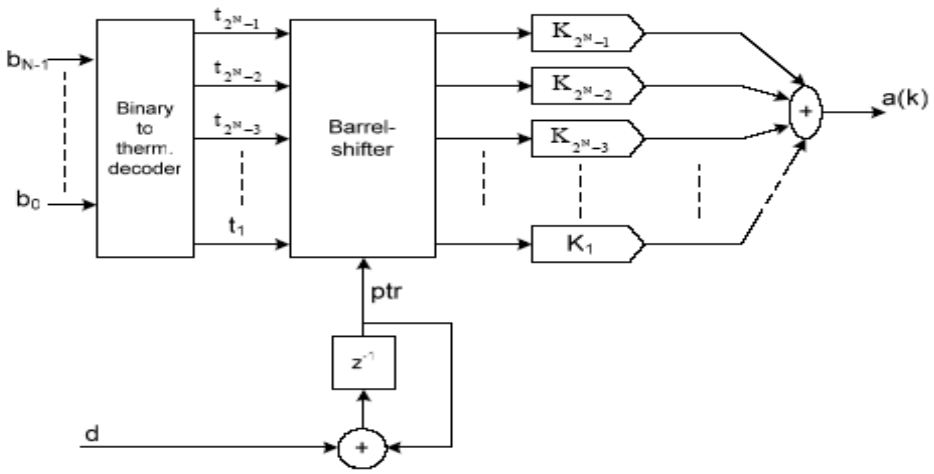

Figure 7.6. First Order Data Weighting Averaging

The shifting is controlled by a pointer that is incremented by the input value *d (k)* for each sample instant *k*. To be rotational it operates in modulo N which is 8 in our application.

$$ptr(k) = (ptr(k-1) + d(k)) \bmod N \tag{7.5}$$

The mismatch as a function of the time instant *k* is given by:

$$\varepsilon(k) = IM\left[ptr(k)\right] - IM\left[ptr(k-1)\right] \tag{7.6}$$

$$E(z) = (1 - z^{-1}) IM\left[PTR(z)\right] \tag{7.7}$$

If the base band resolution for maximum amplitude white input signal is:

$$SNR = \log_2\left(\frac{\sqrt{2^N OSR^3}}{\pi\sigma_\varepsilon\left(1-\frac{1}{2^N}\right)}\right) \tag{7.8}$$

where σ_ε is standard deviation of unit elements used in thermometer code DAC. In Table 7.3, first-order data weighting averaging outputs element can be seen. On the left side of the Table, corresponding digital inputs for thermometer code decoder are also seen.

Table 7.3. First Order Data Weighting Averaging Outputs

d(0)=4	t0	t1	t2	t3	t4	t5	t6	t7
d(1)=4	t0	t1	t2	t3	t4	t5	t6	t7
d(2)=5	t0	t1	t2	t3	t4	t5	t6	t7
d(3)=5	t0	t1	t2	t3	t4	t5	t6	t7
d(4)=4	t0	t1	t2	t3	t4	t5	t6	t7
d(5)=5	t0	t1	t2	t3	t4	t5	t6	t7
d(6)=6	t0	t1	t2	t3	t4	t5	t6	t7
d(7)=6	t0	t1	t2	t3	t4	t5	t6	t7
d(8)=6	t0	t1	t2	t3	t4	t5	t6	t7
d(9)=5	t0	t1	t2	t3	t4	t5	t6	t7

7.2.3. Second Order Data Weighting Averaging

The approach in this section can be generalized to encompass second order or arbitrary noise shaping. We want to achieve:

$$E(z) = H(z)\cdot IM\left[PTR(z)\right] \tag{7.9}$$

where $H(z)$ is a general noise shaping function given by

$$H(z) = \sum_{j=0}^{m} a_j \cdot z^{-j} \tag{7.10}$$

In time domain, $E(z)$ simply becomes:

$$\varepsilon(k) = \sum_{j=0}^{m} a_j \cdot IM\left[ptr(k-j)\right] \tag{7.11}$$

For $H(z) = 1 - z^{-1}$, we have already shown that this can be achieved through the single step

$$\begin{gathered} ptr(k) = (ptr(k-1) + d(k)) \bmod N. \\ \text{For } H(z) = (1 - z^{-1})^2 \end{gathered} \tag{7.12}$$

we get

$$\varepsilon(k) = IM\left[ptr(k) - 2IM\left[ptr(k-1)\right] + IM\left[ptr\left(k-2\right)\right]\right] \tag{7.13}$$

