NOISE-SHAPING STOCHASTIC OPTIMIZATION AND ONLINE LEARNING WITH APPLICATIONS TO DIGITALLY-ASSISTED ANALOG CIRCUITS

By

Ravi Krishna Shaga

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Electrical Engineering

2011

ABSTRACT

NOISE-SHAPING STOCHASTIC OPTIMIZATION AND ONLINE LEARNING WITH APPLICATIONS TO DIGITALLY-ASSISTED ANALOG CIRCUITS

By

Ravi Krishna Shaga

Analog circuits that use on-chip digital-to-analog converters for calibration use DSP based algorithms for optimizing and calibrating the system parameters. However, the performance of traditional online-gradient descent based optimization and calibration algorithms suffer from artifacts due to quantization noise which adversely affects the real-time and precise convergence to the desired parameters. This thesis proposes and analyzes a novel class of on-line learning algorithms that can noise-shape the effect of quantization noise during the adaptation procedure and in the process achieve faster spectral convergence compared to the conventional quantized gradient-descent approach. We extend the proposed framework to higher-order noise-shaping and derive criteria for achieving optimal system performance. The thesis also explores the application of stochastic perturbative gradient descent techniques to the proposed noise-shaping online learning framework where we show the performance of the stochastic algorithm can be improved in the spectral domain.

The thesis applies the proposed optimization method for online calibration of subthreshold analog circuits where artifacts like mismatch and non-linearity are more pronounced. Using measured results obtained from prototype fabricated in a 0.5μm CMOS process, we demonstrate the robustness of the proposed algorithm for the task of: (a) compensating and tracking of offset parameters; and (b) calibration of the center frequency of a sub-threshold gm-C biquad filter.

Dedicated to my parents, Narsaiah and Pushpa Latha,
for their unconditional love and support

ACKNOWLEDGMENT

The completion of my graduate studies has been one of the most significant academic challenges that I have undertaken. This endeavour wouldnot have been possible without the help and support of several people. I wish to express my gratitude to all of them.

First and foremost, I would like to take this opportunity to express my deepest appreciation to Prof. Shantanu Chakrabartty for his support, patience and guidance during my graduate studies. As an academic advisor, he was always approachable and ready to help in every way possible. My special thanks to him for helping me in some of my difficult times both academically and personally. I would also like to thank Prof. Jack R. Deller and Prof. Selin Aviyente for being on my MS thesis committee and guiding me through the process.

I would like to extend my special thanks to my labmates Chenling Huang, Amin Fazel, Ming Gu, Amit Gore and Yang Liu with whom I have spent considerable amount of time and I am grateful for their invaluable guidance during the early stages of my research.

I have been fortunate enough to be in the company of a lot of good friends who made my journey through the graduate school really memorable. They were indeed like a family to me over here helping me in every step of the way.

Atlast, I owe my deepest regards and gratitude to my parents Narsaiah and Pushpa Latha and my brothers Murali and Shivaram for their unconditional love and support which has carried me to where I am today.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Optimization problems in real life are often dynamic in nature, i.e., they keep changing with time. Therefore there is a need to continuously monitor the system and adapt to the variations so that the system performance can be maximized or the losses minimized. From a system level perspective, the block diagram of a generic online learning system is given in Fig 1.1. The system is characterized by a performance index L called the 'cost function' or the 'loss function' which can usually be controlled using a set of adjustable parameters θ. An online learning system makes use of the previous experience of system output and tries to adapt to the changing input by learning and correcting itself accordingly. The objective of the learning system is to be able to predict the output of the system when a previously unseen input stimuli occurs.

A learning algorithm can usually be modelled as an optimization problem where the set of adjustable parameters θ need to be adapted in a way such that the loss function L is minimized. Several different class of optimization algorithms are present in literature [1] [2]. The choice of the optimization method being used depends on the system characteristics,

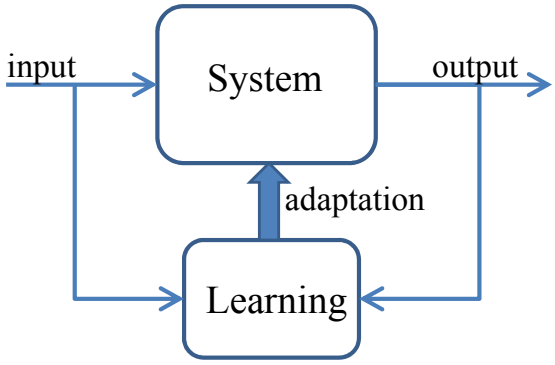


Figure 1.1: *Schematic diagram of a generic online learning system*

the performance index of the system and the constraints of optimization. Zeroth order optimization algorithms like direct search algorithms make use of the direct measurements of the loss function by moving the system adaptation parameter in the direction in which the loss function decreases. First order optimization algorithms [3] depend on the first order gradient measurement of the loss function to adapt the system parameters in the direction of the steepest gradient descent. Second order optimization algorithms like Newtons method make use of the information present in the hessian of the loss function to adapt the optimization parameter. As the order of optimization algorithm increases, the rate of convergence increases. But the requirement of the knowledge of higher order gradients puts an additional cost on the optimization algorithms of higher order. Several other heuristic and stochastic optimization techniques have been proposed in literature including genetic algorithm optimization [4], simulated annealing [5], particle swarm optimization [6] and reactive search algorithms [7] [8]. One of the most popular optimization techniques is the gradient descent learning algorithm. It makes use of the fact that the loss function decreases rapidly along the direction of the slope, imitating a greedy search algorithm in following the steepest descent. One of the major drawback of the gradient descent algorithm is that it

converges to the local minimum point for small enough adaptation rate. So, the optimization process is highly susceptible to the choice of the intial point used for the iterative updates. Digitally assisted analog circuits make use of a A/D conversion step in the calibration of the analog circuits. A conventional A/D converter like nyquist rate converters adds an additional quantization noise to the system making the adaptation process susceptible to noise. This results in a dramatic decrease in the performance of the learning algorithm with respect to the convergence rate.

In this thesis, we propose a novel $\Sigma\Delta$ gradient-descent optimization algorithm which can be used for online tracking of system parameters in real-time. The proposed algorithm has superior performance as compared to the quantized gradient-descent algorithm as the noise-shaping characteristics of the $\Sigma\Delta$ modulator suppress the inband quantization noise and pushes it away into the higher frequency range. The out of band quantization noise can be eliminated by using an appropriate lowpass filter. Also, if the input signal is sufficiently random, the quantization stage of the modulator can be modeled as an additive white-noise with flat power spectrum density. The randomness in the quantization noise actually helps the optimization process by allowing the optimization parameter out of local minimum neighborhood depending on the noise floor level. This enables a larger search space for the optimization process. Although the global optimization is not guaranteed, the algorithm is more robust than the ideal gradient descent which is known to converge to the local minimum.

The remaining part of the thesis is organized as follows. In Chapter 2, mathematical framework for the quantized version of a 1^{st} order optimization algorithm using gradient descent approach has been introduced. A novel $\Sigma\Delta$ gradient descent algorithm has been

proposed which makes use of the noise-shaping properties of the A/D modulator to achieve better convergence properties. The effect of the adaptation parameter on the error rate has been thoroughly studied. The performance enhancement of the proposed algorithm has been validated through simulations on a speech sample from YOHO database. Chapter 3 generalizes the proposed algorithm to include a higher order loop filter and learning mechanism. Chapter 4 deals with gradient-free optimization algorithms (Finite-difference and Simultaneous perturbation techniques) and the extension of the noise-shaping algorithm to these cases. In Chapter 5, we introduce the practical problems (non-linearities, mismatches, temperature variance) in the calibration of analog circuits operating in the subthreshold region of operation. A single channel analog feature extractor implemented using bandpass filter is studied. The $\Sigma\Delta$ learning algorithm is validated on hardware by (a) calibrating an unbalanced first-order $\Sigma\Delta$ modulator for offset cancellation and (b) calibrating the center frequency of a gm-C bandpass filter. Chapter 6 concludes the thesis with some final remarks and observations.

Chapter 2

Gradient Descent Optimization

2.1 Optimization problem

Consider a system where a stochastic parameter X introduces randomness in the measurement of the objective function $L(X;\theta)$. Let θ be an adjustable parameter which can be varied as a response to the change in parameter X . The optimization problem refers to finding the optimal point θ^* where the loss function given by $L(X;\theta)$ goes to a minimum, i.e.,

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, L(X;\theta) \tag{2.1}$$

Assuming that $L(X;\theta)$ is continuous and differentiable w.r.t θ, the partial derivative $g(\theta)$ is given by

$$g(\theta) = \frac{\partial L(X;\theta)}{\partial \theta} \tag{2.2}$$

For the remainder of this chapter, the analysis assumes that the gradient of the loss function exists and is readily available. The case when the gradient is not available or $L(\theta)$ is not

differentiable is discussed in Chapter 4. Also, the performance analysis is studied for a single variable system but can be readily expanded to a multi-dimensional optimization problem with $\theta \in \mathcal{R}^p$ by using a vectorial representation.

2.1.1 Gradient Descent Algorithm

One of the most popular technique for optimization of the above problem is the gradient descent algorithm, where at each step, the parameter θ is decreased in the direction of the slope of the loss function $L(X;\theta)$. The iterative update

$$\theta_n = \theta_{n-1} - \epsilon g(\theta_{n-1}) \tag{2.3}$$

converges the loss function to the minimum point given in eq.(2.1). Taking the summation of the above equation to ∞, the optimal point θ^* can be obtained to be

$$\theta^* = -\epsilon \sum_{k=1}^{\infty} g(\theta_k) \tag{2.4}$$

The effect of the choice of the learning rate ϵ on the convergence of the gradient descent algorithm has been extensively studied in [9],[3]. If the choice of ϵ is too small, the optimization takes longer time to converge. On the other hand, if the learning rate is too high, the convergence is effected as the solution oscillates about the optima point. From Taylor's series first order approximation of $g(\theta)$ about the optima point θ^*, we get

$$g(\theta) = g(\theta^*) + (\theta - \theta^*)g^{'}(\theta) \tag{2.5}$$

But for the optimal point θ^*, $g(\theta^*) = 0$. Comparing the above equation with eq. (2.3), we have

$$\epsilon_n = \frac{1}{g^{'}(\theta)} \tag{2.6}$$

which shows that the optimal learning rate is given by the inverse of the hessian of the loss function.

One of the major drawbacks of the gradient descent approach is that depending on the initial choice of the parameter θ, the algorithm converges to local minimum even in the presence of a global minimum. It has been observed that by careful injection of noise to the learning algorithm, the standard gradient descent converges to the global optimal point. The idea behind deliberately adding noise comes from the fact that the stochastic nature of the noise in the recursion would allow the algorithm to escape the θ neighborhood corresponding to the local minimum.

2.1.2 Noisy Gradient Descent Algorithm

When an intentional noise q_n is added to the gradient estimate, the gradient descent algorithm takes the form:

$$\theta_n \leftarrow \theta_{n-1} - \epsilon[g(\theta_{n-1}) + q_n] \tag{2.7}$$

Taking the summation of the above equation to ∞, the function $L(\theta)$ converges to its minimum value at θ^* given by:

$$\theta^* = -\epsilon(\sum_{k=1}^{\infty} g(\theta_k) + \sum_{k=1}^{\infty} q_k) \tag{2.8}$$

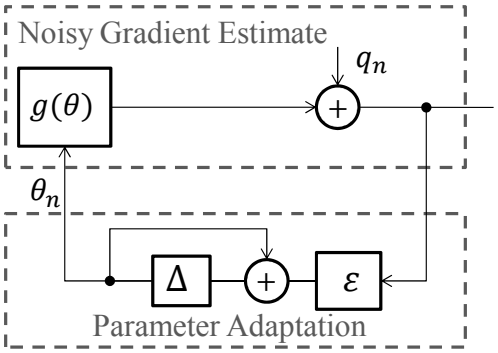


Figure 2.1: *Block diagram of the noisy gradient descent optimization algorithm*

We consider the case when q_n is additive white noise with a flat power spectral density. For a zero mean random white noise, the second term in eq.(2.8) goes to zero, thereby converging to the same optimal point as in the case of the noiseless gradient descent algorithm in eq. (2.4). The conditions for convergence of the above algorithm have been studied in [13] and [16], which prove the *almost sure* convergence of the algorithm when q_n follows certain statistical properties.

Fig. 2.1 shows the block diagram of the noisy gradient descent algorithm. Depending on the choice of learning parameter, the final value of θ oscillates randomly about the optimal point θ^*. In order to reduce the effect of the noise on the optimization, a simulated annealing technique([11],[12]) has been studied in literature. A large value of the adaptation parameter ϵ is initially choosen and is successively damped allowing the initial iterations to move θ out of the local minimum neighborhood. As the number of iterations is increased, the adaptation parameter goes to zero thereby decreasing the effect of the additive noise. The choice of the noise floor level and annealing rate depends on the loss function that is to be minimized. Care has to be taken to make sure that the adaptation parameter ϵ doesnot die down to zero before reaching the global minimum point.The conditions for global minimum optimization of stochastic gradient descent algorithm has been studied in [10]-[12].

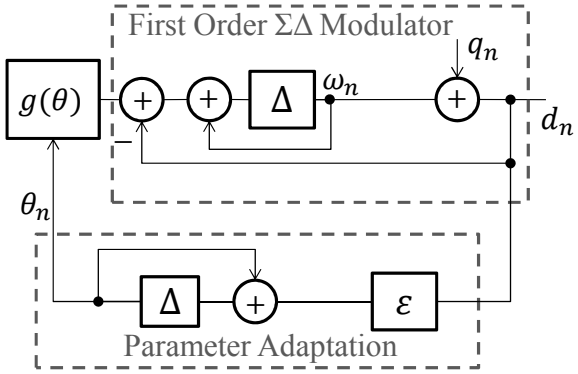


Figure 2.2: *Block diagram of the first order noise-shaping gradient descent algorithm*

2.1.3 Noise Shaping Gradient Descent Algorithm

In this thesis, we propose a new optimization algorithm which makes use of the noise shaping properties of a $\Sigma\Delta$ modulator in finding the optima point. The iterative updates for the optimization of θ through this new algorithm are given by:

$$\omega_n = \omega_{n-1} + \big(g(\theta_{n-1}) - d_n\big) \tag{2.9}$$

$$d_n = \omega_{n-1} + q_n \tag{2.10}$$

$$\theta_n = \theta_{n-1} - \epsilon d_n \tag{2.11}$$

The proposed algorithm has better convergence properties compared to the noisy gradient-descent algorithm. The architecture introduces a $\Sigma\Delta$ loop which shapes the quantization noise q_n before being used in the gradient-descent iteration. Fig.2.2 shows the block diagram of the first order noise-shaped gradient descent algorithm.

For a large number of iterations N, by taking the summation of eq. (2.11), we get

$$\theta_N = \theta_0 - \epsilon \sum_{k=1}^{N} d_k \tag{2.12}$$

But from the $\Sigma\Delta$ update of eqn.(2.9), if the intial state is ω_0, the summation to N iterations yields

$$\sum_{k=1}^{N} d_k = \sum_{k=1}^{N} g(\theta_k) + \omega_N - \omega_0 \tag{2.13}$$

As N tends to ∞, from the stability considerations of a first order $\Sigma\Delta$ modulator we get

$$|\sum_{k=1}^{\infty} d_k - \sum_{k=1}^{\infty} g(\theta_k)| \leq 2\|g(\theta)\|_\infty \tag{2.14}$$

which shows the asymptotic convergence of the algorithm to the optimal point θ^*.

We compare the performance of the above optimization to that of the noisy gradient descent by studying the convergence properties of both the algorithms in the case of a signal tracking problem. Consider the loss function in the case of the least mean square optimization problem given by

$$L(t;\theta) = \lim_{T\to\infty} \frac{1}{T} \int_0^T \frac{1}{2} (y(t) - \theta)^2 \, dt \tag{2.15}$$

where the signal that is to be tracked is a sinusoid, i.e., $y(t) = \sin(\omega t)$. Fig.2.3 shows the noise-shaping characteristics of the proposed algorithm which leads to better convergence properties compared to the noisy gradient descent learning algorithm. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this thesis. In this simulation, the gradient of the loss function $g(\theta)$ is assumed to be available for computation. The value of $\epsilon = 0.5$ is chosen for the simulation. The

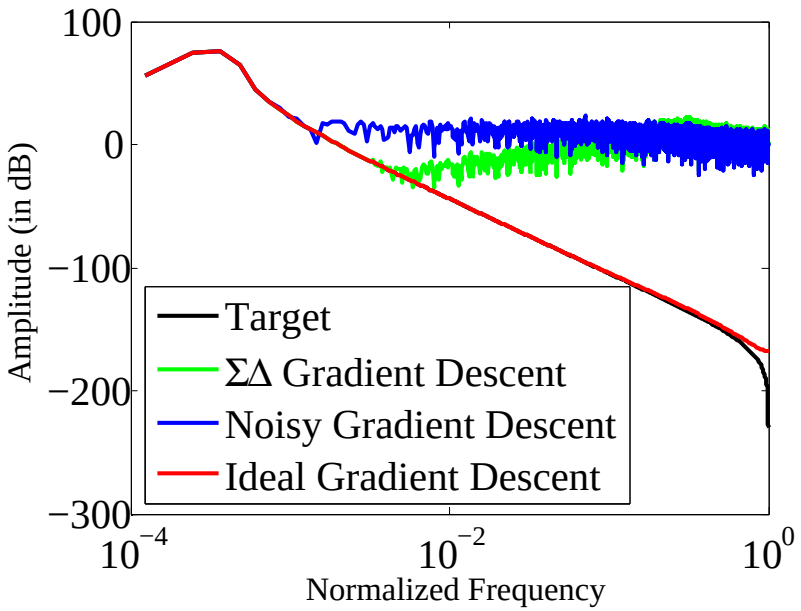


Figure 2.3: *Frequency domain response of the original sinusoid, ideal, noisy and noise-shaped gradient descent algorithms. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this thesis.*

magnitude response of the noisy gradient descent algorithm shows that the noise floor of about 5dB corrupts the tracked signal θ. But in the case of $\Sigma\Delta$ modulator, the noise at lower frequencies is clearly shaped out of the signal band of interest. By using a suitable lowpass filter, the high frequency noise can be eliminated from the output.

The proposed $\Sigma\Delta$ algorithm is more robust to traps introduced by local minimum neighborhood. Fig 2.4 shows the convergence of the algorithm to the global minimum even when the ideal gradient descent algorithm converges to local minimum. The cost function under consideration is $L(\theta) = -e^{-\theta}\frac{sin(\theta^4)}{\theta^3}$. The same initial point $\theta_0 = 1.45$ is used for both the ideal and the $\Sigma\Delta$ gradient descent algorithm. The highly non-linear form of the loss function causes the ideal gradient descent optimization to converge to the local minimum. Whereas the addition of the noise q_n enables the $\Sigma\Delta$ approach to avoid the local minimum and converge to the global optimal solution. The global convergence of the algorithm is not

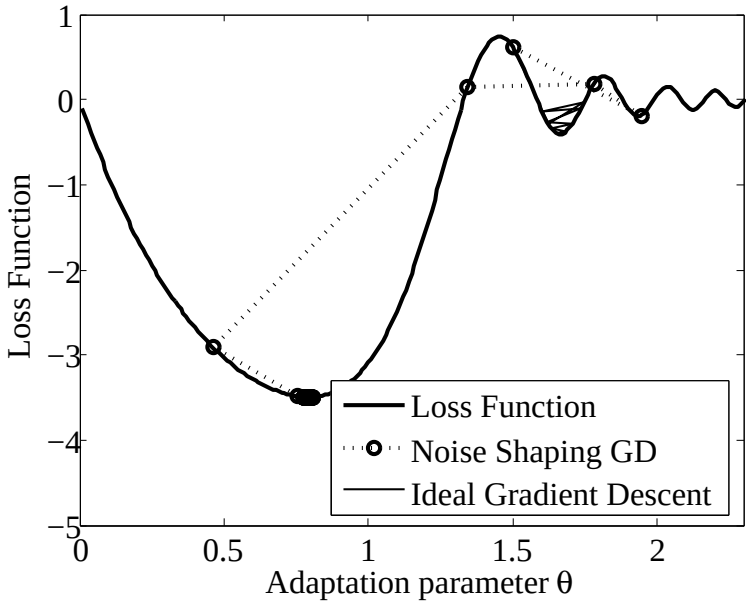


Figure 2.4: *Convergence of Noise shaping gradient descent algorithm to the global minimum*

guaranteed, and depends highly on the loss function under consideration and the initial conditions. But the addition of the noise randomizes the search process enabling more robust global search properties.

Fig. 2.5 shows the real-time tracking performance of the algorithms with $\epsilon = 0.9$. It can be clearly seen that the $\Sigma\Delta$ learning has larger higher frequency spikes compared to the noisy gradient-descent case showing that q_n is shaped. But at lower frequencies, the $\Sigma\Delta$ algorithm output follows the input signal more closely compared to the noisy gradient-descent case.

Fig. 2.6 shows that because of the addition of noise to the learning algorithm, there is no penalty w.r.t the speed of convergence to the optimal point. The loss function under consideration here is the same as in eq. (2.15), where $y(t) = 1$, a constant function.

