OPTIMAL DESIGN OF DISCRETE-TIME DELTA SIGMA MODULATORS

by

Matthew Edward Jackson

Bachelor of Science in Electrical Engineering
University of Wyoming
2003

A thesis presented in partial fulfillment
of the requirements for the

Master of Science Degree in Electrical Engineering
Department of Electrical and Computer Engineering
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
August 2009

ABSTRACT

Optimal Design of Discrete-Time $\Delta\Sigma$ Modulators

by

Matthew Edward Jackson

Dr. Peter A. Stubberud, Examination Committee Chair
Professor of Electrical Engineering
University of Nevada, Las Vegas

In this thesis, optimal signal transfer functions (STFs) and noise transfer functions (NTFs) for discrete time delta sigma ($\Delta\Sigma$) modulators are determined. For a given oversampling rate (OSR), these STFs and NTFs are optimized with respect to a weighted combination of the $\Delta\Sigma$ modulator's signal-to-noise ratio (SNR) and dynamic range (DR). This optimization problem is solved using a novel hybrid orthogonal genetic (HOG) algorithm that uses customized genetic operators to improve algorithm performance and accuracy when applied to multimodal, non-differentiable performance surfaces. To generate optimal system functions, the HOG algorithm is implemented as a constrained global optimizer to minimize cost functions which represent approximations of the system functions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Peter Stubberud, for the countless hours he invested in this work. Moreover, I would like to thank his wife, Laura, for her patience and wisdom which always prevailed when cooler heads did not. Without her compassion this effort would have fallen well short of its mark. Secondly, I would like to thank my wife, Alicia, and all of my wonderful daughters for being supportive and understanding despite all the long weekends, late nights, and missed soccer games.

From a young age my family instilled in me a deep value for education built upon respect for the educated and an appreciation for the hard work and perseverance required to attain it. Thus, this work is dedicated to the educated men who inspired me: my grandfathers, Dr. Clyde T. Stoner and Dr. Raymond Van Pelt, and my father, Dr. Fredrick L. Jackson.

CHAPTER 1

INTRODUCTION

Mixed-signal systems are systems that possess both analog and digital subsystems. Such systems are prevalent in test and measurement platforms, data acquisition systems, and communications devices. Thus, these mixed-signal systems are often central to hardware applications ranging from common consumer products such as cellular telephones to highly specialized real-time data collection systems used in mission critical applications such as space flight.

In mixed-signal systems, the conversion from analog to digital is performed by an analog-to-digital converter (ADC). Conversely, the conversion from digital to analog is performed by a digital-to-analog converter (DAC). These devices are mixed-signal devices that allow for the ebb and flow of information between the analog world and digital or discrete-time systems which are now prevalent throughout electrical applications. Because the performance of digital systems can usually be improved by simple hardware or software changes, the performance of a mixed-signal system is often limited by the performance of its data converters. As a result, the performance of many mixed-signal systems can be improved by improving the system's data converter performance.

Many different ADC architectures exist and each architecture has its own benefits and limitations. $\Delta\Sigma$ modulators are an ADC architecture that uses relatively simple analog circuitry including a low order quantizer and a feedback loop to sample analog signals with high signal to noise ratios (SNRs) and large dynamic ranges (DRs).

Because of the simplicity of the architecture, $\Delta\Sigma$ modulators lend themselves to being implemented in CMOS process technologies which offer mixed-signal electronics, low-power performance and high levels of integration [20].

$\Delta\Sigma$ modulators achieve high SNRs and large DRs by using a feedback loop filter to attenuate the quantizer's noise in the frequency bands of interest while passing the input signal to the output. The transfer function describing the loop filter that attenuates the quantizer's noise is referred to as the $\Delta\Sigma$'s noise transfer function (NTF). Similarly, the transfer function describing the loop filter that passes the input signal to the output is called the signal transfer function (STF). For lowpass $\Delta\Sigma$ modulators, the NTF is designed as a high-pass filter so that the noise energy is attenuated within the low-frequency signal band. Conversely, the STF is designed as a lowpass filter so that the input signals within the low-frequency signal band are not attenuated. The STF can also act as an anti-aliasing filter. Thus, the output of a $\Delta\Sigma$ modulator can be modeled as the sum of an input signal filtered by a STF and a noise source filtered by a NTF.