To achieve this, some elements, the elements up to the current pointer value must contribute with weight "1", the elements up to the previous pointer value must contribute with weight "-2" and the elements up to the pointer value two instances ago must contribute with weight "1". Since these are the same elements, each element's weight at any time instance is given by the sum of the three terms given by the pointer recursion. In the first order case it will always be "-1" or "1" (one and zero binary), and it by updated through a single assignment of each element. In the second order case the sum can grow to larger integers. Hence, to enable an element to contribute proportionally, *each element must be assigned several times in each sample instant*, making practical implementation impossible unless the DAC is run at several times its input sampling speed.

Table 7.4. Second Order Data Weighting Averaging

d(0)=4	t0	t1	t2	t3	t4	t5	t6	t7
d(1)=4	t0	t1	t2	t3	t4	t5	t6	t7
(2)=5	t0	t1	t2	t3	t4	t5	t6	t7
d(3)=5	t0	t1	t2	t3	t4	t5	t6	t7
d(4)=4	t0	t1	t2	t3	t4	t5	t6	t7
d(5)=5	t0	t1	t2	t3	t4	t5	t6	t7
d(6)=6	t0	t1	t2	t3	t4	t5	t6	t7
d(7)=6	t0	t1	t2	t3	t4	t5	t6	t7
d(8)=6	t0	t1	t2	t3	t4	t5	t6	t7
d(9)=5	t0	t1	t2	t3	t4	t5	t6	t7

In Table 7.4. second order data weighting averaging outputs can be seen. On the left side of the Table, corresponding digital inputs for thermometer code decoder are also seen. Figure 7.7. exhibits PSD plots of ideal 3-bit DAC, first and second order data weighting averaging 3-bit DACs. As it is seen Figure 7.7 second order data weighting averaging method introduces better SNR shaping