Fig. 2.7 shows that the convergence of the noisy gradient descent algorithm depends greatly on the noise floor of q_n being added. As the magnitude of the noise level q_n increases,

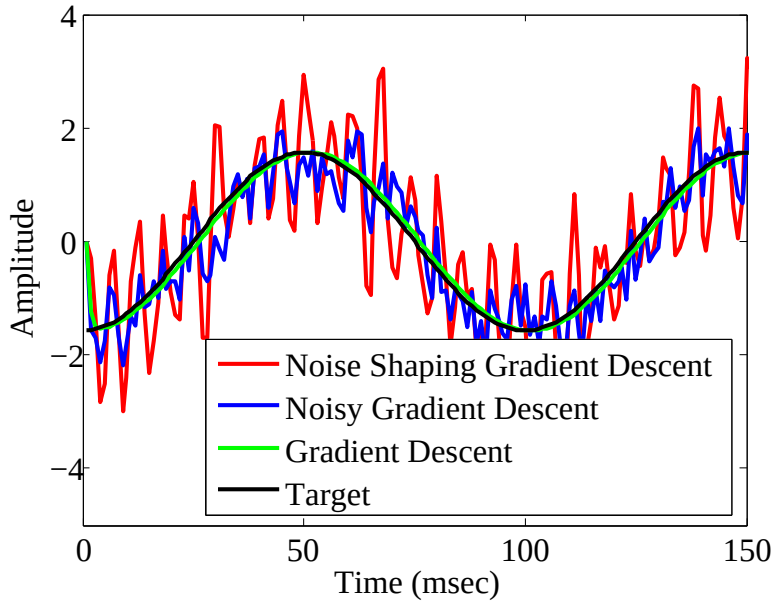


Figure 2.5: *Tracking a sinusoid signal with ideal,noisy and ΣΔ gradient descent algorithms*

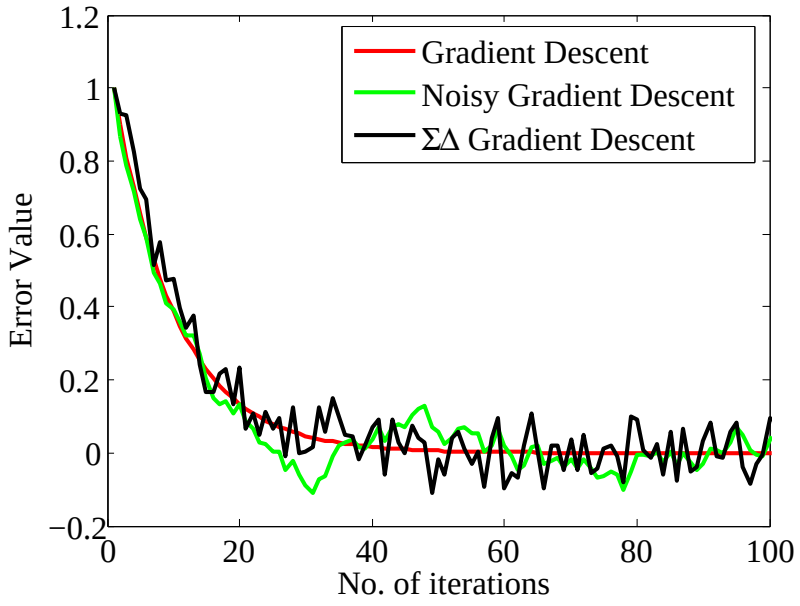


Figure 2.6: *Rate of convergence of the ideal, noisy and ΣΔ gradient descent algorithms*

the error in the estimation of θ increases. Whereas the highpass filter characteristics of the noise shaping in ΣΔ learning reduces the noise q_n in the signal band of interest, thereby improving the convergence. Fig. 2.7(a) and (b) shows that as the noise floor level increases

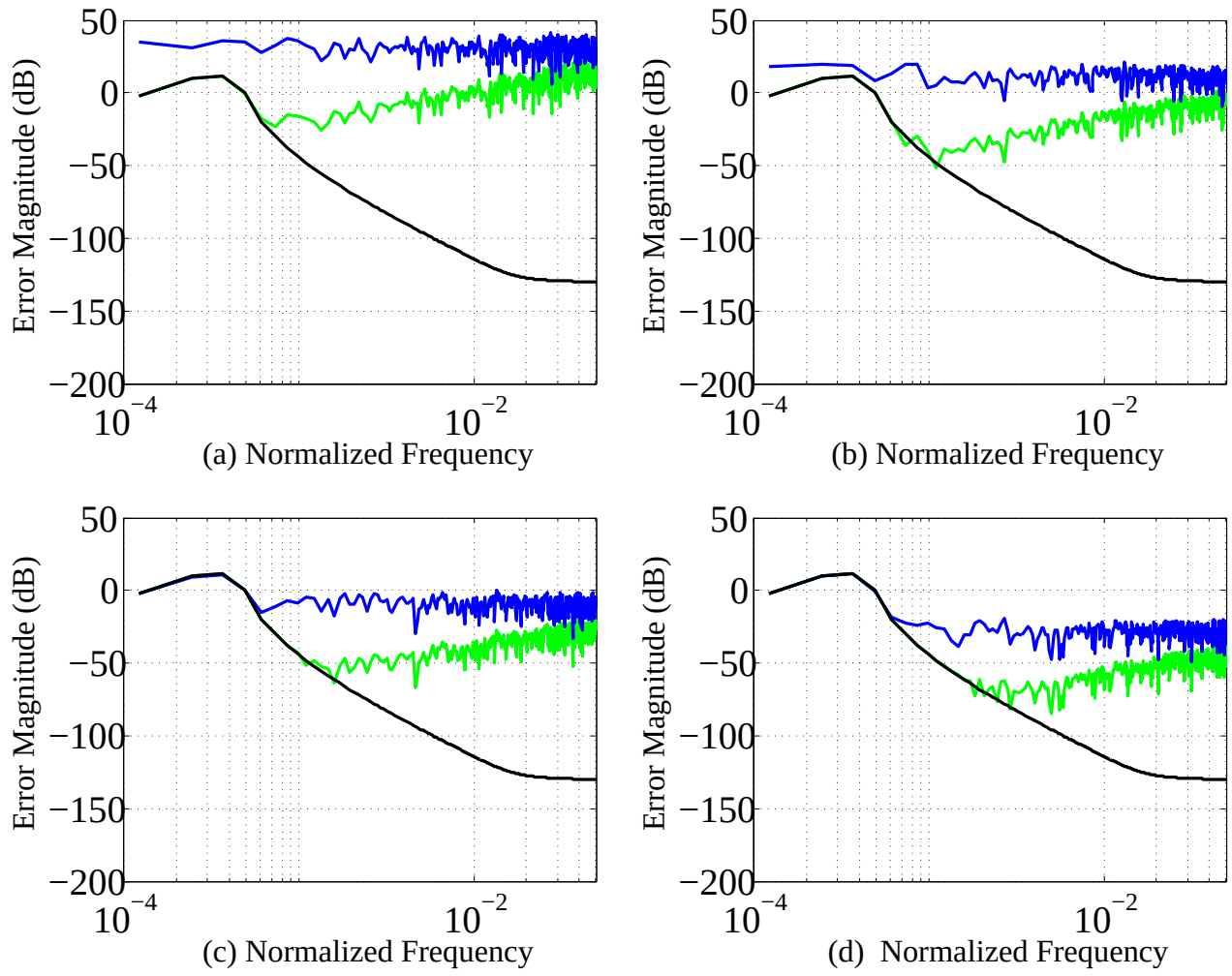


Figure 2.7: *Error magnitude in the case of ideal (in black),noisy (in blue) and* $\Sigma\Delta$ *(in green) gradient descent algorithms for a noise floor of (a) 30dB, (b) 10dB, (c) -10dB and (d) -30dB*

beyond the signal strength, the noisy gradient descent algorithm no longer converges to the original signal. But the $\Sigma\Delta$ learning approach shows better convergence characteristics, as the error is closer to the case of the ideal gradient descent algorithm.

In the next set of simulations, the performance of the noise induced algorithms is validated on a sample speech utterance "26 81 57" taken from the YOHO speaker verification data set. The speech sample at 8KHz is oversampled by OSR=128. Fig. 2.8 shows the FFT of the original speech sample along with the performance of noisy gradient descent and $\Sigma\Delta$ gradient

Table 2.1: *SNR (in dB) for different noise levels of the ideal, noisy and ΣΔ gradient descent algorithms for a sample speech utterance '26 81 57' from the YOHO database.*

Optimization Algorithm	Noise level (in dB)	SNR (in dB) $\epsilon = 0.1$	SNR (in dB) $\epsilon = 0.5$	SNR (in dB) $\epsilon = 0.9$
Ideal Gradient Descent	–	20.622	61.71	105.24
Noisy Gradient Descent	-30	19.998	61.07	87.41
	-10	16.241	41.36	44.31
	10	-2.61	-1.91	-2.13
	30	-46.09	-48.15	-46.15
ΣΔ Gradient Descent	-30	20.01	61.42	101.74
	-10	19.98	60.81	95.767
	10	19.58	52.42	57.52
	30	10.25	10.45	13.43

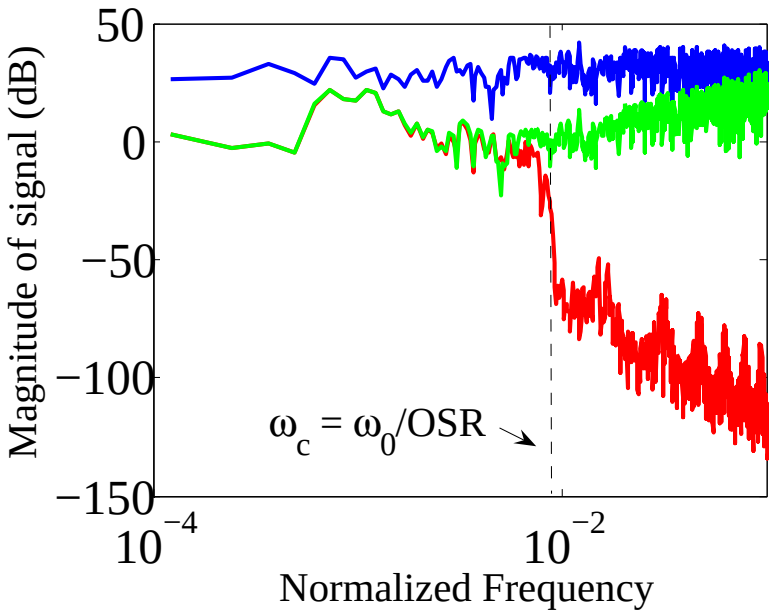


Figure 2.8: *FFT of the sample speech utterance "26 81 57" (in red) along with the noisy (in blue) and ΣΔ (in green) learning algorithm*

descent algorithms. It can be clearly seen that in signal band of interest ($\omega_C = 8KHz$), the performance of the ΣΔ algorithm outweighs the performance of the noisy gradient descent algorithm. Table 2.1 summarizes the Signal-to-Noise ratio of the output speech signal for different noise floor levels in the case of the ideal, noisy and noise shaped gradient descent algorithms.

2.2 Online Learning Example

In this section, we study in detail the performance of the gradient descent algorithms for an online tracking system. The loss function under consideration is

$$L(x) = \lim_{T\to\infty} \frac{1}{T} \int_0^T \frac{1}{2}\{y(t) - x\}^2 dt \tag{2.16}$$

where $y(t)$ is the signal that is being tracked by the parameter x. The loss function is being optimized with respect to x in real-time. The gradient in this case is given by

$$g(x) = \lim_{T\to\infty} \frac{1}{T} \int_0^T \{y(t) - x\} dt \tag{2.17}$$

The mathematical model for the error is derived for each of the algorithm.

2.2.1 Gradient Descent Tracking

The gradient descent algorithm attempts to track the input $y(t)$ by moving in the direction of the gradient given by $g(x)$. At time index n, the update for the tracking signal x is given by

$$x_n = x_{n-1} + \epsilon \left(y_n - x_{n-1}\right) \tag{2.18}$$

where ϵ is the step size.

Taking the Z-transform on both sides of eq. (2.18), we have

$$X(z) = z^{-1}X(z) + \epsilon \left(Y(z) - z^{-1}X(z)\right) \tag{2.19}$$

which gives

$$X(z) = \frac{\epsilon}{1 - (1-\epsilon)z^{-1}} Y(z) \tag{2.20}$$

The error function $E(z)$ would become

$$\begin{aligned} E(z) &= Y(z) - X(z) && (2.21) \\ &= \frac{(1-\epsilon)(1-z^{-1})}{1-(1-\epsilon)z^{-1}} Y(z) && (2.22) \end{aligned}$$

Replacing $1-\epsilon$ by α we have,

$$E(z) = \frac{\alpha(1-z^{-1})}{1-\alpha z^{-1}} Y(z) \tag{2.23}$$

For frequency $\omega \ll \omega_0$, replacing $z^{-1} = 1 - j\frac{\omega}{\omega_0}$

$$E(j\omega) = \frac{\alpha j \frac{\omega}{\omega_0}}{(1-\alpha) + j\alpha\frac{\omega}{\omega_0}} Y(j\omega) \tag{2.24}$$

Taking the magnitude of the relative error, we have

$$\frac{|E(j\omega)|}{|Y(j\omega)|} = \frac{\alpha\frac{\omega}{\omega_0}}{\left((1-\alpha)^2 + \alpha^2\frac{\omega^2}{\omega_0^2}\right)^{\frac{1}{2}}} \tag{2.25}$$

Replacing $\omega_0^{'} = \frac{1-\alpha}{\alpha}\omega_0$, we get

$$\frac{|E(j\omega)|}{|Y(j\omega)|} = \frac{\frac{\omega}{\omega_0^{'}}}{\left(1 + \frac{\omega^2}{\omega_0^{'2}}\right)^{\frac{1}{2}}} \tag{2.26}$$

The choice of the adaptation parameter ϵ effects the error magnitude of the gradient descent learning. Fig. 2.9 shows the real-time learning results of the tracking problem when the input signal is a tone given by $y(t) = sin(\omega t)$. For $\epsilon = 0.01$ the adaptation is not fast enough to track the input $y(t)$. As the value of *epsilon* decreases, the learning becomes less efficient. Eq. (2.6) shows that the optimal value of ϵ is given by the inverse of the hessian. In this case of the tracking problem, the hessian of the loss function is $\frac{\partial^2 L(x)}{\partial x^2} = \frac{\partial g(x)}{\partial x} = 1$.

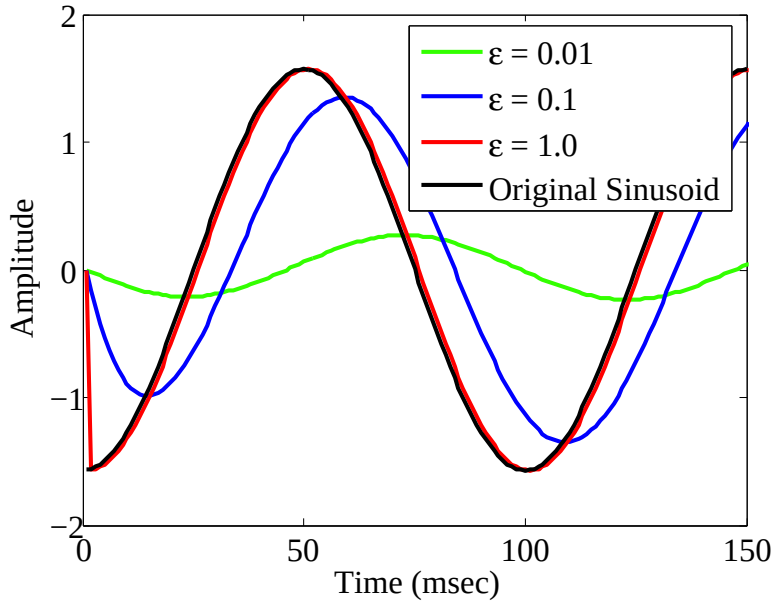


Figure 2.9: *Real-time tracking of a sinusoid signal with ideal gradient descent algorithm for* $\epsilon = 0.01,\ 0.1$ *and* 1.0

Therefore, the optimal value of the learning parameter is $\epsilon = 1$, in which case the output x perfectly tracks the input signal $y(t)$.

Fig. 2.10 gives a quantitative estimate of the relative error $\frac{|E(j\omega)|}{|Y(j\omega)|}$ in the estimate of x with respect to the variation of α. For a given value of the adaptation parameter ϵ, the error $E(z)$ increases linearly with the input signal frequency ω before saturating at a cutoff frequency $\omega^{'}$ given by

$$\omega_0^{'} = \frac{1-\alpha}{\alpha}\omega_0 \tag{2.27}$$

where ω_0 is the sampling rate. It can be seen that for $\alpha \to 0$, the error magnitude decreases to zero.

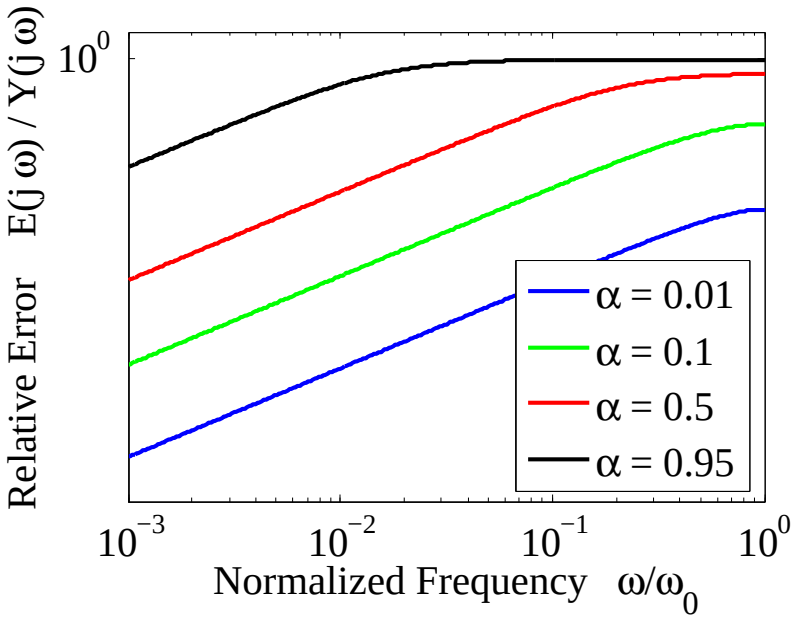


Figure 2.10: *Frequency dependency of the relative error* $\left|\frac{E(j\omega)}{Y(j\omega)}\right|$ *in gradient descent optimization for different values of the adaptation rate* α

2.2.2 Effect of Quantization

In online learning of most analog circuits, the gradient is digitized using an Analog-to-Digital converter(ADC) before it is used for updating the learning parameter. The digitization of the input signal introduces an additional quantization error in the estimate of the system parameter. The quantized gradient descent algorithm can be modeled by adding the quantization noise q_n to the ideal gradient estimate according to

$$x_n = x_{n-1} + \epsilon(y_n - x_{n-1}) + \epsilon q_n \tag{2.28}$$

Although the quantization noise q_n is a stochastic random process, we consider the case

of one particular realization of q_n. Taking the z-transform of the above equation, we get

$$X(z) = \frac{\epsilon}{1-(1-\epsilon)z^{-1}}(Y(z)+Q(z)) \tag{2.29}$$

where $Q(z)$ is the z-transform of the realization of quantization noise q_n. The error function $E(z)$ would be given by :

$$E(z) = \frac{\alpha(1-z^{-1})}{1-\alpha z^{-1}}Y(z) + \frac{1-\alpha}{1-\alpha z^{-1}}Q(z) \tag{2.30}$$

Comparing it to eq. (2.23), we see that quantization of the input signal adds an additional component of error $E_q(z)$ to the gradient descent case, given by

$$E_q(z) = \frac{1-\alpha}{1-\alpha z^{-1}}Q(z) \tag{2.31}$$

As in the case of the ideal gradient descent, replacing $z^{-1} = 1 - j\frac{\omega}{\omega_0}$ for frequency $\omega \ll \omega_0$, and taking the power spectral density(PSD) of the error, we get

$$\frac{|E_q(j\omega)|}{|Q(j\omega)|} = \frac{(1-\alpha)}{\left((1-\alpha)^2 + \alpha^2\frac{\omega^2}{\omega_0^2}\right)^{\frac{1}{2}}} \tag{2.32}$$

where the PSD is defined as $|a| = (aa^*)^{\frac{1}{2}}$. The additional error $E_q(z)$ added because of the quantization is therefore,

$$\frac{|E_q(j\omega)|}{|Q(j\omega)|} = \frac{1}{\left(1+\frac{\omega^2}{\omega_0'^2}\right)^{\frac{1}{2}}} \tag{2.33}$$

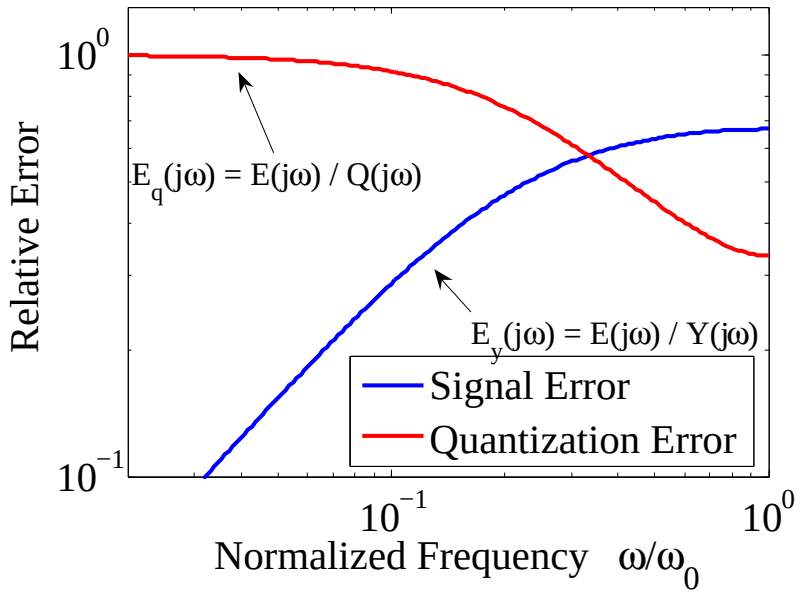


Figure 2.11: *Variation of the relative error w.r.t signal magnitude and quantization error for a quantized gradient descent algorithm for $\alpha = 0.5$*

where $\omega_0^{'} = \frac{1-\alpha}{\alpha}\omega_0$.