In this thesis, optimal signal transfer functions (STFs) and noise transfer functions (NTFs) for $\Delta\Sigma$ modulators are determined using a novel hybrid orthogonal genetic (HOG) algorithm. For a given oversampling rate (OSR), which is loosely defined as the ratio of the $\Delta\Sigma$'s sampling frequency to the input signal's Nyquist frequency, the $\Delta\Sigma$'s STF and NTF are optimized with respect to a weighted combination of the $\Delta\Sigma$ modulator's signal-to-noise ratio (SNR) and dynamic range (DR).

CHAPTER 2

ANALOG-TO-DIGITAL CONVERTERS

Analog-to-digital converters (ADCs) are systems which convert continuous-time, continuous amplitude, or analog, signals into discrete-time, discrete amplitude, or digital signals. Typically, an ADC converts an analog signal, $x_a(t)$, defined over a continuous finite interval, R, into a digital signal, $x(n)$, defined over a discrete number, L, of values which span the interval R. The number, L, of discrete values that an ADC can produce is referred to as the ADC's resolution. Because most ADCs interface with binary electronic systems, an ADC's resolution is typically a power of 2; that is, $L = 2^B$ where B is an integer representing the binary bit-width of the ADC interface. Thus, ADC resolution is often expressed in terms of the number, B, of bits and not the number, L, of available quantization levels.

For linear ADCs, each of the 2^B quantization levels are equidistant over the signal span R. For such ADCs, the quantization step size, Δ, or distance between adjacent quantization levels is expressed as $\Delta = R/2^B$ where R corresponds to the input signal span as defined above. For example, consider an analog input, $x_a(t)$, which has a signal span from -1 to 1; that is, consider an analog input, $x_a(t)$, where

$$-1 \leq x_a(t) \leq 1$$

which has a signal span, R, where $R = 2$. For a 2-bit system, the signal span, R, is

divided into 2^2 equidistant levels where the quantization step size, Δ, is

$$\Delta = \frac{R}{2^B} = \frac{2}{2^2} = 0.5.$$

This example is illustrated in Figure 2.1 for $x_a(t) = \cos(2\pi 1000t)$, $x_a(nT_s) = \cos(\pi n/32)$ for $T_s = 1/64\pi 1000$, and $x(n) = \mathcal{Q}\left[x_a(nT_s)\right]$ where T_s is the sampling period in time per sample and $\mathcal{Q}[\cdot]$ is the transformation that quantizes the continuous amplitude, discrete-time signal, $x_a(nT_s)$, by rounding the amplitude to the nearest quantization level.