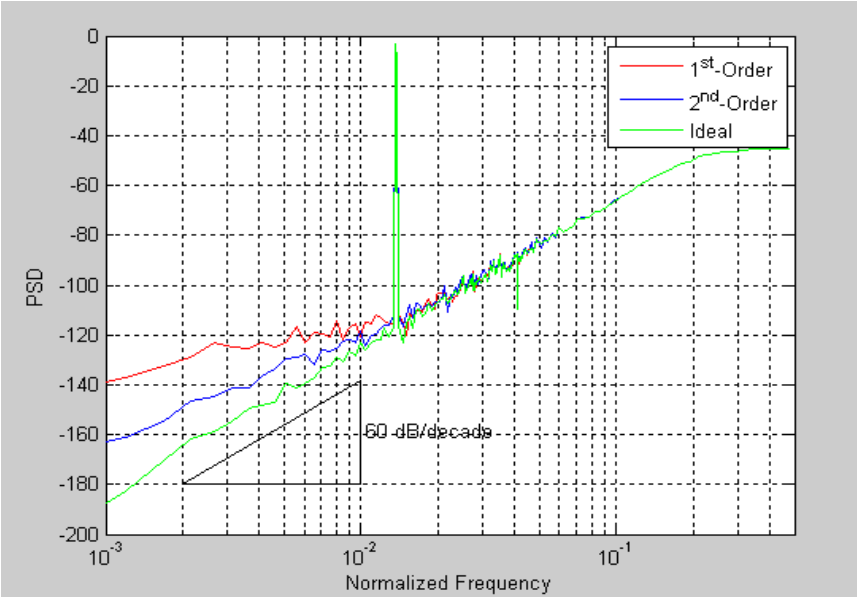

Figure 7.7. PSD of ideal, first order and second order DEM DAC

7.3. Integration of LMS Method with Second Order DEM

Up to now, two different digital error correction and calibration methods are mentioned: 1) LMS Adaptive Filtering explained in Section 6. 2) Second Order DEM explained in 7.2.3. In this section, integration of these two separate methods on Second Order 3-bit $\Sigma\Delta$ Modulator seen in Figure 7.8 will be explained. As it is seen in the figure, there are two different $\Sigma\Delta$ Modulators in the scheme just like in Figure 6.2. The main purpose of the LMS filter is to extract a desired signal, coming from the output of accurate from $\Sigma\Delta$ Modulator, a noise corrupted signal, coming from the output of inaccurate $\Sigma\Delta$ Modulator, by canceling the noise. On the other hand, 3-bit resistor-string DAC (see Figure 4.2) is integrated with second order DEM circuit to convert the error due to element mismatch from a DC offset into an AC signal of equivalent power.

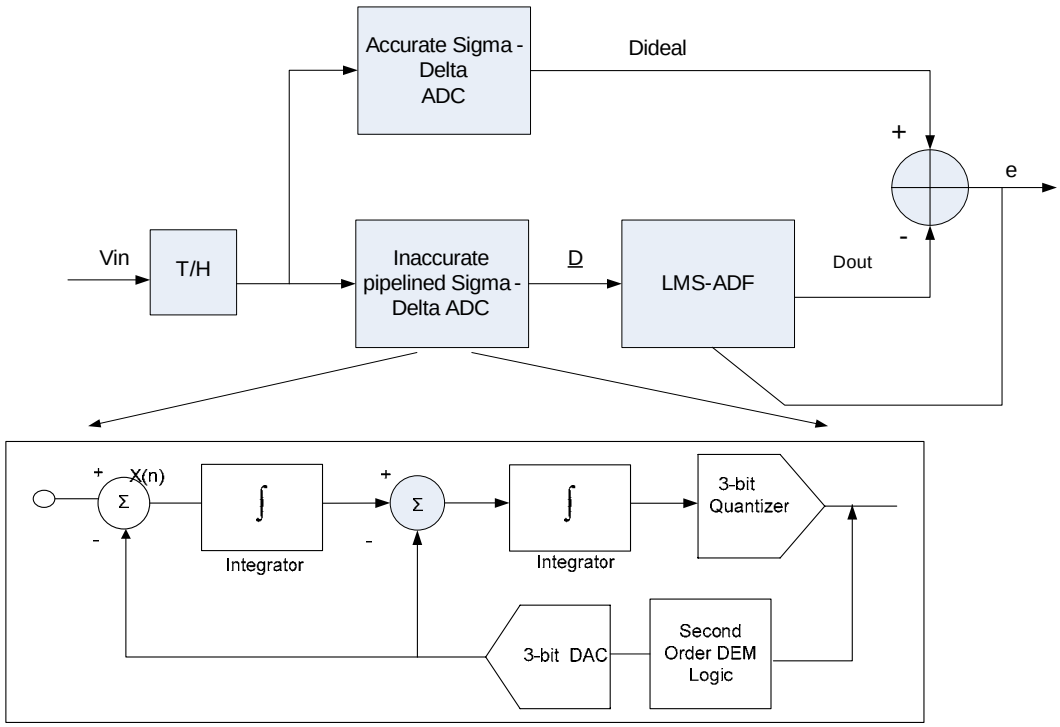

Figure 7.8. LMS Adaptive Filtering and DEM Method

Table 7.5. shows the input simulation parameters and Table 7.6 shows non-ideality parameters added to 3-bit Second Order $\Sigma\Delta$ Modulator. It is tried to evaluate the performance of FIR Adaptive Filter and second order DEM with real designs and production technologies. All simulation parameters and non-idealities are mentioned in previous sections except resistor or capacitor standard deviation (σ_ε).