Fig 2.11 shows the variation of the relative error due to the quantization process $E_q(z)$ and also the signal error $E_y(z)$ from the ideal gradient descent for a particular value of α. The eq. (2.31) shows that for a given α the error component $E_q(z)$ has low pass characteristics with respect to the quantization error q_n. The cutoff frequency $\omega_0^{'}$ is the same as in the ideal gradient descent case and the quantization error decreases after $\omega_0^{'}$. Depending on the statistics of the input signal, a particular value of α can be choosen inorder to reduce the overall error in the quantized gradient case.

The error q_n introduced by the quantization of the input signal can be assumed to be uncorrelated with the input $y(t)$ if the sampling rate ω_0 is much higher than ω. Under this condition, the error q_n can be approximated to white noise with a uniform power spectral density. Fig. 2.12 shows the variation of the noise floor of the quantization error q_n with

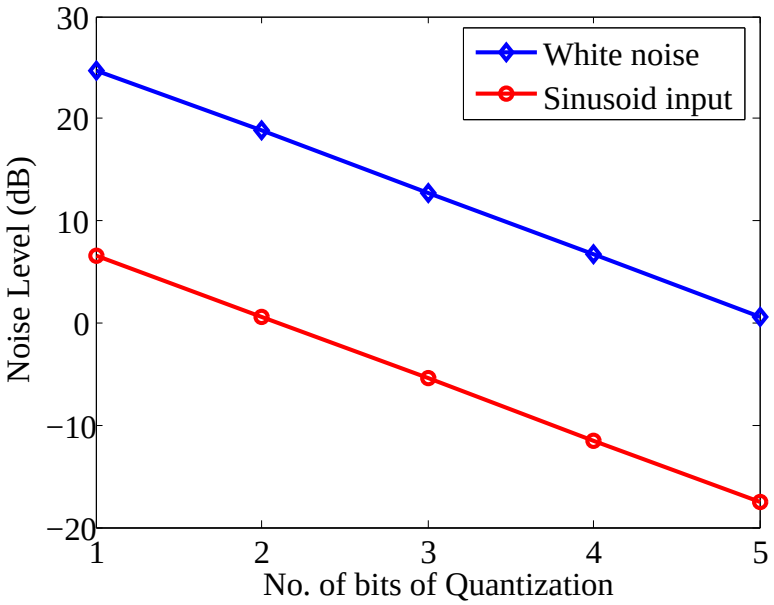


Figure 2.12: *Variation of Noise floor (in dB) with increase in no. of quantization bits*

increase in the number of quantization bits for both a random input signal and tone. For every one bit increment in the quantizer, the noise floor due to quantization reduces by 6dB. As the number of quantization bits increases, the relative error due to quantization process decreases.

2.2.3 ΣΔ Gradient Descent Algorithm

The mathematical model for the ΣΔ learning system in the case of the learning problem is given by

$$w_n = w_{n-1} + (y_n - x_n) - d_n \tag{2.34}$$

$$d_n = w_{n-1} + q_n \tag{2.35}$$

$$x_{n+1} = x_n - \epsilon d_n \tag{2.36}$$

Replacing n by $n-1$ in the above iteration, we get

$$w_{n-1} = w_{n-2} + (y_{n-1} - x_{n-1}) - d_{n-1} \tag{2.37}$$

which gives,

$$d_n = (y_{n-1} - x_{n-1}) + (q_n - q_{n-1}) \tag{2.38}$$

Substituting the above equation back in eq. (2.36) we get

$$x_n = x_{n-1} + \epsilon(y_{n-1} - x_{n-1}) + \epsilon(q_n - q_{n-1}) \tag{2.39}$$

For one particular realization of the quantization noise q_n, taking the z-transform of the above equation we have,

$$X(z) = \frac{\epsilon}{1-\alpha z^{-1}} Y(z) + \frac{\epsilon(1-z^{-1})}{1-\alpha z^{-1}} Q(z) \tag{2.40}$$

The error function $E(z)$ is given by

$$E(z) = \left(\frac{\epsilon}{1-\alpha z^{-1}} - 1\right) Y(z) + \frac{\epsilon(1-z^{-1})}{1-\alpha z^{-1}} Q(z) \tag{2.41}$$

$$E(z) = \frac{1-z^{-1}}{1-\alpha z^{-1}} \left(\alpha Y(z) + \epsilon Q(z)\right) \tag{2.42}$$

The contribution of the error from quantization of the signal $E_q z$ in the $\Sigma\Delta$ gradient descent case is given by

$$E_q(z) = \frac{\epsilon(1-z^{-1})}{1-\alpha z^{-1}} Q(z) \tag{2.43}$$

As in the case of the ideal gradient descent, replacing $z^{-1} = 1 - j\frac{\omega}{\omega_0}$ for frequency

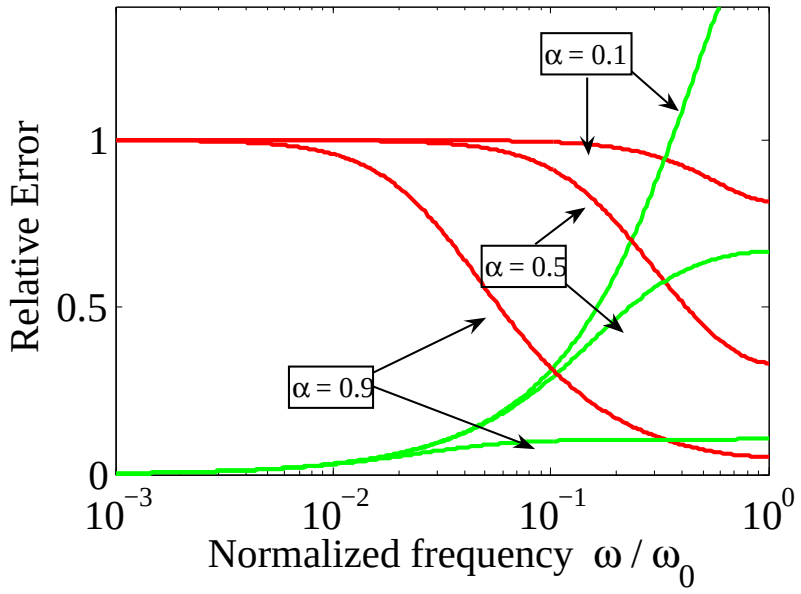


Figure 2.13: *Variation of the relative error w.r.t normalized frequency for a quantized (in red) and ΣΔ (in green) gradient descent algorithm for different values of α*

$\omega \ll \omega_0$, and taking the power spectral density(PSD) of the error, we get

$$\frac{|Eq(j\omega)|}{|Q(j\omega)|} = \frac{(1-\alpha)\frac{\omega}{\omega_0}}{\left((1-\alpha)^2 + \alpha^2\frac{\omega^2}{\omega_0^2}\right)^{\frac{1}{2}}} \tag{2.44}$$

Replacing $\omega_0^{'} = \frac{1-\alpha}{\alpha}\omega_0$, we have

$$\frac{|Eq(j\omega)|}{|Q(j\omega)|} = \left(\frac{1-\alpha}{\alpha}\right)\frac{\frac{\omega}{\omega_0^{'}}}{\left(1+\left(\frac{\omega}{\omega_0^{'}}\right)^2\right)^{\frac{1}{2}}} \tag{2.45}$$

Fig. 2.13 shows the comparison of the error due to quantization in the case of noisy gradient descent (in red) and the ΣΔ gradient descent algorithm (in green). Eq. (2.43)

shows that for a given value of the adaptation parameter α, the quantization error has highpass characteristics with a cutoff frequency $\omega_0^{'}$. Also, it can be clearly seen that the quantization error noise floor decreases with increase in the value of α. For a input signal with known characteristics, a proper choice of α can be made to reduce the overall error due to the signal and quantization error components.

Fig. 2.14 and 2.15 compare the error convergence of the three algorithms with the variation of the adaptation rate ϵ for a 1-bit and 4-bit quantizer respectively. The simulation results validate the error analysis described in the previous sections. For the quantized gradient descent learning algorithm, the overall error increases as the value of ϵ increases. On the other hand, for the $\Sigma\Delta$ gradient descent approach, the noise shaping characteristics can be clearly seen and the overall error actually decreases with increase in the value of ϵ. In Fig. 2.14(d) and 2.15(d) the error in the case of ideal gradient descent is zero and is not plotted.

Fig. 2.16 shows the variation of the error convergence of all the three optimization algorithms with change in the number of quantization bits for a constant $\epsilon = 0.5$. As the quantization bits increases, we see that the error decreases in both the quantized and the $\Sigma\Delta$ algorithms. This is because the noise floor due to the quantization error decreases with increase in the number of bits of quantization as shown in fig 2.12.

In the next set of simulations, the signal-to-noise(SNR) ratio of each of the algorithms is calculated by using a tone at $\omega = 0.01\omega_0$. The output from each optimization algorithm is low pass filtered and the SNR is calculated. Fig. 2.17 shows how the SNR varies as a function of the adaptation parameter ϵ for different values of the quantization bits. It can be observed that the SNR of $\Sigma\Delta$ learning algorithm is much closer in performance to the

Table 2.2: *SNR (in dB) for different quantization levels of the different optimization algorithms on a sample speech utterance "26 81 57".*

Optimization Algorithm	Quantization levels	SNR (in dB) $\epsilon = 0.1$	SNR (in dB) $\epsilon = 0.5$	SNR (in dB) $\epsilon = 0.9$
Ideal Gradient Descent	–	20.622	61.39	105.24
Quantized Gradient Descent	1	-9.87	-44.13	-56.27
	2	6.03	-29.59	-41.93
	4	37.60	0.734	-12.51
	8	22.27	63.08	51.17
$\Sigma\Delta$ Gradient Descent	1	21.54	25.80	12.40
	2	20.50	43.93	28.26
	4	20.15	60.16	69.51
	8	20.04	61.19	102.5

ideal gradient descent case. The variation of the SNR with respect to the no. of bits of quantization is shown in Fig. 2.18 for different values of ϵ.

In the next set of simulations, the performance of the quantized gradient descent and $\Sigma\Delta$ algorithm is validated on the same sample speech utterance "26 81 57" used in the previous section. The oversampling ratio is still set at OSR = 128. Table 2.2 summarizes the Signal-to-Noise ratio of the output speech signal for different levels of quantization in the case of the ideal, noisy and noise shaped gradient descent algorithms.

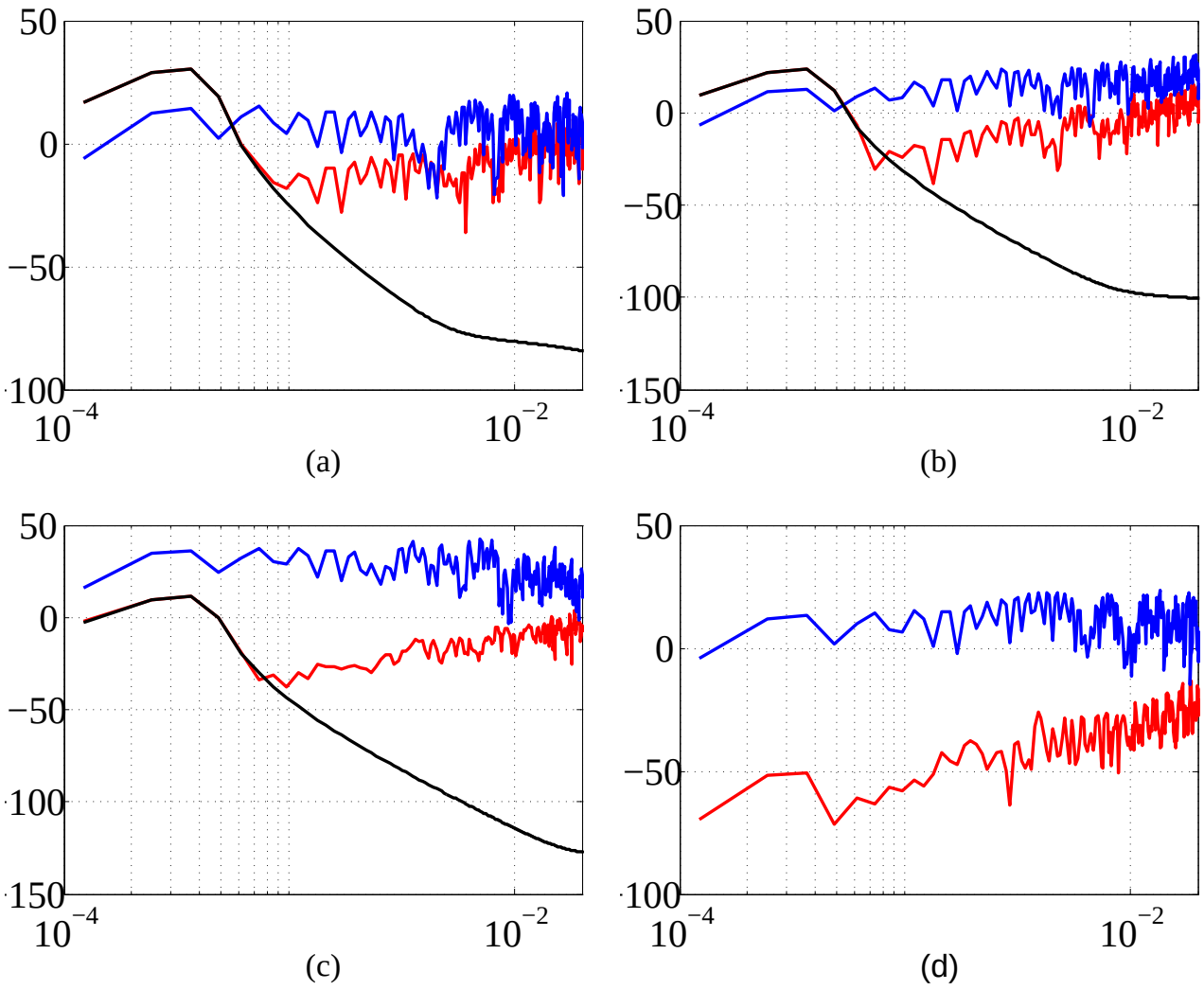


Figure 2.14: *Error(in dB) vs normalized frequency of ideal gradient descent(in black), quantized gradient descent(in blue) and 1st order $\Sigma\Delta$ gradient descent (in red) optimization algorithms with a 1-bit quantizer for a (a) $\epsilon = 0.1$,(b) $\epsilon = 0.2$,(c) $\epsilon = 0.5$ and (d) $\epsilon = 1.0$*

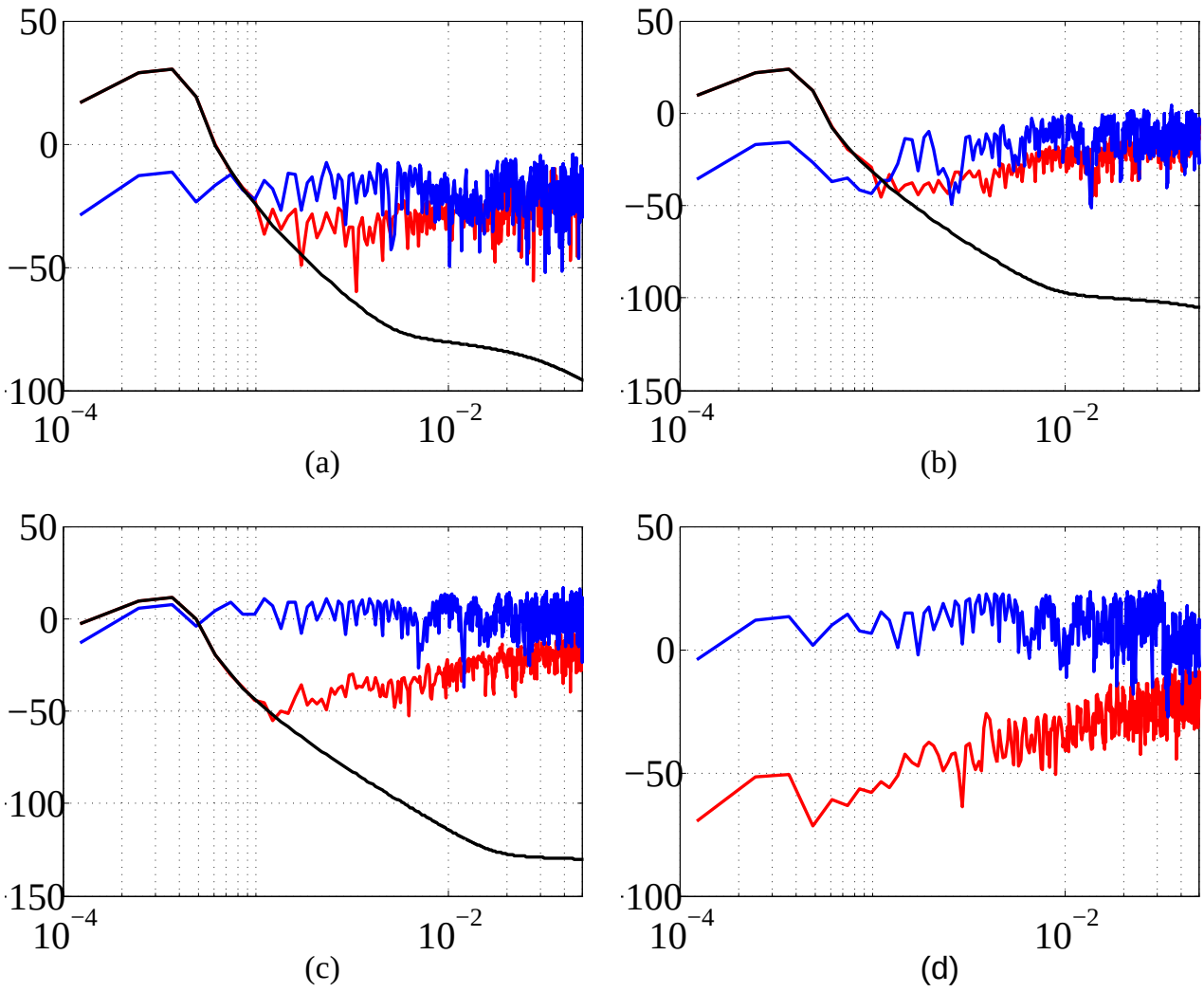


Figure 2.15: *Error(in dB) vs normalized frequency of ideal gradient descent(in black), quantized gradient descent (in blue) and 1st order $\Sigma\Delta$ gradient descent (in red) optimization algorithms with a 4-bit quantizer for a (a) $\epsilon = 0.1$,(b) $\epsilon = 0.2$,(c) $\epsilon = 0.5$ and (d) $\epsilon = 1.0$*

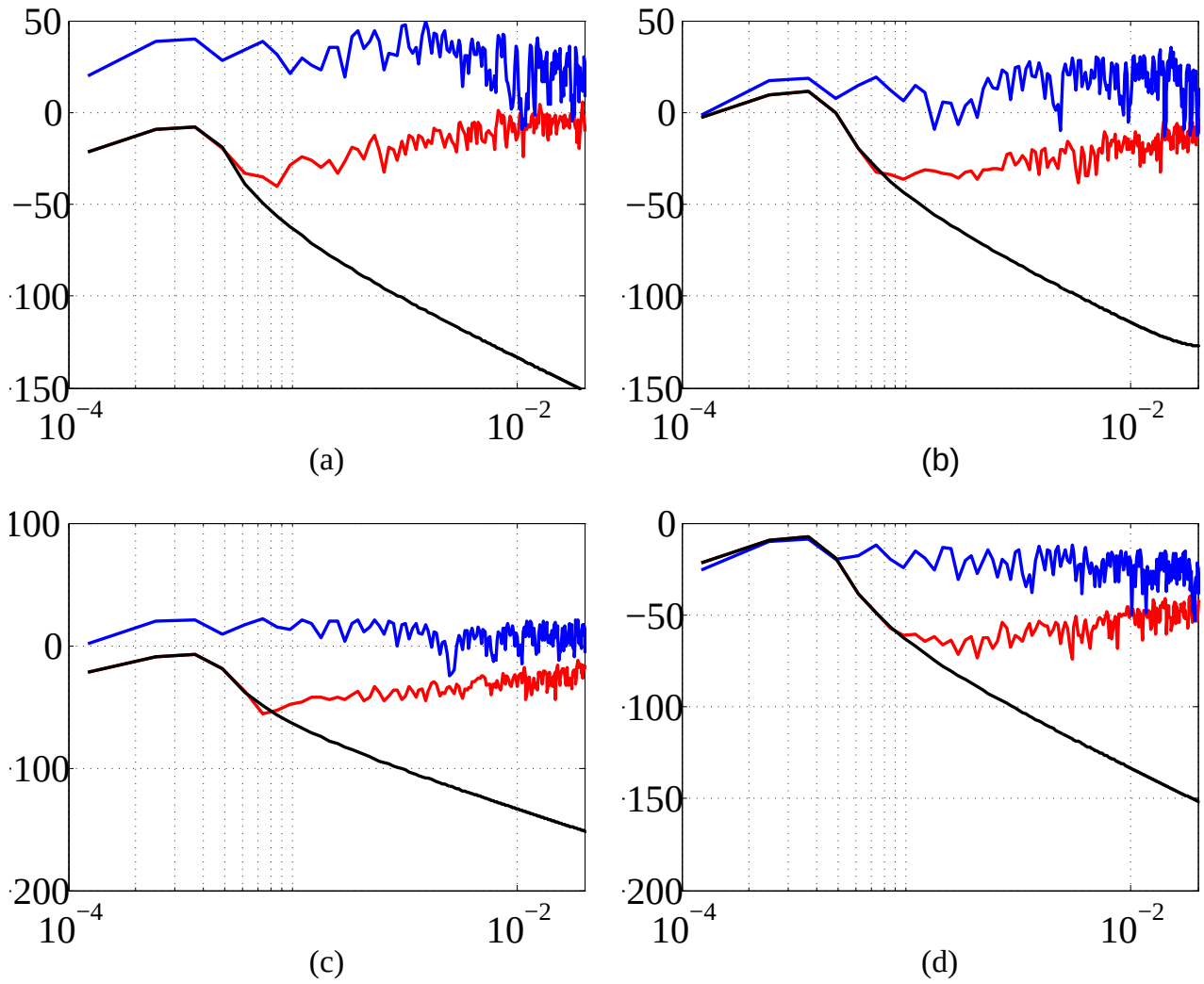


Figure 2.16: *Error(in dB) vs normalized frequency of ideal gradient descent (in black), quantized gradient descent (in blue) and 1st order $\Sigma\Delta$ gradient descent (in red) optimization algorithms for $\epsilon = 0.5$, with a n-bit quantizer for (a) n=1,(b) n=2,(c) n=4 and (d) n=8*