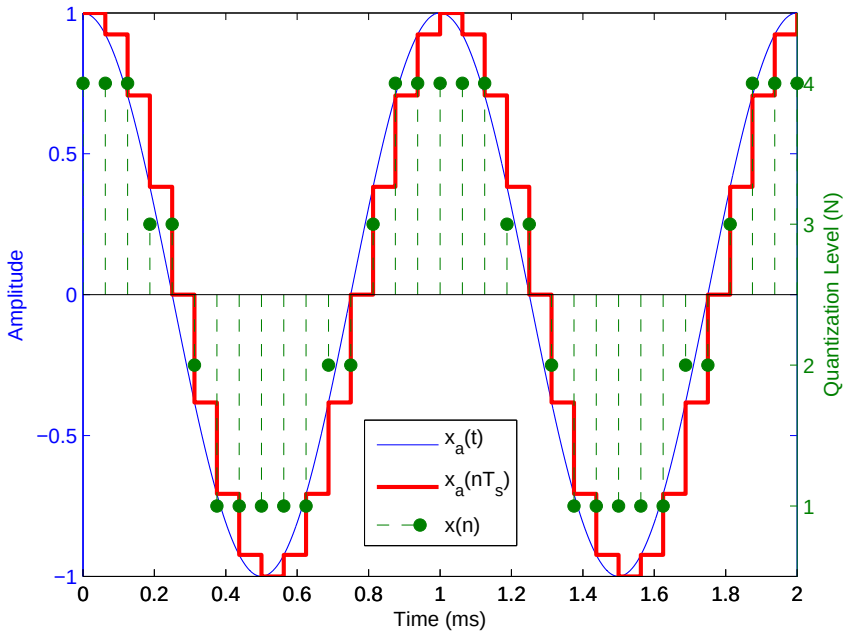

Figure 2.1: 2-Bit Quantization

The difference, $x_a(nT_s) - x(n)$, is commonly referred to as the quantization error

and is often characterized as quantization noise in the ADC's output. As shown in Figure 2.1, the digital signal, $x(n)$, is different than both the analog input signal, $x_a(t)$, and the discrete-time, continuous amplitude signal, $x_a(nT_s)$. As the number, B, of bits increases, the quantization error, or quantization noise, typically decreases. Thus, an ADC's quantization noise is typically a function of the number, B, of quantization bits. In practice, however, ADC levels are not equally spaced; that is, Δ varies slightly over the interval R. This non-uniformity in level spacing creates distortion which is also often characterized as noise in the ADC output. Non-uniform level spacing along with other non-idealities such as improper input signal conditioning, system thermal noise, and sample clock jitter, limit an ADC's effective resolution to less than the ideal. As a result, an ADC's effective resolution is often determined from the ADC's output signal-to-noise ratio (SNR) or dynamic range (DR) which are performance metrics that are independent of the ADC's native architecture.

SNR is defined as the ratio of output signal power to output noise power which contains power from quantization error and other ADC non-idealities. Dynamic range is defined as the ratio of the maximum to the minimum detectable signal levels. DR differs from SNR when the noise floor is not perfectly flat; i.e., noise power in localized frequency regions is greater than the average noise power. The effects of DR further limit the practical performance of ADCs. As such, an ADC's effective resolution is often calculated from the ADC's SNR and DR. The effective resolution is referred to as the ADC's effective number of bits (ENOB) where ENOB is defined as the achievable ADC resolution when its non-ideal ADC characteristics are considered.

$\Delta\Sigma$ modulators are ADCs which achieve high SNRs and large DRs by using a feedback loop filter to attenuate the quantization noise in the frequency band of interest while passing the input signal to the output. The transfer function describing the loop filter that attenuates the quantization noise is referred to as the noise transfer

function (NTF). Similarly, the transfer function describing the loop filter that passes the input signal to the output is referred to as the signal transfer function (STF). For example, the NTF for lowpass $\Delta\Sigma$ modulators is designed as a highpass filter so that the noise energy is attenuated within the low-frequency signal band. The STF for lowpass $\Delta\Sigma$ modulators is designed as a lowpass filter so that the input signals within the low-frequency signal band are not attenuated. In addition, the lowpass characteristics of the STF can also act as an anti-aliasing filter. As such, the output of a lowpass $\Delta\Sigma$ modulator can be modeled as the sum of a noise source that is highpass filtered by the NTF and an input signal that is lowpass filtered by the STF.

$\Delta\Sigma$ modulator NTFs and STFs are typically designed and implemented as either discrete or analog linear recursive filters. As such, a $\Delta\Sigma$ modulator's NTF and STF can be designed using traditional filters such as Chebyshev or Butterworth filters. However, these methods are not easily adaptable to the atypical frequency response characteristics commonly required by many $\Delta\Sigma$ modulators. Historically, numerical optimization methods have been applied to the optimal design of linear recursive filters with good success [30] [5] [8]. As such, design techniques which rely heavily on numerical optimization methods can be used to optimize $\Delta\Sigma$ modulator system design. Some numerical filter design programs include other electronic design automation (EDA) tools which automate much of the design process. For example, the Delta Sigma Toolbox for MATLAB® provides an integrated set of discrete-time $\Delta\Sigma$ modulator design, simulation, and synthesis utilities [35]. However, this thesis will show that the design method in the Delta Sigma toolbox offers only marginal improvement over traditional Chebyshev or Butterworth polynomial based filter design methods.