Performance degradation is related to resistor standard deviation (σ_ε), (defined in 0.35 AMS technology) is given by:

$$\sigma_\varepsilon\left(\frac{\Delta R}{R}\right)=\frac{A_R}{\sqrt{W.L}} \tag{7.14}$$

Where A_R is unit resistor area, W is resistor width and L is resistor length. Capacitor standard deviation (defined in 0.35 AMS technology) that will be used in capacitor mismatch error modeling is also given by:

$$\sigma_{\varepsilon}\left(\frac{\Delta C}{C}\right)=\frac{A_C}{\sqrt{W.L}} \tag{7.15}$$

where A_C is capacitor area, W is capacitor width and L capacitor is length.

Table 7.5. Simulation Parameters

ΔΣ Modulator Parameters	**Value**
Signal bandwidth	BW=62500 Hz
Oversampling frequecy	Fs=4 Mhz
Oversampling ration	OSR=32
Integrator gains	g1=g2=0.5
Finite gain bandwidth	1 kHz
Slew Rate	SR=10V/us
Quantizer Stage	3

Table 7.6. Non-ideality parameters

Non-idealities	**Value**
Sampling Cap. Mismatch	0.09% [21]
DAC Resistor Mismatch	0.09% [21]
Op-Amp DC gain leakage	0.99
Op-amp input-referred offset	10% Vref
Comparator offset	10% Vref
Comparator Noise	1% Vref

Figure 7.9. shows the PSD of Ideal ΣΔ Modulator (SNR 65 dB), while Figure 7.10. shows PSD of ΣΔ Modulator including the non-idealities in Table 7.6. (SNR 52 dB). As it is seen from these two figures, there is 13 dB SNR decrease due to the sampling capacitor mismatch, DAC Resistor Mismatch, Op-Amp DE gain leakage, Op-amp input-referred offset and comparator offset. Figure 7.11. shows ideal case (blue), non-ideal case (red) and enhanced ΣΔ Modulator (green) with LMS Adaptive Filtering and second order DEM (SNR 63 dB).