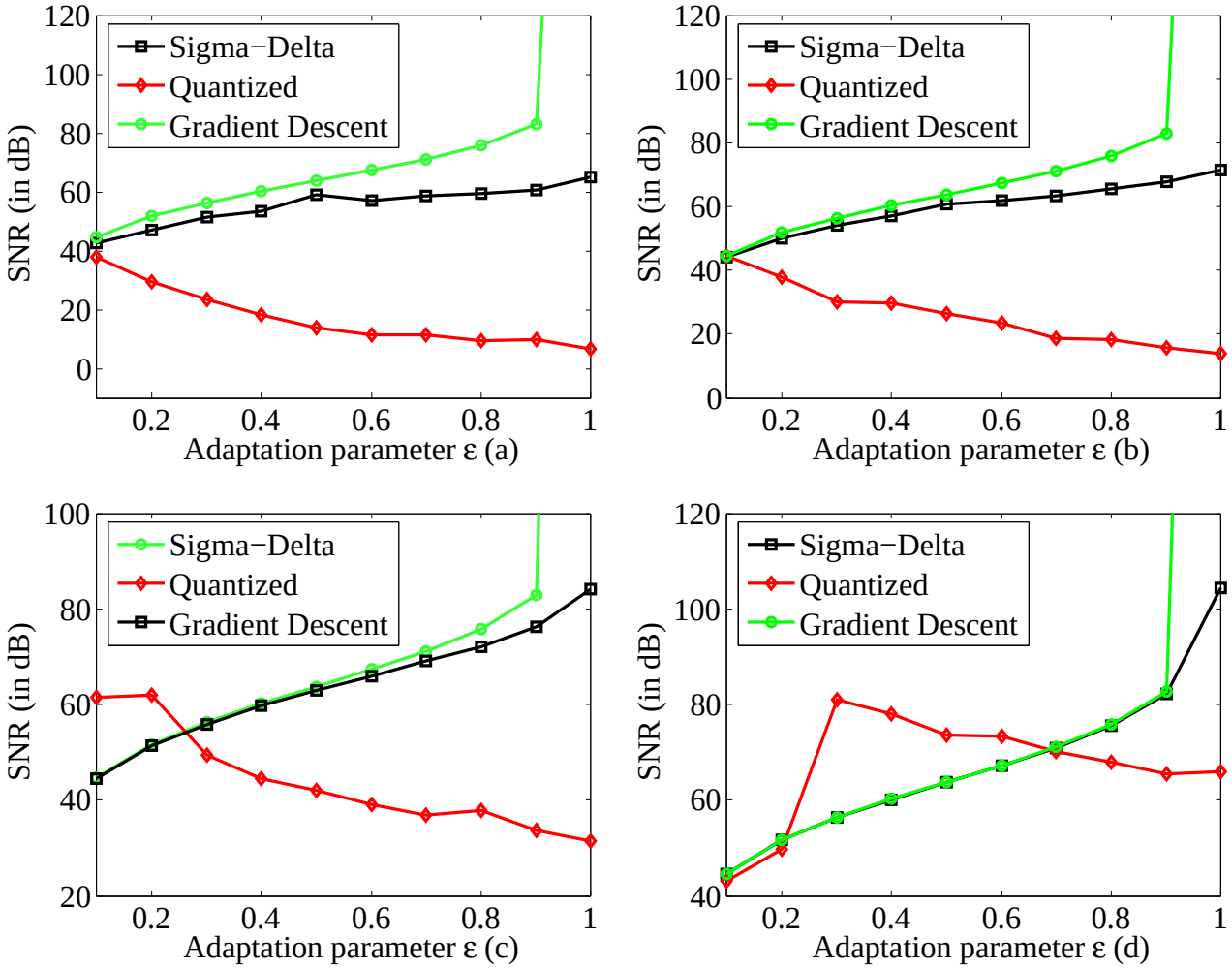


Figure 2.17: *SNR(in dB) vs adaptation parameter ϵ of gradient descent, quantized gradient descent and 1st order $\Sigma\Delta$ gradient descent optimization algorithms with a n-bit quantizer for (a) n=1,(b) n=2,(c) n=4 and (d) n=8*

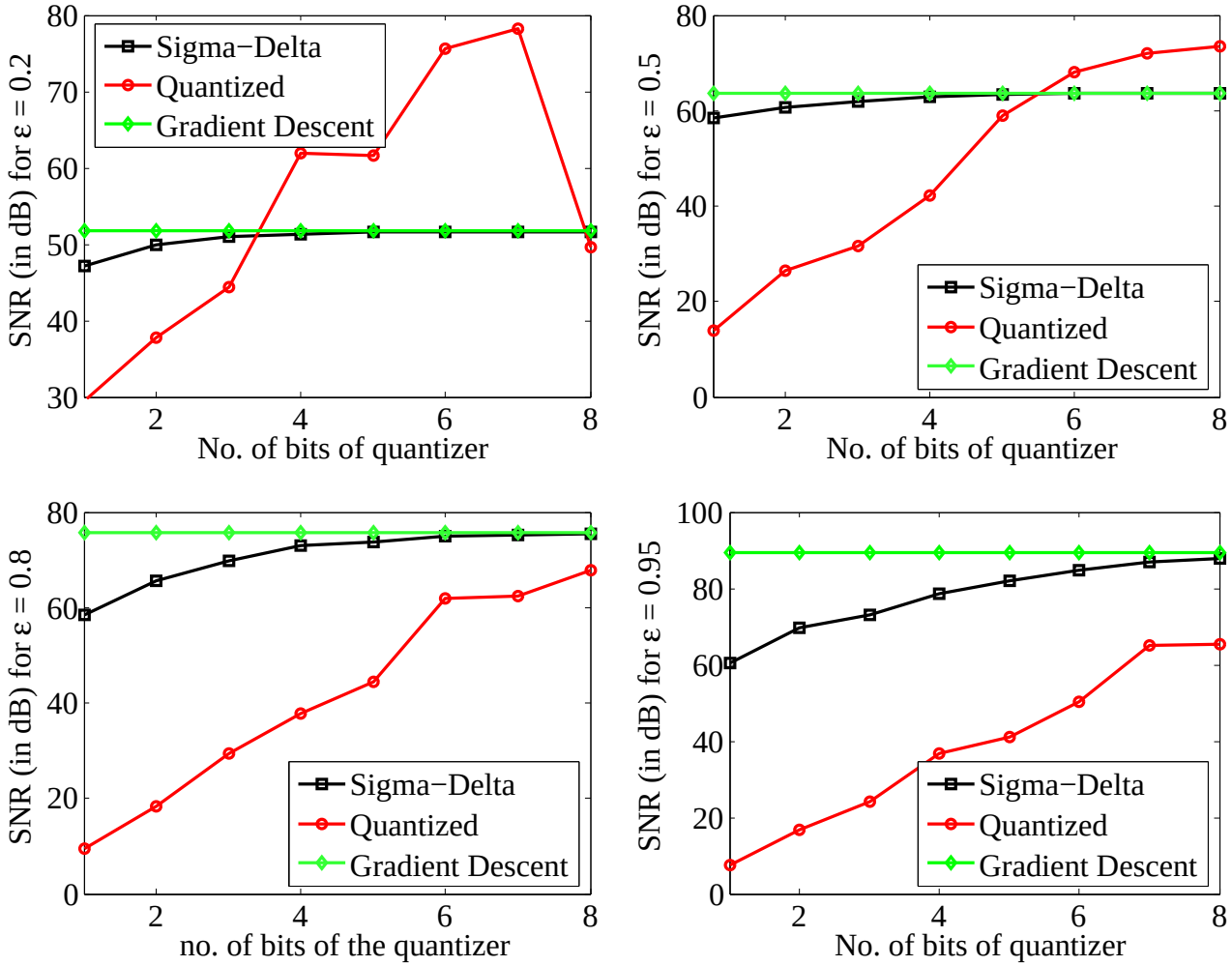


Figure 2.18: *SNR(in dB) vs no. of quantization bits of gradient descent, quantized gradient descent and 1st order $\Sigma\Delta$ gradient descent optimization algorithms for (a) ϵ=0.2,(b)ϵ=0.5,(c) ϵ=0.8 and (d) ϵ=0.95*

Chapter 3

Generalized Noise Shaping Algorithm

The $\Sigma\Delta$ gradient descent learning algorithm introduced in the previous chapter made use of a simple first order integrator as the loop filter. Also a simple first-order gradient descent approximation method was used in the estimation of the updated optimization parameter. In this chapter, we generalize the same algortihm by using a higher order loop filter for the $\Sigma\Delta$ modulation stage and a generalized learning filter for the optimization parameter update. The loss function $L(\theta)$ under consideration is assumed to be continuous and differentiable with $g(\theta)$ being the partial derivative of $L(\theta)$ with respect to the optimization parameter θ. Fig. 3.1 shows the block diagram of the generalized noise-shaping learning algorithm. The generalized loop filter for the $\Sigma\Delta$ modulation stage is $H(z)$ and the feedback learning filter is $F(z)$.

The output $Y(z)$ of the modulator stage is given by

$$Y(z) = H(z)[g(\theta) - Y(z)] + Q(z) \tag{3.1}$$

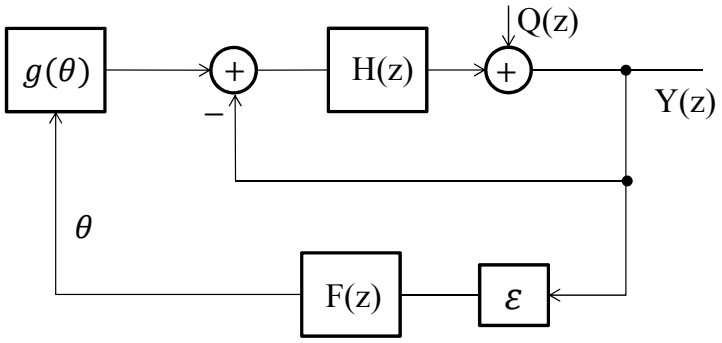


Figure 3.1: *Generalized block diagram of the ΣΔ gradient descent algorithm*

which gives

$$Y(z) = \left(\frac{H(z)}{1+H(z)}\right) g(\theta) + \left(\frac{1}{1+H(z)}\right) Q(z) \tag{3.2}$$

When the parameter estimate θ is close enough to the optimal solution θ^*, the Taylor's series expansion of $g(\theta)$ about the root θ^* gives

$$g(\theta) = g(\theta^*) + (\theta - \theta^*) g^{'}(\theta^*) \tag{3.3}$$

As θ^* is the root of $g(\theta)$, $g(\theta^*) = 0$. Replacing $(\theta - \theta^*)$ as a new error variable Θ we get,

$$g(\theta) = \Theta g^{'}(\theta^*) \tag{3.4}$$

Replacing eq. (3.4) in eq. (3.2),

$$Y(z) = \frac{H(z)\Theta g^{'}(\theta^*)}{1+H(z)} + \frac{Q(z)}{1+H(z)} \tag{3.5}$$

The feedback path in fig. 3.1 shows that the learning update of Θ can be written as

$$\Theta(z) = \epsilon F(z) Y(z) \tag{3.6}$$

Substituting it back in eq. (3.5), we have

$$\Theta(z) = \frac{\epsilon g^{'}(\theta^*) H(z) F(z) \Theta(z) + \epsilon F(z) Q(z)}{1 + H(z)} \tag{3.7}$$

which gives

$$\Theta(z) = \frac{\epsilon F(z) Q(z)}{1 + H(z) - \epsilon g^{'}(\theta^*) H(z) F(z)} \tag{3.8}$$

The above equation shows the dependancy of the error estimate Θ as a function of the quantization error $Q(z)$ in the generalized form of noise shaping algorithm. The error transfer function w.r.t quantization noise $\Theta_q(z)$ is given by

$$\Theta_q(z) = \frac{\Theta(z)}{Q(z)} \tag{3.9}$$

$$\Theta_q(z) = \frac{\epsilon F(z)}{1 + H(z) - \epsilon g^{'}(\theta^*) H(z) F(z)} \tag{3.10}$$

The magnitude of the error is given by

$$|\Theta(z)| = \frac{\epsilon}{\left| \frac{1}{F(z)} + \frac{H(z)}{F(z)} - \epsilon g^{'}(\theta^*) H(z) \right|} |Q(z)| \tag{3.11}$$

From eq. (3.11), depending on the statistics of the loss function being minimzed, proper choice of the loop filter $H(z)$ and feedback learning filter $F(z)$ can be made.

3.1 Higher Order ΣΔ Modulation

The fundamental ideas of noise-shaping of ΣΔ modulator used in gradient descent algorithms can be extended to the use of higher order modulation. The schematic diagram of a second-order modulator is shown in fig. 3.2. The structure makes use of two integrators $H_1(z)$ and $H_2(z)$ as opposed to just one used in first order ΣΔ modulation. The error signal at the end of first integration stage is used as the input for the second integrator to achieve higher resolution than a simple first order modulator. From the z-domain analysis of the second

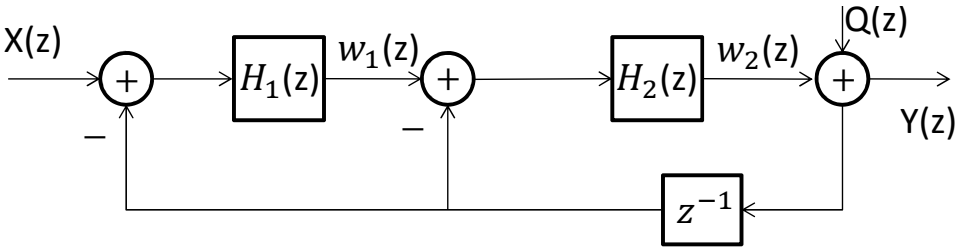


Figure 3.2: *Block diagram of a second order sigma-delta modulator*

order ΣΔ modulator, the output of the first integrator stage $w_1(z)$ can be written as

$$w_1(z) = \Big(X(z) - z^{-1}Y(z)\Big)\, H_1(z) \tag{3.12}$$

and the output $w_2(z)$ of the second integrator can be written as

$$w_2(z) = \Big(w_1(z) - z^{-1}Y(z)\Big)\, H_2(z) \tag{3.13}$$

The quantization at the output stage is modelled by using an additive noise $Q(z)$. The output of the modulator $Y(z)$ is given by

$$Y(z) = w_2(z) + Q(z) \tag{3.14}$$

From the above three equations, eliminating the variables $w_1(z)$ and $w_2(z)$, we get

$$Y(z) = \frac{H_1(z)H_2(z)}{1 + z^{-1}H_2(z)(1 + H_1(z))}X(z) + \frac{1}{1 + z^{-1}H_2(z)(1 + H_1(z))}Q(z) \tag{3.15}$$

Replacing the value of $H_1(z)$ and $H_2(z)$ by the integrator z-domain transfer function

$$H_1(z) = H_2(z) = \frac{1}{1 - z^{-1}} \tag{3.16}$$

in eqn. (3.15), we have

$$Y(z) = z^{-1}X(z) + (1 - z^{-1})^2Q(z) \tag{3.17}$$

which gives the noise transfer function(NTF) as $(1 - z^{-1})^2$. The second order modulator provides more quantization noise suppression over low frequency signal band compared to the first order modulator, thereby giving higher resolution of the output signal. The second order modulator shown in fig. 3.2 can be extended to a higher order modulator by using L integrators in succession. The noise transfer function NTF of such a modulator can be derived to be of the form

$$NTF = \frac{Y(z)}{Q(z)} = (1 - z^{-1})^L \tag{3.18}$$

Figure 3.3 shows the improvement in the noise-shaping characteristics of a $\Sigma\Delta$ modulator as the order of modulator increases. It can be shown that for a given oversampling ratio, the resolution of the modulator increases by 6dB or 1 bit as the order increases.

The noise shaping characteristics of the higher order modulators can be exploited in the optimization of the learning system introduced in Section 2.2. Fig 3.4 shows the performance of the $\Sigma\Delta$ gradient descent algorithm when the loop filter $H(z)$ in fig 3.1 is replaced by a

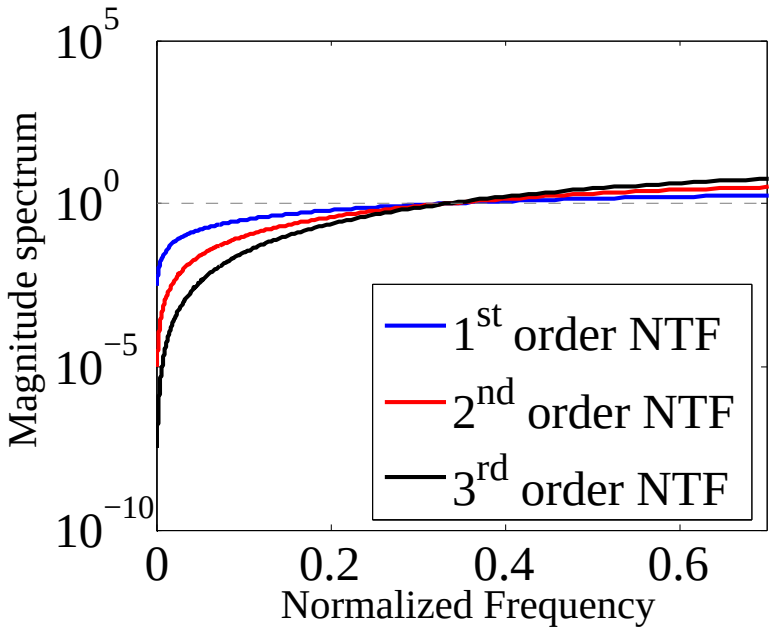


Figure 3.3: *Noise transfer function(NTF) of a first, second and third order* $\Sigma\Delta$ *modulator*

higher order $\Sigma\Delta$ modulator. The figure clearly shows that the tracking error decreases as the order of the $\Sigma\Delta$ modulator increases. For a given oversampling ratio, the SNR of the output increases by 6dB or 1-bit when the order of the $\Sigma\Delta$ modulator increases by one.

3.2 Batch Gradient Descent Optimization

The online gradient descent algorithm introduced in eq. (2.3) uses only the instantaneous value of the gradient for optimization. In the case of noise corrupted signals, a better estimate of the gradient direction can be got by using the knowledge of gradient direction from previous iterations [17]. More precisely, for the standard first order $\Sigma\Delta$ optimization, the feedback learning transfer function F(z) from fig. 3.1 can be used to accommodate previous values of the output pulse d in the gradient estimate. We consider the case of batch gradient descent when the learning parameter θ is updated using the previous two values of $\Sigma\Delta$

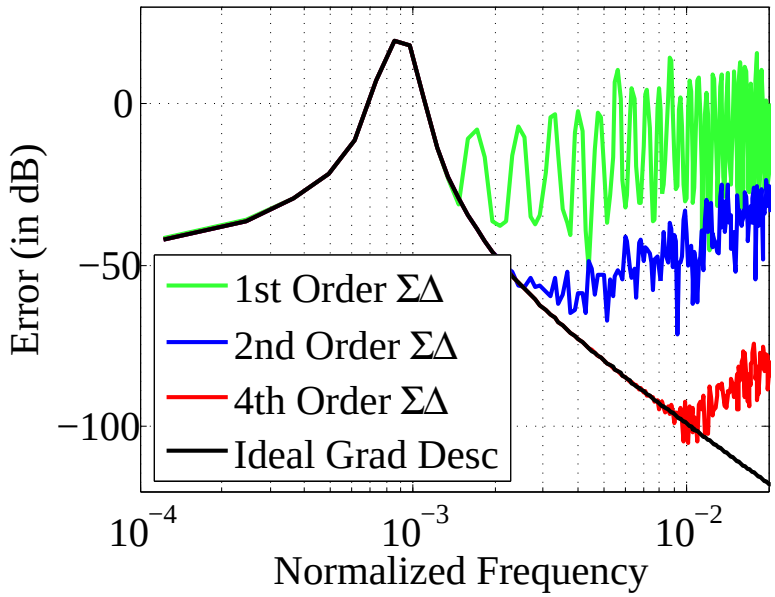


Figure 3.4: *Error (in dB) vs normalized frequency of the gradient descent algorithm with higher order(1^{st}, 2^{nd} and 4^{th} order) $\Sigma\Delta$ modulation*

modulator output. The update for the parameter θ using gradient descent algorithm in this case is given by

$$\theta_n = \theta_{n-1} - \epsilon(d_n + d_{n-1}) \tag{3.19}$$

Taking the z-transform of the above equation, we have

$$(1 - z^{-1})\Theta(z) = \epsilon(1 + z^{-1})D(z) \tag{3.20}$$

which gives

$$\Theta(z) = \epsilon\frac{(1 + z^{-1})}{(1 - z^{-1})}D(z) \tag{3.21}$$

Comparing it to the eqn. (3.6), the feedback learning transfer function $F(z)$ is given by

$$F(z) = \frac{(1 + z^{-1})}{(1 - z^{-1})} \tag{3.22}$$

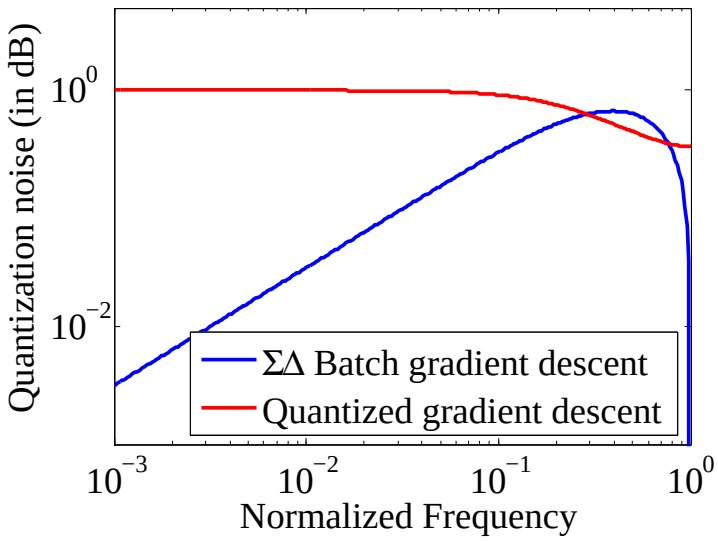


Figure 3.5: *Variation of the relative quantization error w.r.t the normalized frequency for quantized gradient descent and batch gradient descent algorithms.*

For a first order sigma delta modulator, the loop transfer function $H(z)$ is given by

$$H(z) = \frac{z^{-1}}{(1 - z^{-1})} \tag{3.23}$$

Substituting eqns (3.22) and (3.23) back in the generalized error equation (3.10),the error transfer function $\Theta(z)$ becomes

$$\Theta(z) \quad = \quad \epsilon \frac{1 - z^{-2}}{(1 - \epsilon) - (1 + \epsilon)z^{-1}} Q(z) \tag{3.24}$$

Fig. 3.5 shows the noise shaping characteristics of the above batch optimization technique compared to the standard quantization gradient descent approach. The quantization noise level at lower frequencies is attenuated in the batch gradient descent algorithm whereas for the quantized gradient descent case, the noise floor is fairly constant throughout the frequency range.