In this thesis, a global numerical optimization algorithm, called the hybrid orthogonal genetic (HOG) algorithm, is developed which can determine the optimal

design of both discrete and analog linear recursive filters. In this thesis, the HOG algorithm is used to optimize the performance of $\Delta\Sigma$ modulator's NTFs and STFs by maximizing the in-band SNR and DR.

2.1 Operational Theory

Figure 2.2 shows a basic mathematical model of an ADC. As illustrated, ADCs can be modeled as a sample-and-hold (S/H) circuit in series with a quantizer and binary encoder. The sample-and-hold circuit samples the analog input signal at discrete times where the sample-and-hold process is defined as the process of capturing the input signal's amplitude at the sample times, nT_s, where $n \in I$, and holding it over the sampling period, T_s. The quantizer then approximates the sampled signal's amplitude, $x_a(nT_s)$, by converting it to one of the ADC's L quantization levels which are uniformly spaced by the distance Δ for a linear ADC. Finally, the binary encoder converts the digital signal, $x(n)$, into a B-bit binary code word.

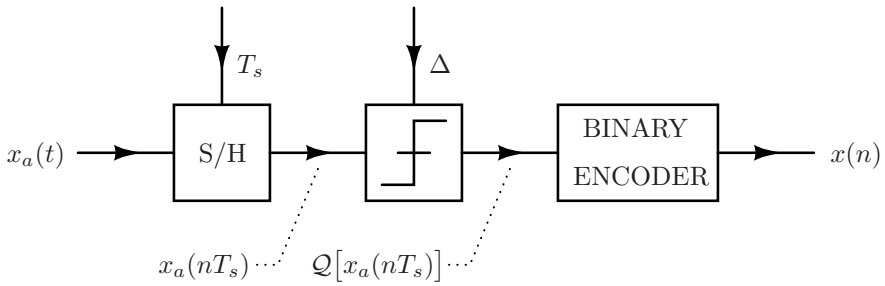

Figure 2.2: Basic ADC Block Diagram

2.1.1 Sampling

The Shannon-Nyquist sampling theorem states that an analog signal must be sampled at a rate that is at least twice its bandwidth for the analog signal to be reconstructed from its samples. Specifically, the Shannon-Nyquist sampling theorem states that if an analog signal, $x_a(t)$, is strictly bandlimited such that its Fourier transform, $X_a(f)$, has the property that

$$X_a(f) = 0 \quad |f| > f_0,$$

where f is the instantaneous frequency in cycles per second or Hertz (Hz) and f_0 is a fixed frequency, then $x_a(t)$ can be recovered from its samples, $x_a(nT_s)$, if

$$T_s \leq \frac{1}{2f_0} \tag{2.1}$$

where T_s is the sampling period in time per sample. The frequency, f_0, is referred to as the Nyquist frequency and the frequency, $2f_0$, is referred to as the Nyquist rate. Converters which sample the input at or near $2f_0$ samples per second are referred to as Nyquist rate converters. Common Nyquist rate converter architectures include flash, dual-slope, successive approximation (SAR), and pipelined converters [7].

2.1.2 Quantization

In this thesis, the quantization transformation, denoted $\mathcal{Q}[\cdot]$, is a nonlinear transformation which approximates a discrete-time, continuous amplitude signal by a digital signal that has a finite number of fixed quantization levels. To illustrate, consider an analog signal, $x_a(t)$, and its corresponding quantized signal $x(n)$ where $x(n) = \mathcal{Q}[x_a(nT_s)]$. If $x(n)$ is a B-bit quantized signal, then the number of quanti-

zation levels, L, can be expressed as

$$L = 2^B. \tag{2.2}$$

If the quantized signal, $x(n)$, is bounded such that

$$\Big|x(n)\Big| \leq X_m \tag{2.3}$$

where X_m represents the quantizer's maximum input amplitude without saturation, the quantization interval or step size, Δ, defined as the distance between any two adjacent quantization levels, can then be expressed as