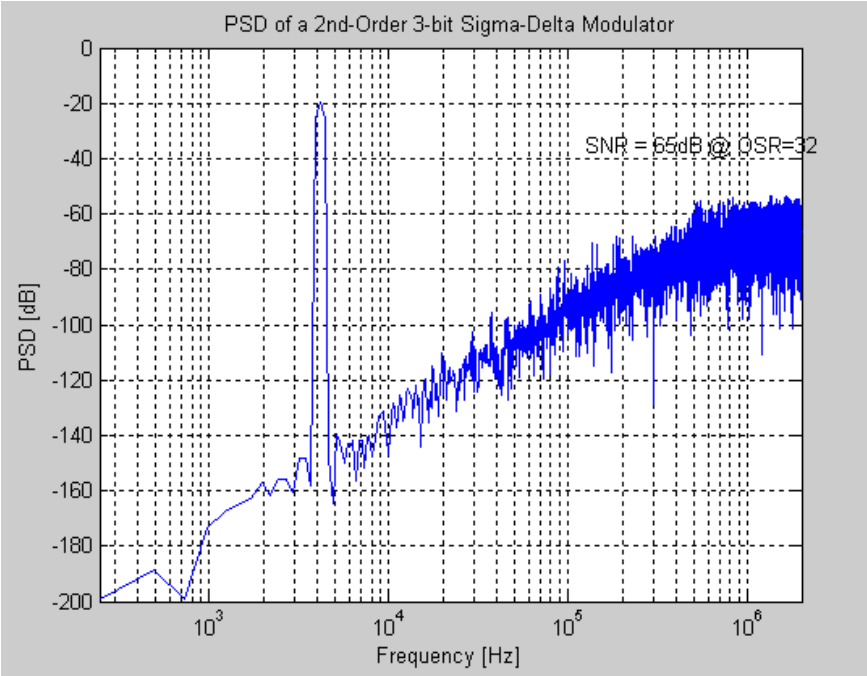

Figure 7.9. PSD of Ideal ΣΔ Modulator

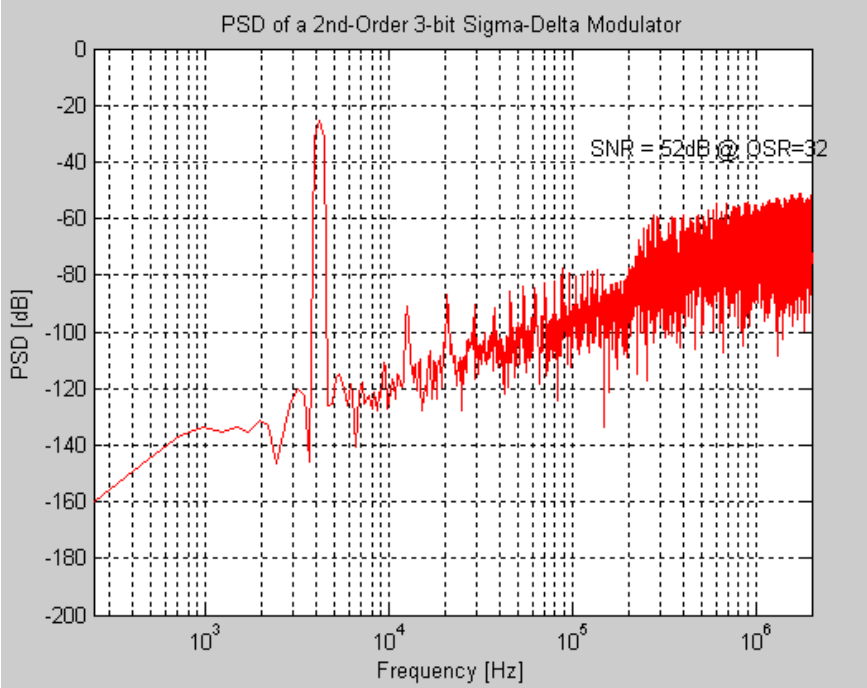

Figure 7.10. PSD of ΣΔ Modulator with Non-idealities in Table 7.6.