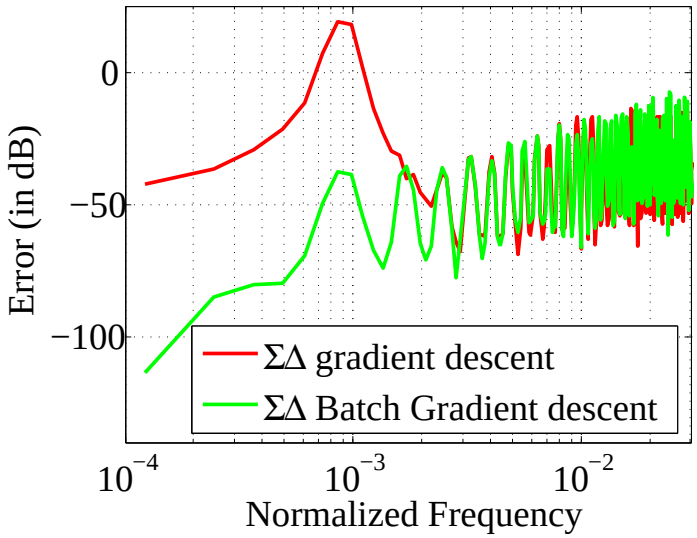


Figure 3.6: *Error (in dB) vs normalized frequency for gradient descent,first order* $\Sigma\Delta$ *and* $\Sigma\Delta$ *gradient descent with feedback filter* $F(z)$ *for* ϵ*=0.5*

Fig. 3.6 shows the results when batch gradient descent algorithm is used for optimization of the tracking problem discussed in Section 2.2. The figure clearly shows that at lower signal frequencies, the batch gradient descent has a much lower error as compared to the $\Sigma\Delta$ version of the algorithm.

Chapter 4

Gradient-Free Stochastic Optimization

In many real-life applications, the functional relationship between the optimization parameters and the loss function measurements is not readily available. The gradient based optimization algorithms discussed in Chapter 2 assume that such a relationship exists and that the gradient can be readily computed based on the relationship. But in many complex systems such as problems dealing with adaptive control, image restoration from noisy data, training of neural networks and discrete-event systems the gradient measurement is either not possible or computationally expensive. The models representing such complex system may be highly inaccurate and not dependable. In such cases, several gradient-less algorithms have been proposed in literature which make use of direct loss function measurements instead of the gradient measurement. Direct search algorithms refers to the class of optimization techniques which donot require the gradient value in the estimate of the optimal value. Pattern search algorithm [1], Nelder-Mead simplex algorithm [14], Rosenbrock algorithm [15]

are some of the popular zeroth order optimization techniques which make use of only the loss function measurement in the optimization. The general idea behind these algorithms is that the optimization parameter is moved in the direction in which the loss function is decreasing.

Another class of gradient-free algorithms like Keifer-Wolfowitz finite-difference approximation(FDSA) and Simulataneous Perturbation Stochastic Approximation(SPSA) attempt at finding the approximation to the gradient using only the loss function measurement. These gradient approximation algorithms lead to asymptotic convergence of the parameter estimate to the optimal solution. The problem with such algorithms though lies in the fact that the convergence of these gradient-approximation algorithms is usually slower than that of gradient based algortihms [18]. But depending on the loss function that is being considered, the cost of evaluating the gradient may outweigh the speed of the gradient based algorithms in which case gradient-approximation algorithms become handy. In this Chapter, we extend the noise shaping optimization technique introduced in Chapter 2 to the case when only the loss function measurement can be made and the gradient is not available. We introduce the $\Sigma\Delta$ optimization framework into the gradient-free algorithms discussed above.

4.1 Finite Difference Stochastic Approximation

Consider the optimization of a system with p adjustable parameters and let the objective function be $L(\theta)$. Here the optimization parameter θ is a p-dimensional vector such that $\theta \in \mathcal{R}^p$, where $p > 1$. If $\hat{g}(\theta)$ represents the approximation of the gradient made by using the loss function measurements, then the approximation can be used in the iterative learning of

the optimization parameter θ using gradient descent method as follows:

$$\theta_{n+1}^{k} = \theta_{n}^{k} - \epsilon_n \hat{g}^{k}(\theta_{n}^{k}) \tag{4.1}$$

Here, the index n represents the iteration number and the index k represents the k^{th} element of the vector θ, $k \in \{1, 2, ..., p\}$. In the finite difference approximation method, each of the gradient $\hat{g}^{k}$ with respect to the k^{th} element is calculated by perturbing the parameter θ along the direction of the unit vector u^{k} representing the direction of θ^{k}. The approximation can either be uni-directional or bi-directional with respect to the unit vector u^{k}. In the case of bi-directional estimation, the finite-difference approximation is given by

$$\hat{g}^{k}(\theta_n) = \frac{L(\theta_n + c_n u^{k}) - L(\theta_n - c_n u^{k})}{2c_n} \tag{4.2}$$

In the case of finite-difference approximation of $\hat{g}(\theta)$, at each iteration $2p$ number of evaluations of the loss function need to be made to get the complete gradient estimate along all the p-dimensions.

4.2 Simultaneous Perturbation Stochastic Approximation

In the case of simultaneous perturbation approximation of the gradient $\hat{g}(\theta)$, the vector θ is perturbed simultaneously along all the p-dimensions instead of individually evaluating the loss function across each dimension. The SPSA approximation of the gradient makes use of the fact that a properly generated random change of the vector θ contains the same amount

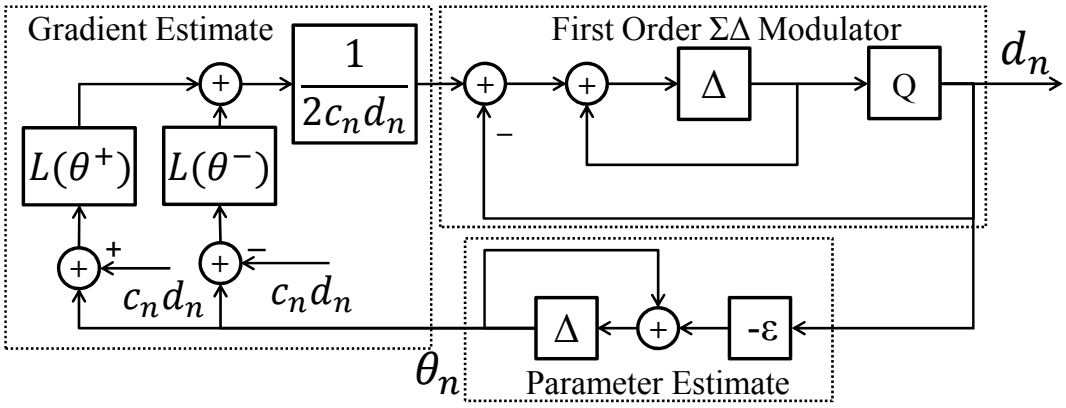


Figure 4.1: *Block diagram of the ΣΔ SPSA gradient descent algorithm*

of information about the gradient as in the case of p perturbations of the same vector θ along each dimension. The gradient estimate in the case of SPSA is given by

$$\hat{g}^k(\theta_n) = \frac{L(\theta_n + c_n\Delta_n) - L(\theta_n - c_n\Delta_n)}{2c_n\Delta_n^k} \tag{4.3}$$

where Δ_n is the perturbation vector along all the p-dimensions. Imposing certain statistical conditions on c_n, Δ_n and gain sequence ϵ_n [20], the SPSA algorithm is known to converge asymptotically to the optimal solution $\theta \to \theta^*$. One particular choice of Δ_n known to aid the convergene is a bernoulli sequence $\{\pm 1\}$. The SPSA algorithm gains heavily over the finite difference method as only 2 measurements of the loss function are required to calculate the gradient along all the dimensions p. The convergence rate of both the algorithms is the same.

In this chapter, we extend the ΣΔ gradient approximation framework from chapter 2 to the estimation of gradient using SPSA. As the true gradient $g(\theta)$ is not available, the approximation $\hat{g}(\theta)$ generated by either SPSA (eq. (4.3)) or FDSA (eq. (4.2)) can be used in the noise-shaping optimization. Fig. 4.1 shows the block diagram of the SPSA algorithm

introduced into the $\Sigma\Delta$ optimization framework. The mathematical model for the stochastic optimization is given by

$$\omega_n \leftarrow \omega_{n-1} + [\hat{g}_n(\theta_n) - d_n] \tag{4.4}$$

$$d_n \leftarrow Q(\omega_{n-1}) \tag{4.5}$$

$$\theta_n \leftarrow \theta_{n-1} - \epsilon d_n \tag{4.6}$$

where all the variables are in vector format of dimension p. For a single-level quantizer the quantization function $Q(.)$ is given by $d_n = sgn(.)$, in which case d_n becomes a Bernoulli sequence $\{\pm 1\}$. We can replace Δ_n by d_n. As d_n is a single bit quantizer, eq. (4.4) can be written as

$$\omega_n = \omega_{n-1} + \left[\frac{L(\theta_n + c_n d_n) - L(\theta_n - c_n d_n)}{2c_n d_n^k} - d_n \right] \tag{4.7}$$

which gives

$$\omega_n = \omega_{n-1} + [L(\theta_n + c_n d_n) - L(\theta_n - c_n d_n) - 2c_n] \frac{d_n}{2c_n} \tag{4.8}$$

The above equation shows that there is no need to generate the random direction vector Δ_n as the output bit stream of the modulator itself can be used as the direction of perturbation.

Fig. 4.2 shows that infact for the optimization of a simple enough loss function, the performance of the $\Sigma\Delta$ SPSA gradient descent algorithm is as good as an ideal gradient descent algorithm with respect to the rate of convergence. The loss function under consideration is

$$L(t; \theta_1, \theta_2) = \frac{1}{2} \left(y(t) - \theta_1 x_1(t) - \theta_2 x_2(t) \right)^2 \tag{4.9}$$

where the input signal $y(t) = \mathrm{a}_1 \sin(\omega_1 t) + \mathrm{a}_2 \sin(\omega_2 t)$ was choosen as a mixture of two

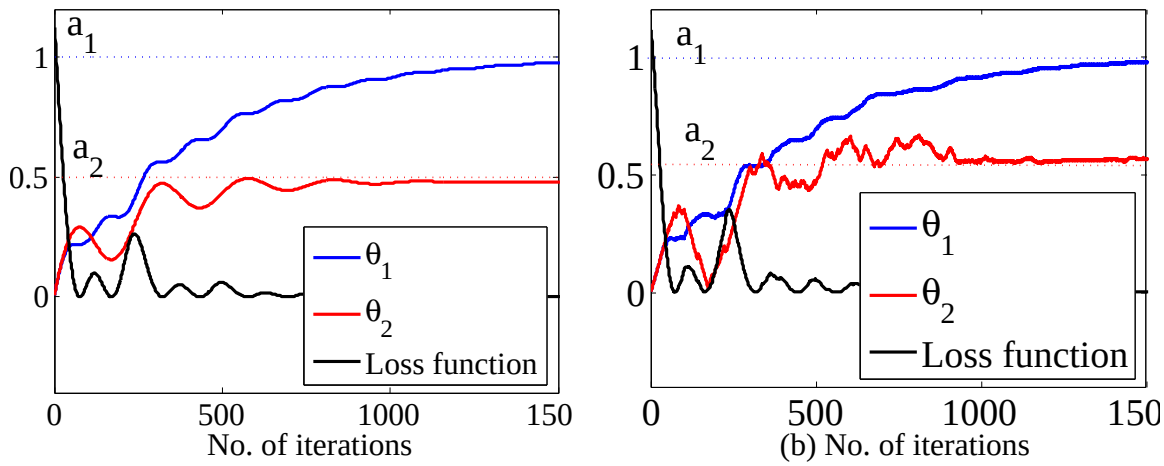


Figure 4.2: *The loss function along with the optimized parameters θ_1,θ_2 for a 2-variable optimization problem using (a) true gradient descent optimization and (b) ΣΔ SPSA gradient descent optimization*

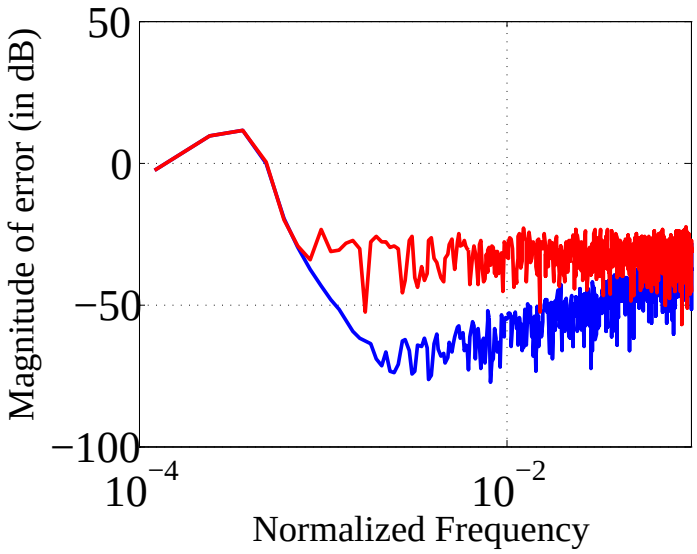


Figure 4.3: *Error magnitude (in dB) of the tracking system optimized with ΣΔ SPSA gradient approximation(in blue) and quantized SPSA gradient approximation(in red) methods*

different tones at ω_1 and ω_2. The optimization aims at finding the magnitude of a particular frequency signal present in the input $y(t)$. From the simulations in fig 4.2 it can indeed be seen that the ΣΔ SPSA gradient algorithm also converges the values of θ_1,θ_2 to a_1,a_2 respectively.

Fig 4.3 shows the performance of the $\Sigma\Delta$ gradient descent algorithm with gradient estimated from the SPSA as opposed to the quantized version of the SPSA gradient. The noise-shaping characteristics of the $\Sigma\Delta$ learning algorithm discussed in the previous chapters is preserved even if the gradient is approximated through stochastic perturbation methods.

Chapter 5

Application to Analog Circuit Calibration

As opposed to digital circuits, the performance of fabricated prototypes of analog circuits usually has a high degree of variance from the expected nature. This effect is even more pronounced in subthreshold analog circuits. But low-power operation and direct interfacing to the real world signals make analog circuits indispensable in any practical design. Dependance of transistor characteristics on temperature,process variations and mismatches in different components on the die make calibration of analog circuits even more difficult [23]. Digitally-assisted calibration techniques have been proposed in literature to calibrate analog circuits [21], [22]. But the use of A/D conversion introduces unnecessary quantization noise in the calibration process making them not feasible for use in calibrating analog circuits operating in subthreshold region because of the high amount of non-linearity and mismatch in such systems.

In this chapter, we present the hardware realization of the noise shaping optimization

framework discussed in previous chapters.We introduce it as an online learning algorithm to calibrate a feature extraction unit that was developed for speaker verification purposes. The feature extraction unit consists of a bandpass filter stage, a rectification stage and a $\Sigma\Delta$ modulator is used to measure the magnitude envelope of the speech signal in each frequency band. Each of the stages is described in detail and the problems arising in the calibration of the $\Sigma\Delta$ modulator and the bandpass filter stages are discussed.The variability of the circuit performance especially in subthreshold region of operation due to mismatch in device parameters is also studied. The optimization framework is validated on the fabricated chip for two different cases, (a)for balancing an unbalanced $\Sigma\Delta$ modulator and (b)the center frequency calibration of a bandpass filter.

5.1 Analog Circuit Design of a Spectral Feature Extractor

The block diagram of a single-channel in analog spectral feature extraction unit along with the calibration circuit is shown in figure 5.1. It consists of a bandpass filter stage, rectification stage and a $\Sigma\Delta$ modulator. The bandpass filter captures only the in-band frequency component of the input speech signal. The rectification stage gives the magnitude envelope of the filtered speech signal. The modulator in each channel generates a bit-stream proportional to the energy of the speech signal in that particular frequency band. This bit-stream is used as features for speaker verification purposes. Each of the above blocks are programmed with the help of serial-in current DACs.

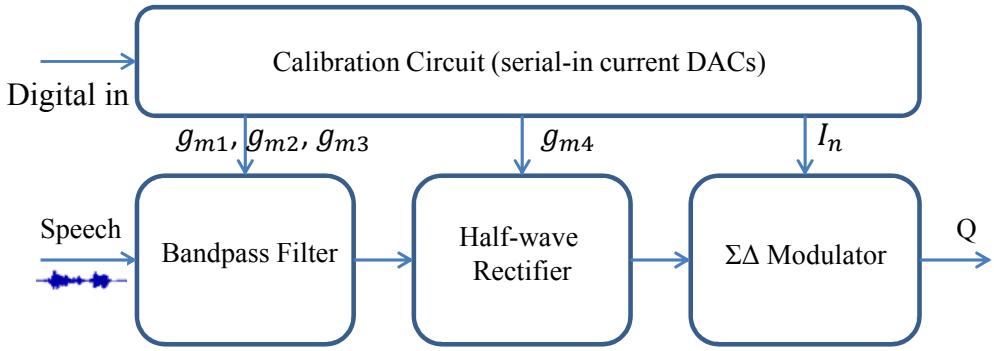


Figure 5.1: *Block diagram of the feature extraction unit used in speaker verification system*

5.1.1 Bandpass Filter Realization

Figure 5.2 shows the realization of the bandpass filter stage using $gm - C$ filter design. The transconductors used for realization of the filter are described in the next section. The output current of each of the transconductors in fig. 5.2 are given by:

$$i_1 = g_{m1} V_{in} \tag{5.1}$$

$$i_2 = g_{m2}(V_{lp} - V_{bp}) \tag{5.2}$$

$$i_3 = -g_{m3} V_{bp} \tag{5.3}$$

The currents through the capacitors C_1,C_2 are given by

$$i_1 + i_2 = C_1 s V_{bp} \tag{5.4}$$

$$i_3 = C_2 s V_{lp} \tag{5.5}$$

Putting together eqns. (5.1)- (5.5) and solving for the bandpass filter output $v_{bp}(s)$, we

get

$$V_{bp}(s) = \frac{\frac{g_{m1}}{C_1}s}{s^2 + \frac{g_{m2}}{C_1}s + \frac{g_{m2}g_{m3}}{C_1C_2}}V_{in}(s) \tag{5.6}$$

The transfer function of a generic second-order bandpass filter with center frequency ω_0, quality factor Q and gain G is given by

$$H(s) = \frac{G\frac{\omega_0}{Q}s}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2} \tag{5.7}$$

Comparing eqn. (5.6) and (5.7), we have the expressions for center frequency ω_0, gain G and quality factor Q as following:

$$\omega_0 = \sqrt{\frac{g_{m2}g_{m3}}{C_1C_2}} \tag{5.8}$$

$$Q = \sqrt{\frac{C_1g_{m3}}{C_2g_{m1}}} \tag{5.9}$$

$$G = \frac{g_{m1}}{g_{m2}} \tag{5.10}$$

Equations (5.8)- (5.10) show that there is a non-linear relationship between biasing parameters g_{m1}, g_{m2}, g_{m3} and the desired performance of the bandpass filter given by ω_0,Q and G. This gives rise to difficulty in programming the bandpass filter to its desired operating point. Considering the problems with subthreshold operation of the filters and the DACs used to calibrate them, this non-linear relationship increases the complexity of programming.