$$\Delta = \frac{X_m}{2^{B-1}}. \tag{2.4}$$

The difference between the discrete-time, continuous amplitude signal, $x_a(nT_s)$, and the digital signal, $x(n)$, is referred to as the quantization error, $e(n)$. As such, the quantizer's output, $x(n)$, can be expressed as the sum of the sampled analog signal, $x_a(nT_s)$, and the quantization error, $e(n)$; that is,

$$x(n) = \mathcal{Q}\Big[x_a(nT_s)\Big] = x_a(nT_s) + e(n) \tag{2.5}$$

where $\mathcal{Q}[\cdot]$ represents the nonlinear quantization transformation. If a rounding quantizer is implemented and it is assumed that:

- $e(n)$ is a stationary random process
- $e(n)$ is uncorrelated with the quantizer's input
- $e(n)$ is a white noise process; i.e. it's samples are uncorrelated

- $e(n)$ has a probability density that is uniform over the quantization error range $\left[-\Delta/2, \Delta/2\right]$

then the quantizer shown in Figure 2.3(a) can be modeled by the linear system shown in Figure 2.3(b) [13] [26]. Using this linear quantizer model greatly reduces the complexity associated with ADC analysis at the expense of modeling accuracy. However, it has been shown that for rapidly varying input signals and small quantization intervals or Δ's, the results obtained from this linear noise model are sufficient for most calculations [26] [14].

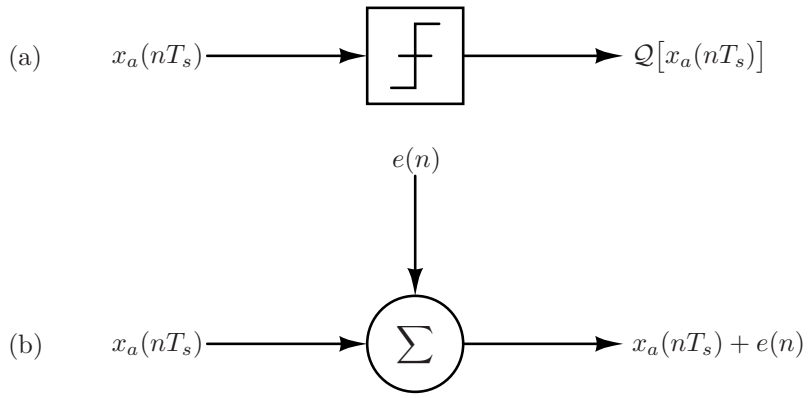

Figure 2.3: Linear Quantizer Model
(a) Nonlinear Quantizer (b) Linear Quantizer Model

2.2 Performance Metrics

An ADC's performance is often described in terms of SNR and DR where both metrics compare the relative output signal power to the output noise power. Because deterministic signals and stochastic noise are modeled differently, their respective powers are calculated using different techniques. For deterministic signals, power is

calculated analytically from the available signal information. For stochastic signals, power is calculated in terms of the statistical characteristics which define the signal.

2.2.1 Signal and Noise Power

The mean or expectation of a continuous random process, $x(t)$, is defined as

$$E\Big[x(t)\Big] = \int_{-\infty}^{\infty} x(t) p_x\Big(x\,(t)\Big) dx(t) \tag{2.6}$$

where $p_x\Big(x\,(t)\Big)$ is the probability density function of $x(t)$. Similarly, the mean or expectation of a discrete random process, $x(n)$, is defined as

$$E\Big[x(n)\Big] = \sum_{k=1}^{N} x_k(n) P_{x_k(n)}\Big(x_k(n)\Big) \tag{2.7}$$

where $\{x_k(n) : k = 1, \cdots, N\}$ is the range of $x(n)$, and $P_{x_k(n)}\Big(x_k(n)\Big)$ is the probability mass function of $x(n)$. Equations (2.6) and (2.7) are referred to as the ensemble or state averages of a random process.