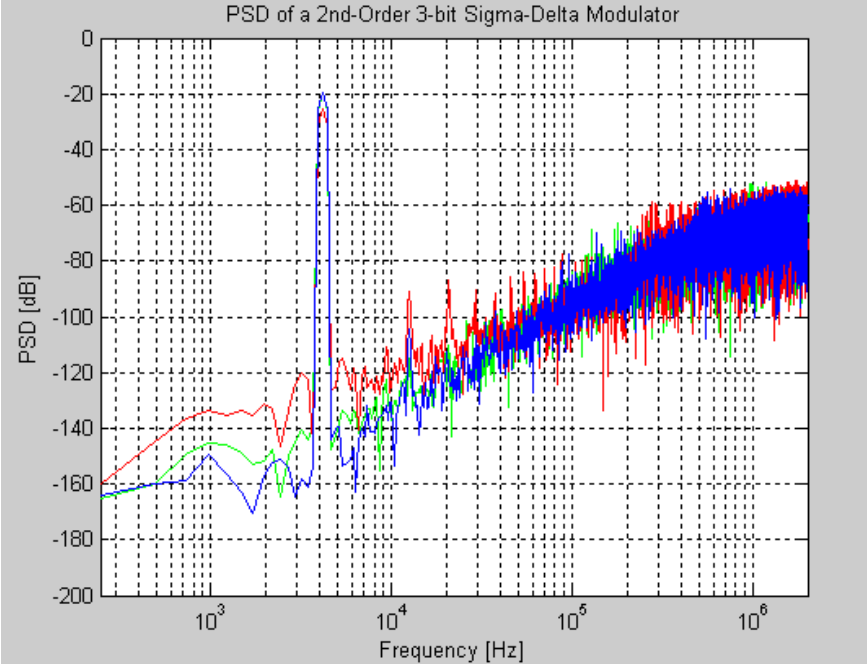

Figure 7.11. PSD of ideal, non-ideal,enhanced ΣΔ Modulator

8. CONCLUSION AND FUTURE WORK

8.1. Conclusion

Sigma Delta (ΣΔ) Modulators, introduced by Inose in 1962, are the most suitable A/D converter topologies for digitizing with high resolution analog signals characterized by a bandwidth much smaller than the sampling frequency fs. The output of ΣΔ modulators is a coarse quantization of the analog input and these converters require digital filter circuitry to remove out-of-band quantization noise and to achieve sample rate reduction. In the first part of my thesis, two different types (polyphase FIR and CIC decimation) of accurate and generic VHDL implementation of low pass finite impulse response (FIR) and decimation filters for ΣΔ converters are performed. As it is mentioned in Section 3, many parameters such as number of taps, input/output world lengths, and filter coefficients for each tap are defined as generic type to ensure effective and easy integration of these decimation filters with Sigma Delta (ΣΔ) Modulators or any other digital blocks.

As it is explained in Section 4 in detail, ΣΔ Modulators' non-idealities limit their overall performance. Simulation models play an important role in the design of ΣΔM and high level languages such as Matlab and C have been commonly used for modeling them. Another language, VHDL-AMS, - an extension of VHDL introduced in 1999) give designers the chance of hierarchical description and simulation of discrete, continuous and mixed systems with conservative and non-conservative semantics. With using VHDL-AMS, behavioral and structural modeling of ΣΔM is performed and these simulation outputs are used in the design of digital error correction and calibration blocks.

Even though, ΣΔ Conversion is intrinsically less sensitive to non-idealities than other A/D conversion techniques [1]; inherent non-idealities in ΣΔ building blocks limit the overall performance of these converters and they usually suffer from both static errors that distort the overall transfer function, resolution and linearity of ΣΔ Modulators. In my thesis, a LMS based background self calibration technique has been implemented to overcome static errors in ΣΔ Modulators without stopping input conversion. Section 6

gives a detailed description of this error correction filter. In Section 6.2.1 and 6.2.2, performance analysis of LMS based error correction and calibration filter is performed in Matlab Simulink and VHDL-AMS. Finally, it is stated that second order ΣΔMs' static errors such as capacitor mismatch, finite op-amp gain and non-linearity, switch-induced charge injection can be enhanced extensively with the help of this filter.

The error correction and calibration filter in Section 6 was designed for 1-bit output pipelined ΣΔMs. On the other hand, there is a requirement to develop an error correction and calibration schemes for multi-bit modulators; because, they are extensively insensitive to the feedback D/A converter (DAC) non-idealities. High DAC linearity requires precise matching of the DAC unit elements that are typically capacitors and current sources. To overcome this trade-off, second order DEM is implemented as a digital block that can easily be integrated with any N bit Unit Element DACs. As it is stated in Section 7.3, integration of this DEM scheme with LMS filter is also done. The performance analysis of this integrated scheme is performed on 3-bit second order ΣΔM and significant SNR improvements are achieved.

8.2. Future Work

Polphase FIR and CIC decimation filters were designed as generic as possible to remove out-of-band quantization noise and to achieve sample rate reduction of ΣΔMs. As a future work, different digital filter topologies that can enhance offset errors before LMS Adaptive Filtering will be studied. I am sure that, these studies will bring us more effective and less costly filter topologies for FIR Decimation and LMS Adaptive Filtering.

Finally, the proposed system can be designed and produced as an IC to measure its real time performance.

REFERENCES

1. Norsworthy, S. R., R. Schreier and G. C. Temes, *Delta-Sigma Data Converters: Theory, Design, and Simulation,* Wiley-IEEE Press. New York, 1997.

2. Chiu,Y., C. W. Tsang, B. Nikolic and P. R. Gray, "Least mean square adaptive digital background calibration of pipelined analog-to-digital converters", *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications,* Volume 51, Issue 1, Page(s): 38 – 46, Jan. 2004.

3. Hamoui, A. A., K. Martin, **"**Linearity enhancement of multibit /spl Delta//spl Sigma/ modulators using pseudo data-weighted averaging", *ISCAS 2002,* Volume 3, Page(s): III-285 - III-288, 2002.