5.1.2 Transconductor

Fig. 5.3 shows the schematic diagram of the transconductor circuit used in the realization of the bandpass filter described above. The output current of the transcondutor is di-

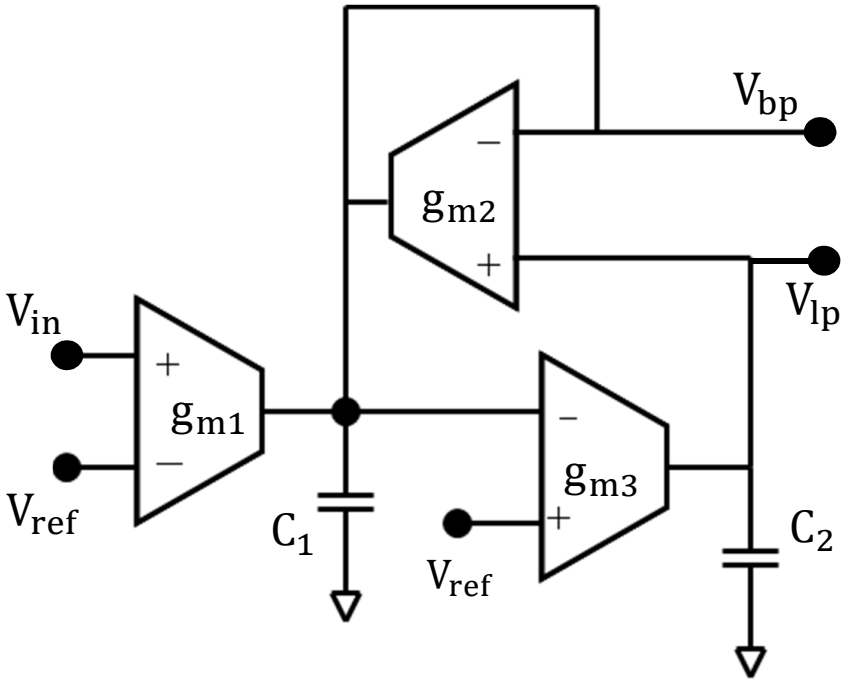


Figure 5.2: *Schematic diagram of the gm-C biquad filter with the bandpass filter output*

rectly proportional to the differential input signal ($V_{in} - V_{ref}$) and the bias current of the transconductor(I_{bias}). The output stage uses a cascaded structure (M4,M6,M7 and M8) in order to increase the output impedance of the transconductor. This helps in reducing the loading effect on the output current of the transconductor. The range of operation of the bandpass filter described in fig. 5.2 directly depends on the linear range of operation of the transconductor. Several topologies have been studied in literature [24] to improve the linearity of the transconductor operation region. In our circuit, a bump circuit proposed in [24] has been used. The bump circuit (formed by transistors $MB1 - MB4$) draws part of the input bias current away from the input transistors ($M3 - M4$), thereby reducing their g_m. This improves the linear range of the transconductor. The linear range of this transconductor has been shown to be around $300mV$ in saturation region [26].

The transconductance and also the linear range of a FET-OTA have a square root rela-

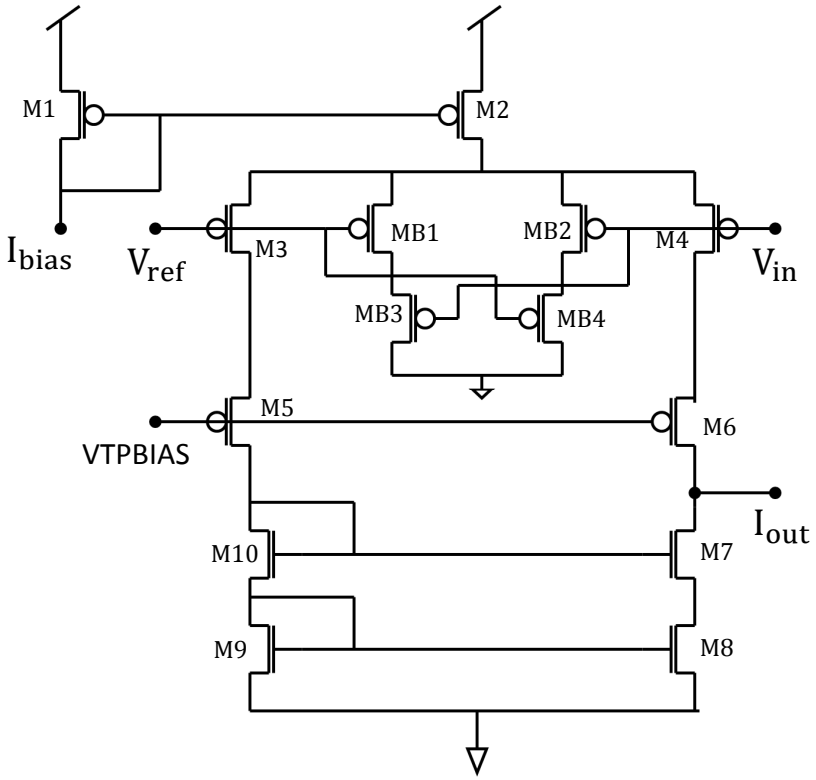


Figure 5.3: *Schematic diagram of the transconductor circuit used in the bandpass filter*

tionship [25] to the input bias current when operating in the saturation region. The tuning of the bandpass filters over the audible frequecy range (50Hz to 4KHz) require the bias currents of the transconductors to change $10pA$ to $4nA$, which is in the subthreshold region of operation. The linear range of operation decreases even more dramatically in this region. Figure 5.4 shows that the linear range of the transconductor is only about $\pm 120mV$ for subthreshold currents of the order of $100pA$. This reduces the linear range of the bandpass filter input voltage even further. This is because, the input voltage across the transconductor g_{m2} is the difference between the lowpass V_{lp} and the bandpass V_{bp} output stages. Therefore, for higher values of the quality factor (greater than 3), g_{m2} is no longer operating in linear region, and the relationship in eq. (5.2) doesn't hold good anymore. The whole system is

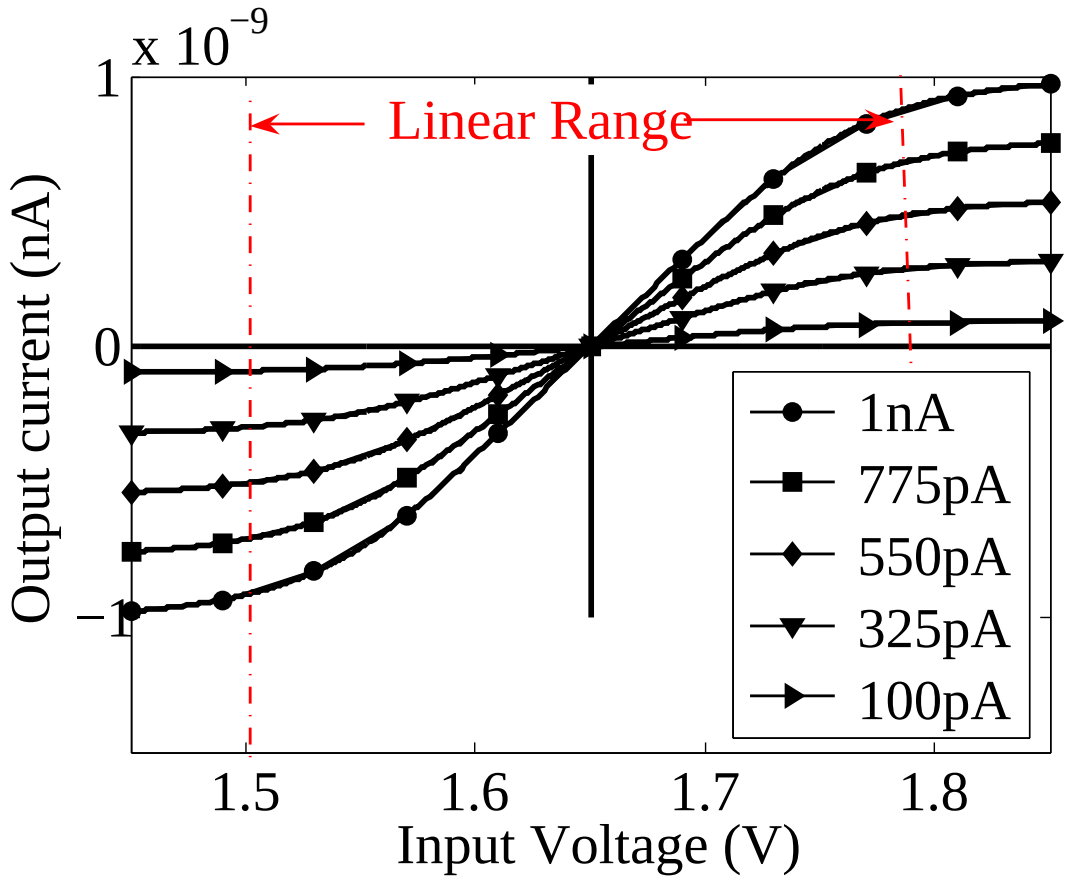


Figure 5.4: *DC Analysis of the transconductor Output current for different values of the bias currents (in subthreshold region)*

no longer a linear system. As a work around to this problem, a simple capacitive voltage attenuator as shown in fig. 5.5 has been used. The attenuation at this stage is given by

$$\frac{V_{fg}}{V_{in}} = \frac{C_1}{C_1 + C_2 + C_{gs}} \tag{5.11}$$

where C_{gs} is the sum of the gate-to-source parasitic capacitances of all the transistors connected to the floating gate output node V_{fg}. For $C_1 = 80fF$ and $C_2 = 320fF$, the input to the filter stage was seen to attenuate by a factor of 10, thereby increasing the dynamic range of operation of the bandpass filter. A separate source of injection current V_{inj} is also provided to program the floating gate to V_{ref}.

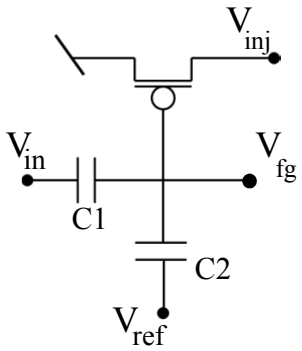


Figure 5.5: *Voltage Attenuator to increase the linear range of operation of the bandpass filter*

5.1.3 Current DAC

Each of the transconductors used in the bandpass filter are biased through a current DAC.Figure 5.6 shows the schematic of a 10-bit current DAC. The current division in the DAC is done by a standard MOS resistive network. The transistors of each stage of the DAC are so sized that the current divides by half, thereby generating a binary current DAC. The DACs are programmed through a serially connected shift-register. Depending on the digital bit stream $(b0 - b9)$ that is programmed, the current in each branch is either bypassed or gets added to the I_{DAC}. The current division at each stage is accurate only if all the transistors operate in saturation region. But as the DAC resolution increases (for LSB bits), the transistors operate at lesser and lesser currents due to current division and therefore go out of saturation region. This leads to non-linear operation of the current DAC. The monotonic nature of the DAC response has been characterized in [26] for saturation region.

For input bias currents I_{bias} in subthreshold region, the current division at each stage is no longer accurate. Infact, this effect is so pronounced in subthreshold region that the DAC ceases to be not only linear but also monotonic. This can be observed from fig. 5.7 which shows the performace of a 6-bit DAC in both saturation region(in red, for $I_{bias} = 5\mu A$) and

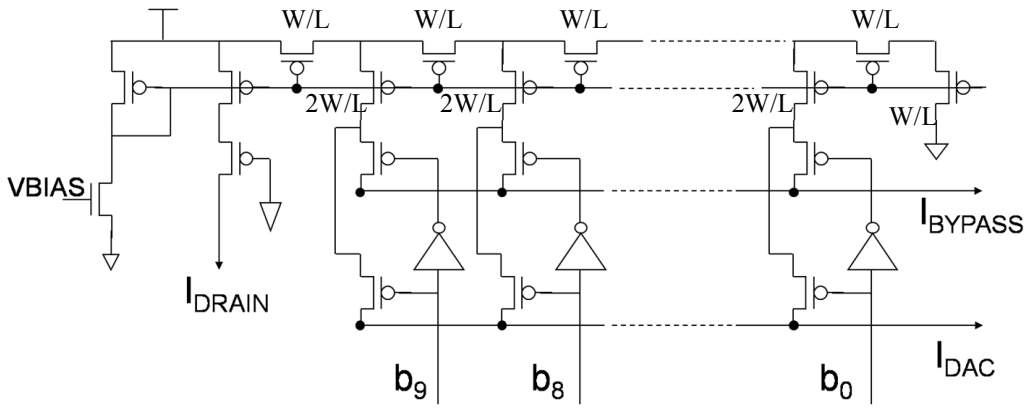


Figure 5.6: *Schematic diagram of the 10-bit current DAC providing the bias current for each transconductor*

subthreshold region (in black, for $I_{bias} = 1.3nA$).When a large number of input bits flip say from 011111 to 100000, the compounded effect of inaccurate current division leads to a huge non-monotonicity in the output of the DAC which is clearly seen in the fig. 5.7.

The effect of this non-monotonic behavior of the current DACs on programmability of the bandpass filter is shown in fig 5.8. Three different 10 bit current DACs are used to program the bias currents of the transconductors of the bandpass filter in fig 5.2. The quality factor and the gain are set to unity by ensuring that all the three DACs have the same 10-bit input. The DAC inputs are swept from 0 to 1023, and for each value of the input, the response of the modulator is plotted. The modulator output is lowpass filtered by averaging the bitstream and the DC value of the output is plotted showing the non-monotonic behaviour of the filter response because of the DAC currents.

5.1.4 Halfwave Rectifier

The filter output from the bandpass filter is passed through a half-wave rectifier. This rectification stage extracts the magnitude envelope of the speech signal present in that particular

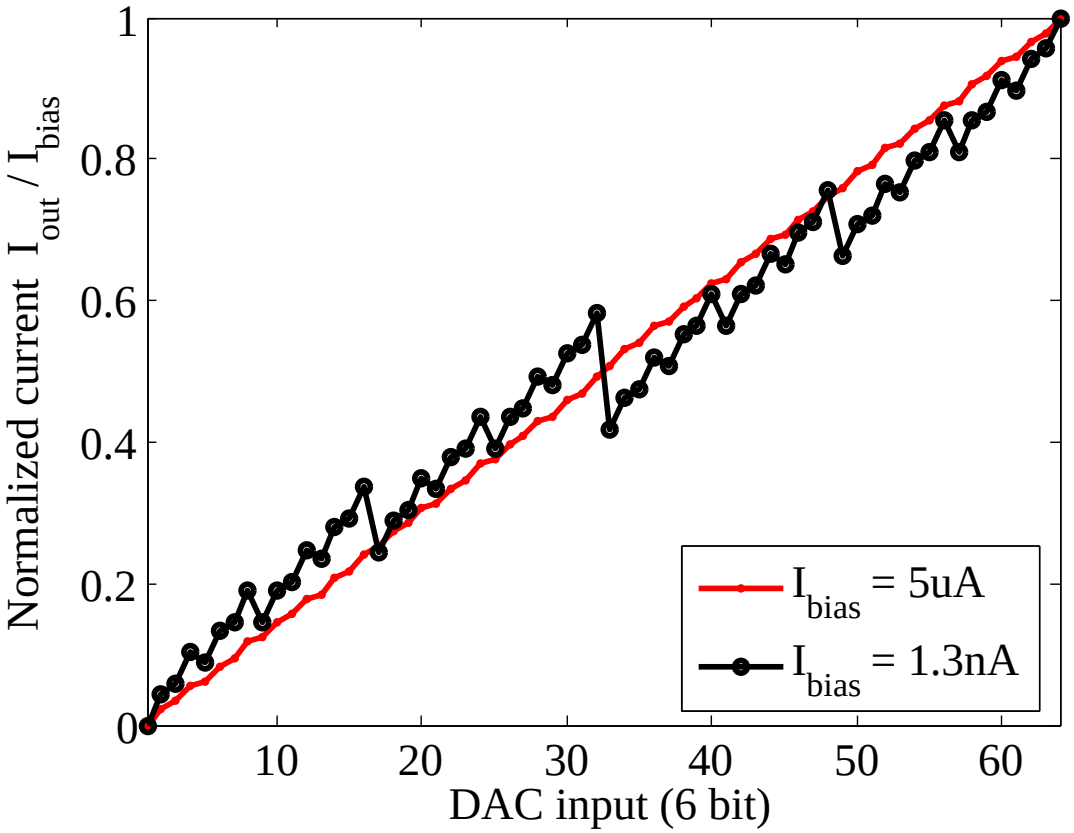


Figure 5.7: *Output current vs. input bit of a 6-bit DAC showing the non-monotonic nature of current DAC in subthreshold region of operation*

frequency band. The rectification stage consists of a transconductor which converts the output of the bandpass filter stage V_{bp} to current, with a diode connected PMOS at the output as shown in Fig. 5.9. The diode connection ensures that the current is only fed into the $\Sigma\Delta$ converter. The negative feedback across the amplifier in the $\Sigma\Delta$ modulator stage ensures that the output node I_{out} is held at V_{ref} provided the amplifier is strong enough. The output impedance of the transconductor is increased by the addition of a cascaded stage as in fig. 5.3. This enables a precise current summation at the input stage of the $\Sigma\Delta$ modulator.

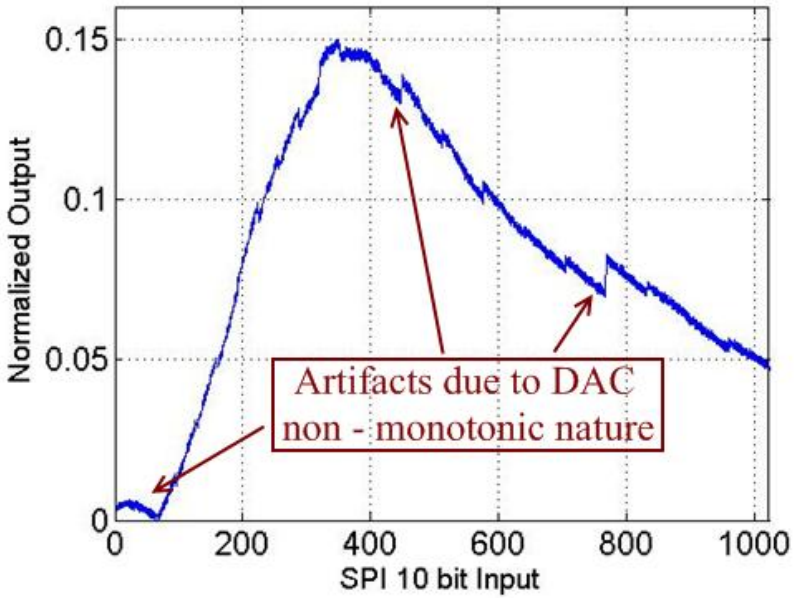


Figure 5.8: *The bandpass characteristics of the filter biased with subthreshold current DAC's at a unity gain and $Q = 1$.*

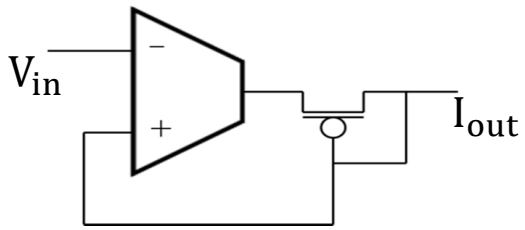


Figure 5.9: *Schematic diagram of the half-wave rectifer used in the feature extractor*

5.1.5 ΣΔ Modulator

Figure. 5.10 shows the schematic diagram of the first-order current mode ΣΔ modulator. It is used to measure the magnitude of the rectified signal from the half-wave rectification stage described above. The reference currents for the ΣΔ modulator are generated by a cascaded transistor pair $M2, M3$ and $M4, M5$. The positive reference current Ip generated by $M2, M3$ acts as a current source and the negative reference current In acts as a current sink. A standard fold-cascoded opamp is used for the integration stage of the modulator.

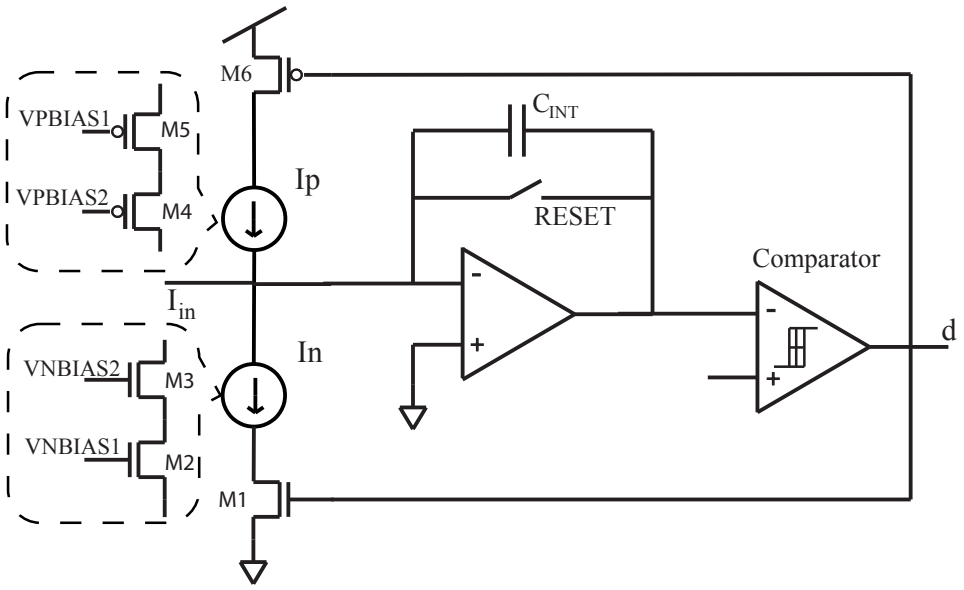


Figure 5.10: *The schematic diagram of the* $\Sigma\Delta$ *modulator*

In order to ensure the correct functionality of the integrator, the opamp was biased to have an open-loop gain of atleast 60dB. The quantization stage of the modulator consists of a hysteretic comparator which produces 1-bit quantization. The hysteresis comparator is formed by using a simple differential amplifier followed by current starved inverter. The performance of the $\Sigma\Delta$ modulator along with the hysteretic comparator has been previously studied [27] and is not a subject of discussion here. It has been shown that the modulator has about 10-bit resolution over a wide input current range ($50fA - 100nA$).

For a completely balanced $\Sigma\Delta$ modulator, Ip=In=$Iref$. The integrator voltage in this case at time instance i with a sampling rate T_s is given by

$$V_{i+1} = V_i + \frac{T_s}{C_{INT}}(I_{in} - d_i Iref) \tag{5.12}$$

where $d_i = sign(V_i)$. Taking the summation of the above equation over N iterations and taking the limit as $N \rightarrow \infty$, we have

$$\sum_{i=1}^{N} d_i = \frac{I_{in}}{Iref} \tag{5.13}$$

which shows that the pulse modulated bit stream d_i gives a faithful representation of the input current I_{in} provided the sampling rate T_S is much higher than the bandwidth of the input current. The case of an unbalanced $\Sigma\Delta$ modulator where $Ip \neq In$ is dealt with in detail later. The calibration of an unbalanced $\Sigma\Delta$ modulator wherein the optimization technique described in Chapter.2 is applied to self-calibrate a $\Sigma\Delta$ modulator is also presented in the later sections along with the results from hardware testing.