The variance, σ_x^2, of a random process, x, is defined as

$$\sigma_x^2 = E\Big[\Big(x - E[x]\Big)^2\Big] \tag{2.8}$$

which can be expressed as

$$\begin{aligned} \sigma_x^2 &= E\Big[\Big(x - E[x]\Big)^2\Big] \\ &= E\Big[x^2 - 2xE[x] + E^2[x]\Big] \\ &= E\Big[x^2\Big] - E^2[x]. \end{aligned} \tag{2.9}$$

For a zero mean random process, (2.9) reduces to

$$\sigma_x^2 = E\left[x^2\right]. \tag{2.10}$$

Calculating the expectation or variance of a random process requires its probability density or probability mass function. In practice, the probability densities or probability mass functions for the random variables of a random process are not necessarily well defined. However, if a random process is said to be ergodic, then the random process' ensemble average is equivalent to its time average, and time averages can be used to estimate the random process' mean and variance [27] [22].

For a continuous-time random process, $x(t)$, the time average, $\mu_{x(t)}$, of $x(t)$ is defined as

$$\mu_{x(t)} = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(\tau) d\tau. \tag{2.11}$$

Similarly, for a discrete-time random process, $x(n)$, the time average, $\mu_{x(n)}$, of $x(n)$ is defined as

$$\mu_{x(n)} = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{k=-N}^{N} x(k). \tag{2.12}$$

Thus, if a zero mean, continuous-time random process, $x(t)$, is ergodic, the variance, $\sigma_{x(t)}^2$, of $x(t)$ can be represented in terms of its time average, $\mu_{x^2(t)}$, as

$$\sigma_{x(t)}^2 = E\left[x^2(t)\right] = \mu_{x^2(t)} = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x^2(\tau) d\tau. \tag{2.13}$$

Similarly, if a zero mean, discrete-time random process, $x(n)$, is ergodic, the variance, $\sigma_{x(n)}^2$, of $x(n)$ can be represented in terms of its time average, $\mu_{x^2(t)}$, as

$$\sigma_{x(n)}^2 = E\left[x^2(n)\right] = \mu_{x^2(t)} = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{k=-N}^{N} x^2(k). \tag{2.14}$$

The power, $P_{x(n)}$, for a deterministic discrete-time signal, $x(n)$, can be calculated as

$$P_{x(n)} = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{k=-\infty}^{\infty} x^2(k) \tag{2.15}$$

which is identical to (2.14). Therefore, for a zero mean random signal, $x(n)$, $P_{x(n)} = \sigma^2_{x(n)}$. In practice, SNR and DR are estimated using a finite length sequence. Thus, for a signal of length $2N+1$, the average signal power, $P_{x(n)[-N,N]}$, is given as

$$P_{x(n)[-N,N]} = \frac{1}{2N+1} \sum_{k=-N}^{N} x^2(k). \tag{2.16}$$

Because (2.14) and (2.16) are identical for finite length signals,

$$P_{x(n)[-N,N]} = P_{x(n)} = \sigma^2_{x(n)}$$

for zero mean signals of length $2N+1$.

Recall that a quantizer's output, $x(n)$, which is also the ADC's output, can be modeled linearly as the sum of the continuous amplitude, discrete-time signal, $x_a(nT_s)$, and the quantization error, $e(n)$, as given in (2.5). Because $x_a(nT_s)$ and $e(n)$ are assumed to be uncorrelated and $e(n)$ is modeled as a zero mean random process, the output power, $P_{x(n)}$, of $x(n)$ can be expressed as

$$\begin{aligned}
P_{x(n)} &= E\left[x^2(n)\right] \\
&= E\Big[\Big(x_a(nT_s) + e(n)\Big)^2\Big] \\
&= E\Big[x_a^2(nT_s)\Big] + 2E\Big[x_a(nT_s)e(n)\Big] + E\Big[e^2(n)\Big] \\
&= E\Big[x_a^2(nT_s)\Big] + 2E\Big[x_a(nT_s)\Big]E\Big[e(n)\Big] + E\Big[e^2(n)\Big] \\
&= E\left[x_a^2(nT_s)\right] + E\left[e^2(n)\right] \\
&= P_{x_a(nT_s)} + P_{e(n)}
\end{aligned} \tag{2.17}$$