4. Oppenheim, A. V., *Signals and Systems*, Prentice Hall; 2nd edition, August 6, 1996.

5. Chou, W., P. Wong and R. Gray, "Multi-Stage Sigma-Delta Modulation", *IEEE Transactions on Information Theory,* Vol.35, pp.784-796, July 1989.

6. Longo, L. and M. A. Copeland, "A 13-bit ISDN-band ADC using Two-Stage Third Order Noise Shaping", *IEEE Custom Integrated Circuit Conference,* pp.21.2.1-4, June 1988.

7. Sarhand-Nejad, M. and G.C.Temes, "A high-resolution multibit sigma-delta ADC with digital correction and relaxed amplifier requirements", *IEEE Journal of Solid-State Circuits,* Vol.28, No.6, pp.648-660, June 1993.

8. Hogenauer, E. **"**An economical class of digital filters for decimation and interpolation", *IEEE Transactions on Signal Processing,* Volume 29, Issue 2, Page(s): 155 – 162, Apr 1981.

9. Boser,B.E. and B.A.Wooley, “The design of sigma-delta modulation analog-to-digital converters”, *IEEE Journal of Solid State Circuits*, Vol.23, No.6, pp.1298-1308, December 1988.

10. Fernando, M., A. Pérez-Verdú and A. Rodríguez-Vázquez, *Top-down design high-performance sigma-delta modulators*, Imprint Boston:Kluwer Academic Publishers, c1999.

11. Malcovati, P., S. Brigati, F. Francesconi, F. Maloberti, P. Cusinato and A. Baschirotto, “Behavioral Modeling of Switched-Capacitor Sigma-Delta Modulators”, *IEEE Transactions on Circuits and Systems*, Vol. 50, Issue 3, March 2005.

12. Koe, W. M and J. Zhang, “Understanding the effect of circuit non-idealities on sigma-delta modulator”, University of Texas at Dallas, 2002.

13. Haykin, S. S., *Adaptive Filter Theory,* Third Edition, Prentice Hall, 1996.

14. Larsson, A. and Sonkusale, S., “A Background Calibration Scheme for Pipelined ADC Including Non-linear Operational Amplifier Gain Correction and Reference Error Compensation”, *IEEE System on Chip Conference,* 2004.

15. Sarhand-Nejad, J. and G. C. Temes, “A high-resolution multibit sigma-delta ADC with digital correction and relaxed amplifier requirements”, *IEEE Journal of Solid-State Circuits,* Vol.28, No.6, pp.648-660, June 1993.

16. Baird, R. T. and T. S. Fiez, “A low oversampling ratio 14-b delta-sigma ADC with self- calibrated multibit DAC”, *IEEE Journal Solid-State Circuits,* Vol.31, No.3, pp.312-320, 1996.

17. Petrie, C. and M. Miller, “A background calibration technique for multibit delta-sigma modulators”, *ISCAS 2000*, pp,29-32, May 2000.

18. Wang, X. , P. Kiss, U. Moon, J. Steensgaard and G. C. Temes, "Digital estimation and correction of DAC errors in multibit ΣΔ ADCs", *IEEE Electronics Letters,* vol.37, no.7,pp.414-415, May 2001.

19. Leung, B. H. and S. Sutarja: "Multibit Sigma-Delta A/D Converter Incorporating A Novel Class of Dynamic Element Matching Techniques", *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing,* Vol. 39, No. 1, pp. 35-51, 1992.

20. Baird, R. T. and T. S. Fiez, "Linearity enhancement of multi-bit Δ-Σ A/D and D/A converters using data weighted averaging", *IEEE Circuits & Systems 11, CASII-42,* pp. 753-762, December 1995.

21. Linares-Barranco, B., T. Serrano-Gotarredona, R. Serrano-Gotarredona, G. Vicente-Sanchez, "On mismatch properties of MOS and resistors calibrated ladder structures", *ISCAS 2004 ,* Volume 1, Issue, pp. I-377 - I-380, 23-26 May 2004.