5.2 Test Station Setup

The block diagram of the test station used to characterize the chip is shown in figure 5.11. The experimental setup consists of an OpalKelly XEM3010 FPGA which is used for generating all the digital clock pulses required for the chip. The state machine running on the FPGA can be controlled through Matlab and C interface from the computer. The FPGA is also used for storing the pulse encoded bit stream generated from the $\Sigma\Delta$ modulator. The FPGA has an integrated 32MB SDRAM which was used for this purpose. The FPGA has an USB interface for high-speed data transfer with PC. The collected data is later post-processed and analyzed in Matlab.

All the analog biases for the chip are programmed through the National Instruments data acquisition card. The custom made motherboard has 40 different analog output ports

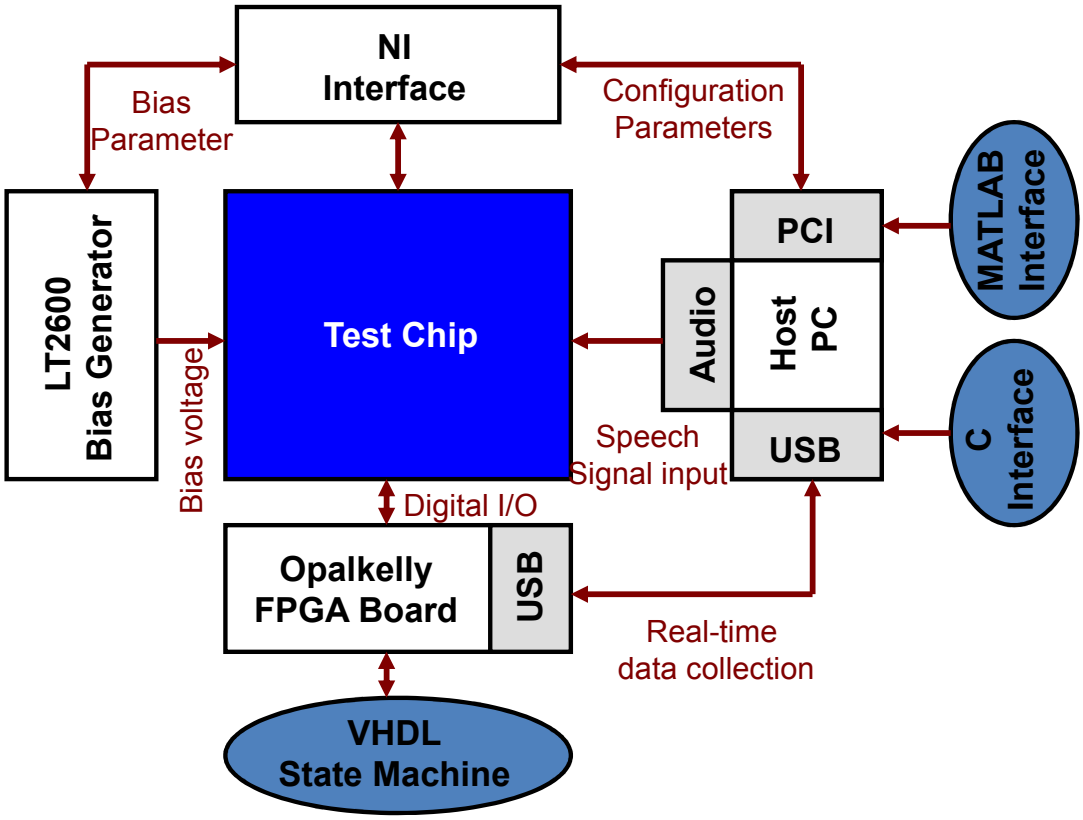


Figure 5.11: *System level diagram of the test station setup*

each of which can be digitally programmed through the NI card. These analog output ports are generated by 5 different 16-bit DACs (LTC2600) present on the motherboard each of which have 8 analog outputs. The motherboard also generates reference voltage for these DACs and also the power supply for the test chip. The daughter board offers the flexibility of interfacing the test station setup with any design under test. Figure 5.12 shows the actual setup of the test station. The motherboard, FPGA, daughterboard and the test chip can be clearly seen in the figure. Figure 5.13 shows the micrograph of the fabricated chip with the bandpass filter stage (a) and the current DAC (b). The prototype has been fabricated on a 0.5μm CMOS process from MOSIS. A single channel of the bandpass filter occupies

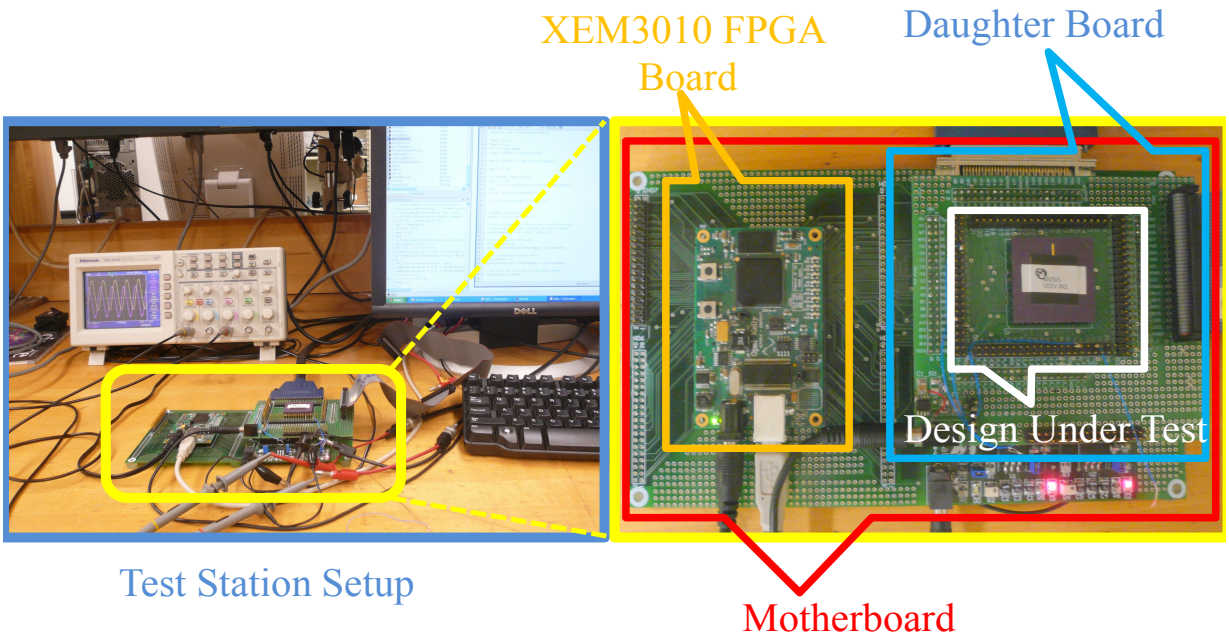
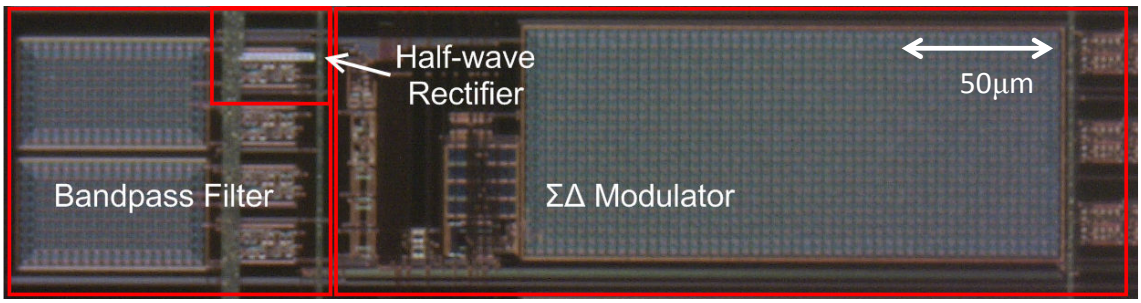


Figure 5.12: *Test station setup showing the interface between the Opalkelly XEM3010 FPGA, NI data acquisition card, motherboard and the test chip*

100μm X 300μm area. Table 5.1 summarizes the performance of the fabricated prototype. The most power hogging element in the whole design was observed to be the $\Sigma\Delta$ modulator. The filter and the rectifier were consuming less than $300nW$ of power.

5.3 Calibration of Sigma Delta Modulator

In this section, the unbalanced $\Sigma\Delta$ modulator is studied in detail. The noise shaping $\Sigma\Delta$ optimization introduced in Chapter.2 is extended to self-calibrate an unbalanced $\Sigma\Delta$ modulator. The results from the simulation and the hardware are also presented. Equation (5.12) shows the functionality of a balanced $\Sigma\Delta$ modulator. For an unbalanced $\Sigma\Delta$ modulator, the



(a)

(b)

Figure 5.13: *Micrograph of (a) single channel feature extractor with bandpass filter,rectifier and ΣΔ modulator stages and (b) the 10-bit current DAC fabricated on a 0.5µm CMOS process*

integrator voltage at time instant i is given by

$$V_{i+1} = V_i + \frac{T_s}{C_{INT}} \left(I_{in} - d_i Iq \right) \tag{5.14}$$

$$d_i = sgn(V_i) \tag{5.15}$$

where Iq is the pulse averaged reference current given by $Iq = q_i In + (1 - q_i) Ip$. Here, q_i is the bitstream d_i in 0's and 1's given by $q_i = \frac{(1+d_i)}{2}$. Taking the summation of eq (5.14) for N iterations,

$$\frac{V_N - V_0}{N} = \frac{T_s}{C_{INT}} \left(I_{in} - \sum_{i=1}^{N} d_i \left(q_i In + \left(1 - q_i \right) Ip \right) \right) \tag{5.16}$$

Table 5.1: *Measured specifications of the fabricated single-channel bandpass filter chip*

Technology	0.5μm CMOS process
Single Channel size	1000μm X 300μm
Power - Bandpass filter	150nW
Power - Rectifier	100nW
Power - Current DACs	150nW
Power - $\Sigma\Delta$ modulaor	40μW
Tunable Frequency Range	10Hz - 10KHz
Input Range	400mV
Quality Factor Q	3 (max)
Transconductor Linear Range	240mV
Input Pulse Rate	250KHz

Assuming V_i is bounded, as $N \rightarrow \infty$ we have

$$I_{in} = \left(\frac{Ip + In}{2}\right) \frac{1}{N} \sum_{i=1}^{N} d_i - \left(\frac{Ip - In}{2}\right) \tag{5.17}$$

Equation (5.17) shows that for an unbalanced $\Sigma\Delta$ modulator, the average of the bit-stream at the modulator output doesn't truthfully represent the input current I_{in}. There is a finite error in the measurement of the input current given by the last term in eq. (5.17). Infact, for zero input current $I_{in} = 0$, the modulator has a non-zero offset given by

$$d_{off} = \frac{1}{N} \sum_{i=1}^{N} d_i = \frac{Ip - In}{Ip + In} \tag{5.18}$$

It can clearly be seen that the offset goes to zero for a balanced $\Sigma\Delta$ modulator.

Figure 5.14 shows the performance of an array of $\Sigma\Delta$ modulators biased with the same input voltages. The current references for each of the modulator are generated by the cascaded MOS pair as described in fig. 5.10. Even though the $\Sigma\Delta$ modulators operate at the egde

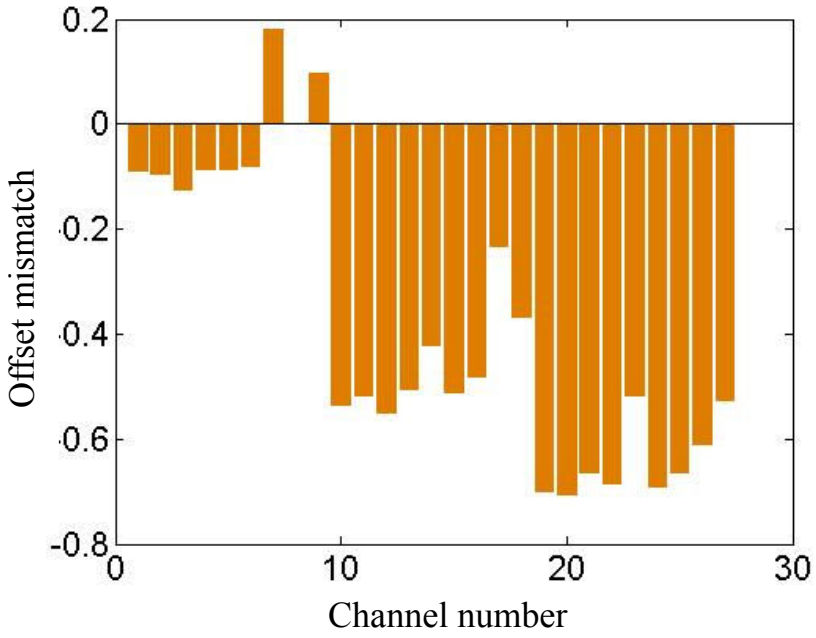


Figure 5.14: *Offset mismatch between individual channels of an array of* $\Sigma\Delta$ *modulators for constant input voltage across all the channels*

of saturation region (at about 1μA current), we see that the offset variation is as high as 70 percent in some of the channels. All the modulators are biased with the same input voltage, but because of the differences in routing length of the metal lines supplying input voltages $VNBIAS1$ and $VPBIAS1$ in the chip for each modulator, there is a significant IR drop before the voltage appears at each of the modulator. This voltage difference between individual modulators results in unbalanced reference currents causing the offset shown.

The offset described above can lead to erroneous estimates of the input current measurement. This current offset can be cancelled by extending the $\Sigma\Delta$ optimization technique described in earlier chapters to self-calibrate a $\Sigma\Delta$ modulator.

Lemma 1: For an unbalanced modulator with $Ip \neq In$ and a zero input current I_{in}, the

iterative update

$$In_{i+1} = In_i + \epsilon_i d_i \tag{5.19}$$

converges In to Ip for large enough value of N.

Proof: For input current $I_{in} = 0$, eq. (5.14) can be written as

$$V_{i+1} = V_i + \frac{T_s}{C_{INT}} \left(-d_i \left(q_i In + (1 - q_i) Ip\right)\right) \tag{5.20}$$

Let $I_{error} = In - Ip$. Substituting it in above equation,

$$V_{i+1} = V_i + \frac{T_s}{C_{INT}} \left(-d_i \left(q_i Ip + q_i I_{error} + (1 - q_i) Ip\right)\right) \tag{5.21}$$

Observing that $d_i q_i = d_i$, we have

$$V_{i+1} = V_i + \frac{T_s}{C_{INT}} \left(-d_i I_{error} - d_i Ip\right) \tag{5.22}$$

Comparing the above equation with eqn. (5.12), we observe that the effective input current $\hat{I}_{in}$ is given by

$$\hat{I}_{in} = -d_i I_{error} \tag{5.23}$$

The $\Sigma\Delta$ gradient descent algorithm introduced earlier converges the objective function $\frac{1}{2}\hat{I}_{in}^2$ to its minimum for large enough number of iterations N. The minimum is obtained when the expected value of the gradient $\hat{I}_{in} = 0$, i.e.,

$$E\left(\hat{I}_{in}\right) = E\left(-d_i I_{error}\right) = 0 \tag{5.24}$$

The correlation between d_i and I_{error} means that this happens only when $E[I_{error}] = 0$ and $E[d_i] = 0$. So, for a large enough N,

$$I_{error} \overset{n\to\infty}{\longrightarrow} 0 \tag{5.25}$$

The above equation shows that $In \overset{n\to\infty}{\longrightarrow} Ip$ which is a balanced $\Sigma\Delta$ modulator.

The performance of the above $\Sigma\Delta$ calibration has been shown in fig. 5.15. The objective function being minimized $\frac{1}{2}\hat{I}_{in}^2$ is plotted by varying the 10-bit current DAC input bits from 0 to 1023. Figure 5.15(a) shows the loss function for $Ip = 0.7I_{bias}$, where I_{bias} is the current bias of the DAC. The presence of a large number of local minimum can be seen from the figure. Figure 5.15(b) shows the convergence of the modulator output to zero because of the adaptation in eq. (5.19). It can be seen clearly that despite the presence of other local minimum, the optimization converges to the global minimum at the DAC input of around 780.

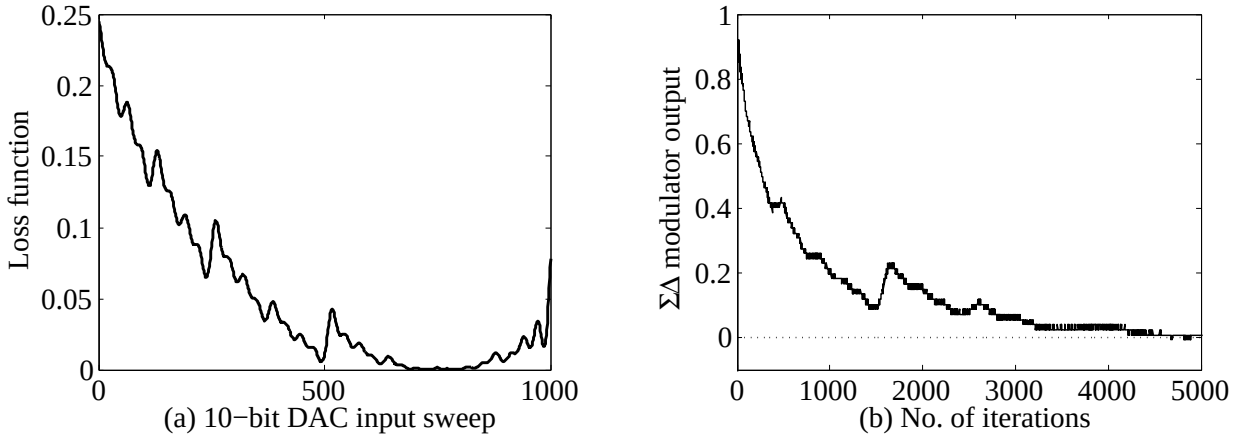


Figure 5.15: *Figure showing (a) the loss function $\frac{1}{2}\hat{I}_{in}^2$ as a variation of the 10-bit current DAC sweep, (b) the convergence of the modulator output to zero*

Figure 5.16 shows the verification of $\Sigma\Delta$ calibration on hardware. The $\Sigma\Delta$ modulator

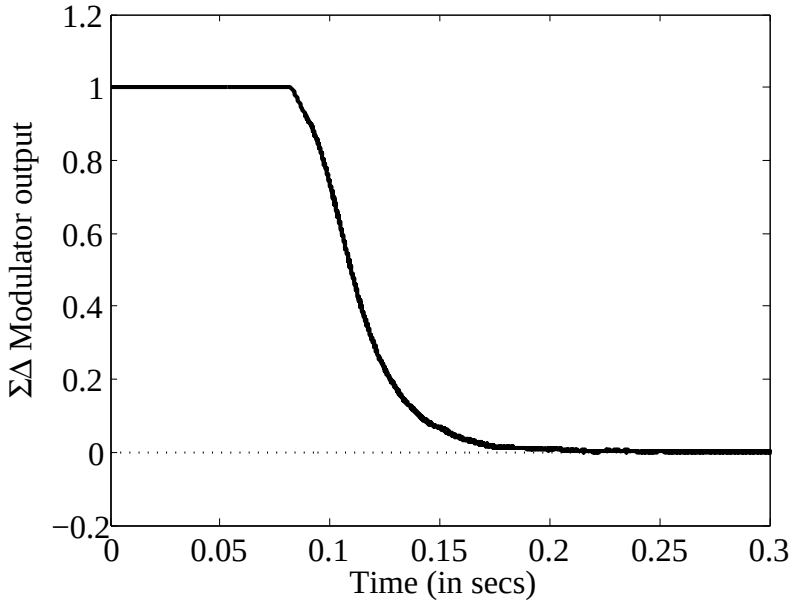


Figure 5.16: *Convergence of the modulator output due to In adaptation from the bandpass filter chip*

output converges to zero once the negative current In is adapted. In this experiment, the reference currents are not generated by the DAC but by the voltage biases of the cascaded transistors shown in fig. 5.10. Transistors $M2, M3$ form a voltage($VNBIAS1$)controlled current source. For a given positive reference current Ip, the adaptation is done by programming $VNBIAS1$ using the Opalkelly FPGA. Depending on the modulator output d_i, the 16-bit DAC(LTC2600) on the motherboard is either increased or decreased by 1 LSB. The voltage $VNBIAS1$ ia initially set to 0 at $t = 0$.

5.4 Calibration of a bandpass filter

The performance of any analog IC design is greatly effected by the mismatch between different components within the chip. The design of high presicion analog circuits needs special care in order to mitigate the effect of mismatch. This mismatch is majorly because of the

Table 5.2: *Variation of the bandpass filter characteristics across two different chips operating under similar conditions*

-	Chip 1	Chip 2
Center Frequency ω_0	420 Hz	370 Hz
Quality Factor Q	3.2	0.8
Gain G	66.62 dB	68.6 dB

variability and reliability of the physical process used in fabricating the device. Analog circuits are also sensitive to ambient temperature. Also, the variation of the performance across different chips is a major problem in designing precise analog circuits. Several approaches ([28],[29]) have been used to model the mismatches between devices based on the drain-current relationship of a transistor in saturation region. Even a small device mismatch of transistors in weak inversion causes a huge performance variation across components. Figure 5.17 shows the variation in the performance of the bandpass filter characteristics across two different chips, both of which were biased under similar ambient conditions and same input bias. The variation across the chip shows that the mismatch can be as high as 50 percent among different components. One way to deal with the mismatch is to use an on-chip learning mechanism[30] that can auto-correct for the variation in performance of the chip to the desired output by using a feedback system.

In this section, we introduce the problem of calibration of a bandpass filter to a particular center frequency as an on-chip learning framework, by using the $\Sigma\Delta$ gradient descent approach. Figure 5.18 shows the block diagram of the bandpass filter calibration system. The bandpass filter described in previous section acts as a low-pass filter when the input signal is modulated on the reference voltage of the transconductor g_{m3} instead of V_{in1}.

Assuming that all the transconductors are operating in their linear region, the current

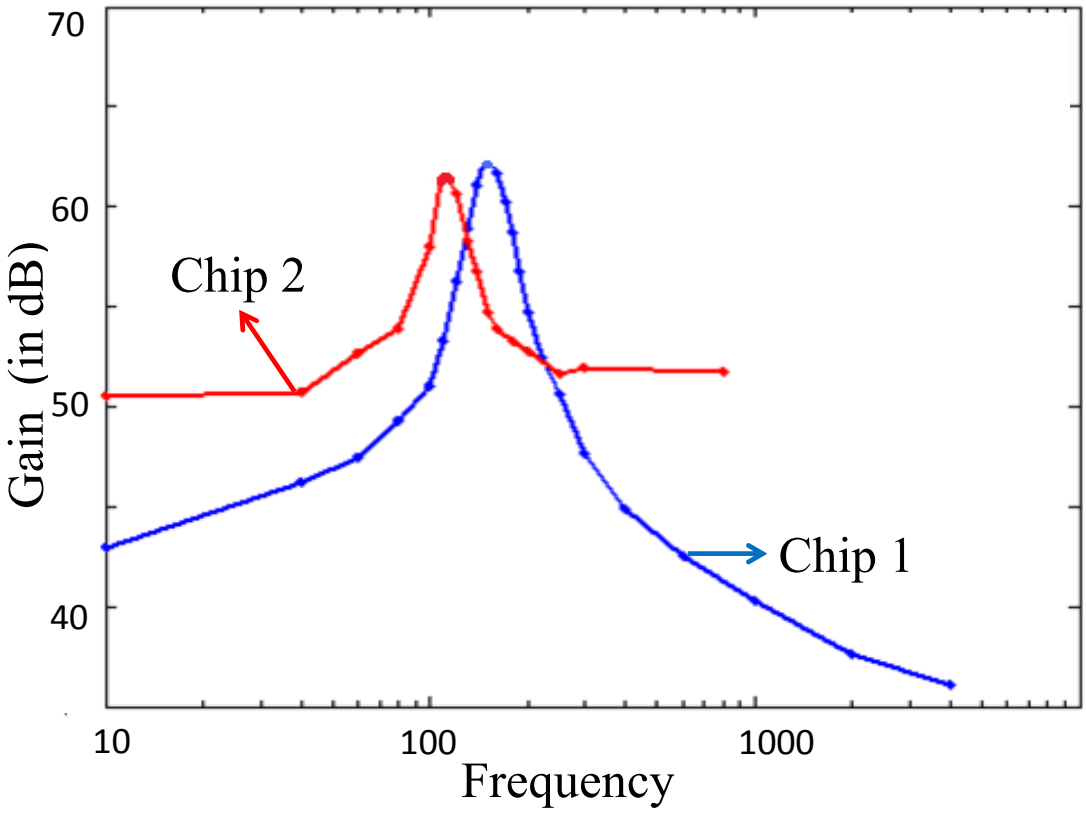


Figure 5.17: *Mismatch between the performance of bandpass filters from two different chips with same inputs under similar operating conditions*

equations for each of the transconductor are given by

$$i_1 = g_{m1}V_{in1} \tag{5.26}$$

$$i_2 = g_{m2}(V_{lp} - V_{out}) \tag{5.27}$$

$$i_3 = g_{m3}(V_{in2} - V_{out}) \tag{5.28}$$

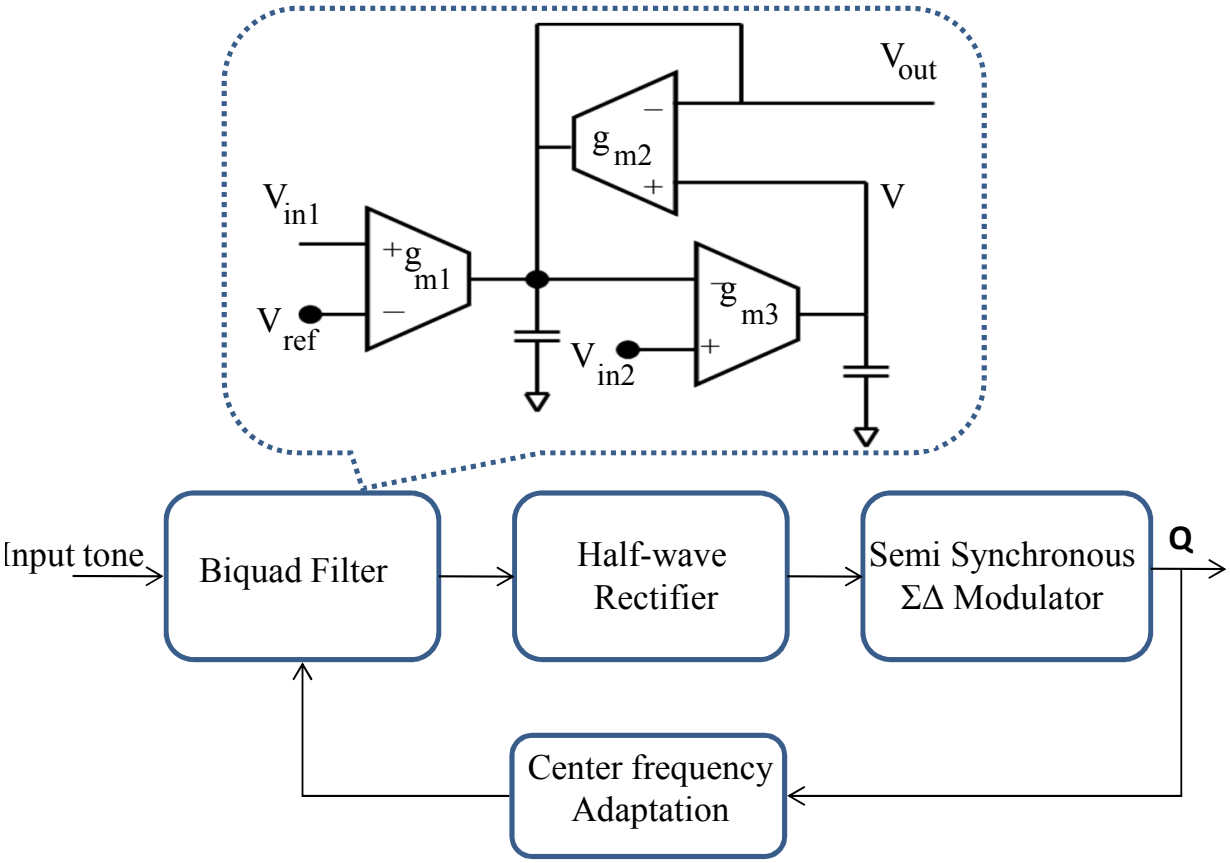


Figure 5.18: *Block diagram of a single channel feature extraction unit with center frequency calibration*

The currents flowing through each of the capacitor C_1, C_2 are given by

$$i_1 + i_2 = C_1 s V_{out} \tag{5.29}$$

$$i_3 = C_2 s V_{lp} \tag{5.30}$$

Solving for V_{out} from the above equation we have,

$$V_{out}(s) = H_{bp}(s)V_{in1}(s) + H_{lp}(s)V_{in2}(s) \tag{5.31}$$

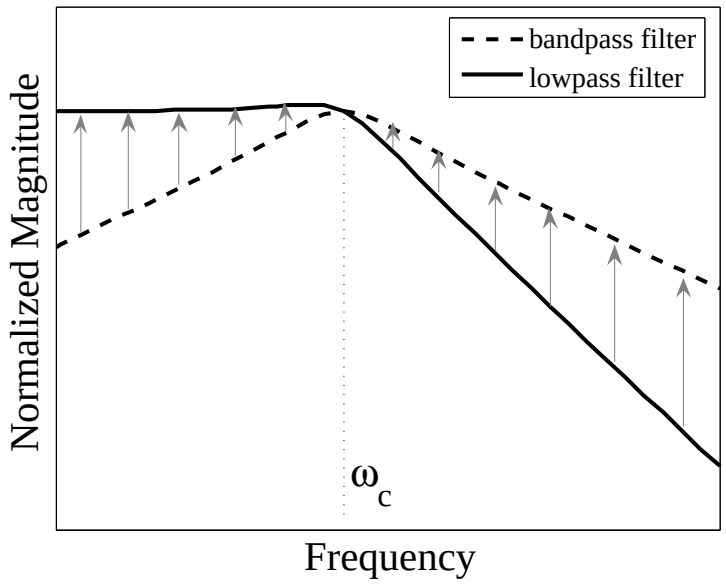


Figure 5.19: *Frequency response of the bandpass and the lowpass stages of the biquad filter showing the magnitude of the residual signal for input tone of different frequencies*

where

$$H_{bp}(s) = \frac{\frac{g_{m1}}{C_1}s}{s^2 + \frac{g_{m2}}{C_1}s + \frac{g_{m2}g_{m3}}{C_1C_2}} \tag{5.32}$$

and

$$H_{lp}(s) = \frac{1}{s^2 + \frac{g_{m2}}{C_1}s + \frac{g_{m2}g_{m3}}{C_1C_2}} \tag{5.33}$$

which shows that the systems has second-order lowpass filter characteristics w.r.t the input V_{in2}. The cutoff frequency of $H_{lp}(s)$ is the same as the center frequency of the bandpass filter $H_{bp}(s)$. From eq. (5.31), define a loss function $L(\omega)$ as

$$L(\omega) = \frac{1}{2}\Big(H_{bp}(\omega)V_{in1}(\omega) + H_{lp}(\omega)V_{in2}(\omega)\Big)^2 \tag{5.34}$$

The problem of calibration of the bandpass filter to a center frequency ω_0 reduces to optimization of $L(\omega)$.

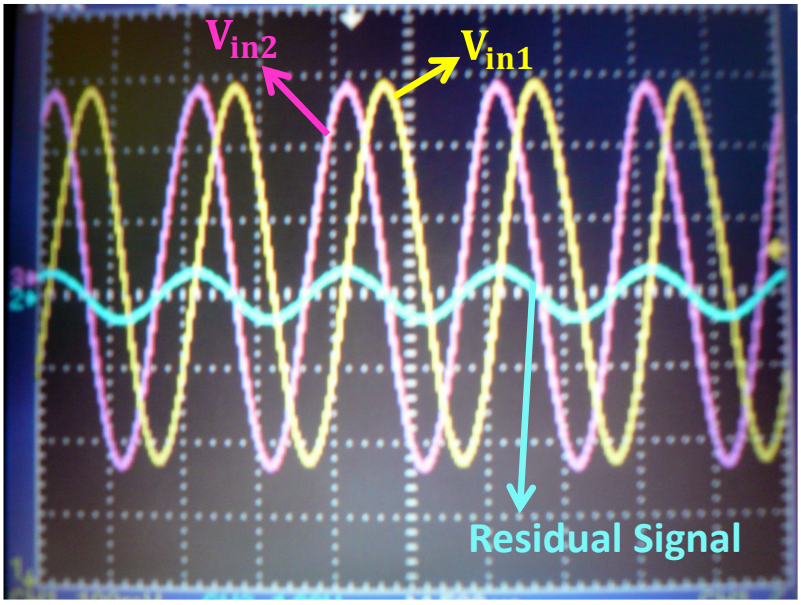


Figure 5.20: *Oscilloscope figure showing the input tones seperated by* 90^o *phase and the residual signal from the filter*

Figure 5.19 shows the frequency response of both the bandpass and the lowpass stages of the biquad filter for a unity quality factor. The magnitude of the residual signal from the filter goes to zero at the center frequency ω_0. For input frequencies lesser than ω_0, the lowpass filter stage has a unity gain and the bandpass filter gain tapers at 20dB/decade, resulting in a non-zero output signal at V_{out}. For frequencies greater than ω_0 the bandpass filter gain reduces by only 20dB/decade as opposed to 40dB/decade reduction of the lowpass filter gain. As a result for $\omega > \omega_0$, there is a non-zero residual signal at V_{out}. The minimum point at ω_0 can be reached by using a $\Sigma\Delta$ gradient descent algorithm using the loss function described in eq (5.34). The on-chip calibration method described above has been verified on hardware using the fabricated prototype. For this experiment, the bandpass filter is biased at unity gain G and quality factor Q. The input V_{in1} to the filter is a tone whose frequency is the same as the center frequency ω_0 to which the bandpass filter is to be calibrated.

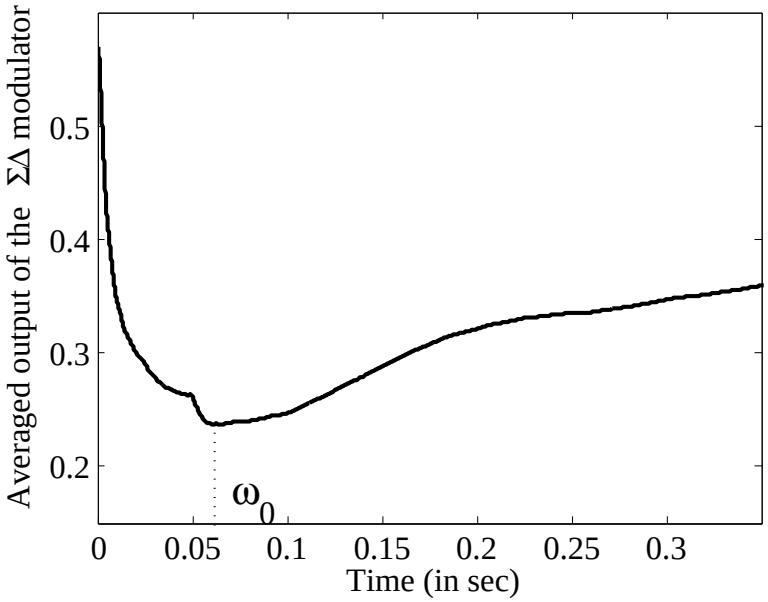


Figure 5.21: *Moving average of the ΣΔ modulator showing the minimum when center frequency reaches the frequency of the input tone*

Equation (5.32) shows that the bandpass filter has a zero at origin, thereby introducing a phase difference of 90^o at the output compared to V_{in1}. In order to facilitate the correct cancellation of the signal at the output due to the lowpass filter stage, the second input V_{in2} is phase shifted by the same 90^o with respect to V_{in1}. Figure 5.20 shows both the inputs to the filter and also the non-zero residual filter output V_{out} for an uncalibrated bandpass filter with center frequency not equal to the input tone ω_0.

Figure 5.21 shows the results of the calibration from the fabricated chip. The ΣΔ modulator was run at a sampling frequency of 250KHz. The input tone was set at ω=2KHz and the bandpass filter was initially programmed to around 1KHz using voltage biasing. The center frequency of the filter was adjusted according to the learning algorithm by using a FPGA which tuned the voltage biasing of the transconductors. The modulator output was filtered by taking a 1024 sample moving average of the bit-stream. Fig 5.21 shows that the modulator output does not go to zero after optimization. This is because of the presence of

a constant offset current from the rectifier due to imperfect cancellation of the inputs at the filter stage. Also, once the minimum point ω_0 was achieved, the modulator doesn't remain at that point because the rectifier does not estimate the true gradient of the loss function defined in eq. (5.34) but only gives the magnitude, which is always positive. The actual calibration point can be achieved by taking the summation of the bitstream up until the minimum point ω_0 is achieved.

Chapter 6

Conclusion

A novel $\Sigma\Delta$ gradient descent optimization algorithm has been presented in this work. The proposed learning algorithm has been shown to have better precision compared to the traditionally used noisy gradient descent algorithm by using the noise-shaping characteristics of the $\Sigma\Delta$ modulator in real-time tracking of system parameters. The stochastic nature of the noise in the quantization stage has been shown to be helpful in the case of global convergence as opposed to the ideal gradient descent algorithm which converges to local minimum. The proposed algorithm was later extended to include higher-order noise shaping and learning mechanisms. The performance improvement of the algorithm was shown through simulation on a sample speech utterance from YOHO database. The $\Sigma\Delta$ optimization framework was later extended to accommodate gradient-free optimization techniques like finite-difference stochastic approximation and simultaneous perturbation stochastic approximation.

The performance of the $\Sigma\Delta$ gradient descent algorithm has later been validated on hardware by modeling the problem of calibration of analog circuits as an optimization problem. The $\Sigma\Delta$ learning was shown to be robust enough to account for mismatches and

non-linearities even in subthreshold region of operation. The self-calibration of an unbalanced $\Sigma\Delta$ modulator converged to the global minimum even in the presence of several local minima points because of the non-monotonicity of the calibration DACs in the subthreshold region. The algorithm was also used to calibrate the center frequency of a gm-C biquad filter by exploiting the lowpass and bandpass filter stages present within the biquad filter.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] R. Hooke and T.A. Jeeves, "Direct search solution of numerical and statistical problems", J. ACM, 8 (1961), pp.212-229.

[2] M. J. D. Powell, "Direct search algorithms for optimization calculations"', Acta Numerica, 7 , pp 287-336(1998).

[3] Battiti R.," First- and Second-Order Methods for Learning: Between Steepest Descent and Newtons Method", Neural Computation, Vol. 4,141-156., 1992.

[4] Goldberg, E. David, "Genetic Algorithms in Search Optimization and Machine Learning", Addison Wesley, p41, ISBN 0201157675.

[5] D. Bertsimas and J. Tsitsiklis, "Simulated Annealing", Journal of Statistical Science, 1993, Vol 8, No. 1, pp 10-15.

[6] J. Kennedy, R. Eberhart, "Particle Swarm Optimization", IEEE International Conference on Neural Networks, Vol 4, pp. 1942 - 1948, Nov. 1995

[7] Battiti R.," Reactive Search: Toward Self-Tuning Heuristics", John Wiley & Sons Ltd., pages 61-83, Chichester, 1996.

[8] R. Battiti and M. Brunato, "The Reactive Tabu Search", ORSA Journal of Computing, 6(2):126-140, 1994

[9] Luo, Z.," On the convergence of the LMS algorithm with adaptive learning rate for linear feedforward networks," Neural Computation, Vol. 3, 226-245., 1991.

[10] S. Gelfald and S.K. Mitter, "Recursive Stochastic Algorithms for Global Optimization",J. Control and Optimization, Vol 29, pp 999-1018, Sept. 1991.

[11] Fang, H., Gng, G., and Qian M., "Annealing of Iterative Stochastic Schemes," J. Control and Optim., 35, pp. 1886-1907, 1997.

[12] S. Gelfald and S.K. Mitter, "Simulated annealing with noisy or imprecise energy measurements",J. Optim. Theor. Appl., 62:49-62, 1989.

[13] H. Robbins and S. Monro, " A Stochastic approximation method", Ann. Math. Statist., 22: 400-407, 1951.

[14] J.A. Nelder and R. Mead, "A simplex method for function minimization", Comput. J., 7 (1965), pp.308-313.

[15] H. H. Rosenbrock, "An automatic method for finding the greatest or least value of a function", Comput. J., 3 (1960), pp. 175-184.

[16] J. R. Blum, " Approximation methods which converge with probability one", Ann. Math. Statist., 25:382-386, 1954.

[17] L. Bottou, " Stochastic Learning", Advanced Lectures in Machine Learning, pages 146-168, Springer, 2003.

[18] J. Kiefer and J. Wolfowitz, "Stochastic estimate of a regression function", Ann. Math. Statistics., 23: 462-466, 1952

[19] J. C. Spall, "A stochastic approximation technique for generating maximum likelihood parameter estimates", Proc. Amer. Control Conf., 1987, pp. 1161-1167.

[20] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation", IEEE Trans. Automat. Control, 37: pp. 332-341, 1992.

[21] B. Murmann,"Digitally assisted analog circuits", IEEE Micro,Vol. 26, pp.38-47, Mar. 2006.

[22] B. Murmann, B. Boser,"A 12-bit 75-MS/s pipelined ADC using open-loop residue amplification", IEEE J. Solid-State Circuits, vol. 38, no. 12, pp.2040-2050, Dec. 2003.

[23] P. R. Kinget,"Device Mismatch and Tradeoffs in the design of analog circuits", in IEEE Journal of Solid-State Circuits, Vol. 40, Issue:6, pp.1212-1224, Jun 2005.

[24] P.M. Furth and H.A. Ommani. "Low-voltage highly-linear transconductor design in subthreshold CMOS", Proceedings of the 40th Midwest Symposium on Circuits and Systems, 1997. Volume 1, 3-6 Aug. 1997 Page(s):156 - 159 vol.1

[25] Achim Gratz, "Operational Transconductance Amplifiers".

[26] A. Gore, "Design of High-Dimensional Oversampling Data Converters with On-chip Learning : Theory, Algorithm and Hardware Realization".

[27] A. Gore, S. Chakrabartty, S. Pal and E. C. Alocilja, "A Multichannel Femtoampere-Sensitivity Potentiostat Array for Biosensing Applications," in IEEE Transactions of Circuits and Systems -I, Vol. 53, No. 11, November 2006.

[28] S. C. Wong et al., "A CMOS mismatch model and scaling effects," IEEE Electron Device Lett., vol. 18, pp. 261263, June 1997.

[29] S. Lovett et al., "Characterizing the mismatch of submicron MOS transistors," in IEEE Int. Conf. Microelectronic Test Structures, vol. 9, Mar.1996, pp. 3942.

[30] G. Cauwenberghs.,"Neuromorphic Learning VLSI Systems: A Survey"

[31] Yannis P. Tsividis,Ken Suyama.,"MOSFET Modeling for Analog Circuit CAD: Problems and Prospects," IEEE Journal of Solid-State Circuits, vol.29, no. 3,March 1994