where $P_{x_a(nT_s)}$ and $P_{e(n)}$ correspond to the output signal power and output noise power, respectively. Because the quantizer's input, $x_a(nT_s)$, is typically modeled deterministically, its average signal power, $P_{x_a(nT_s)}$, can be given as

$$P_{x_a(nT_s)} = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x_a^2(nT_s) \tag{2.18}$$

or for finite length signals of length $2N+1$ or periodic signals that have a period of $2N+1$ as

$$P_{x_a(nT_s)} = \frac{1}{2N+1} \sum_{n=-N}^{N} x_a^2(nT_s). \tag{2.19}$$

Because the quantization error, $e(n)$, is modeled as a random process, its average power, $P_{e(n)}$, is given as

$$P_{e(n)} = E\left[e^2(n)\right] = \sigma_{e(n)}^2 = \int_{-\Delta/2}^{\Delta/2} e^2(n) p_{e(n)}\Big(e(n)\Big) de(n) \tag{2.20}$$

2.2.2 Signal-to-Noise Ratio (SNR)

Recall that SNR is defined as the ratio of output signal power to output noise power. SNR_{dB}, which is the SNR expressed in decibels (dB), is given as

$$\text{SNR}_{\text{dB}} = 10\log\left(\text{SNR}\right) = 10\log\left(\frac{P_s}{P_e}\right) \tag{2.21}$$

where P_s and P_e correspond to the output signal power and output noise power, respectively. Assuming that the quantizer is modeled as an additive random white-noise source, $e(n)$, the theoretical SNR for a sampled deterministic input signal,

$x_a(nT_s)$, is given by substituting (2.19) and (2.20) into (2.21) which implies that

$$\mathrm{SNR}_{\mathrm{dB}} = 10\log\left(\frac{P_{x_a(nT_s)}}{P_{e(n)}}\right) = 10\log\left(\frac{\lim\limits_{N\to\infty}\dfrac{1}{2N+1}\displaystyle\sum_{n=-N}^{N} x_a^2(nT_s)}{\displaystyle\int_{-\Delta/2}^{\Delta/2} e^2(n)p_{e(n)}\Big(e(n)\Big)de(n)}\right). \tag{2.22}$$

Sinusoids are often used to stimulate ADCs under analysis so the signal energy is located at a unique frequency. As such, the ADC's output, $x(n)$, can be written as

$$x(n) = x_a(nT_s) + e(n) = A\sin\left(\omega_0 nT_s\right) + e(n).$$

Thus, the average power of the sinusoidal output signal, $P_{x_a(nT_s)}$, is given as

$$\begin{aligned} P_{x_a(nT_s)} &= \lim_{N\to\infty}\frac{1}{2N+1}\sum_{n=-N}^{N} A^2\sin^2(\omega_0 nT_s) \\ &= \lim_{N\to\infty}\left\{\frac{A^2}{4N+2}\sum_{n=-N}^{N}\Big(1-\cos(2\omega_0 nT_s)\Big)\right\} \\ &= \frac{A^2}{2} \end{aligned} \tag{2.23}$$

where A is the signal amplitude. If the ADC input is a full-scale sinusoid with amplitude, A, then $\Delta = 2A/2^B$ which implies that

$$A = \frac{2^B\Delta}{2}, \tag{2.24}$$

where Δ is the quantization step size, and B corresponds to the ADC's resolution in bits. Substituting (2.24) into (2.23), the average output signal power can be expressed as

$$P_{x_a(nT_s)} = \frac{A^2}{2} = \frac{\left(\dfrac{2^B\Delta}{2}\right)^2}{2} = \frac{\Delta^2 2^{2B}}{8}. \tag{2.25}$$

Recall that for a rounding quantizer, the quantization noise is modeled as a white noise process with uniform distribution over $\Big[-\Delta/2, \Delta/2\Big]$. As such, the probability density function, $p_{e(n)}\Big(e(n)\Big)$, is $1/\Delta$ for $-\Delta/2 \leq e(n) \leq \Delta/2$ which implies that

$$P_{e(n)} = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} e^2(n)de(n) = \frac{\Delta^2}{12}. \tag{2.26}$$

By substituting (2.25) and (2.26) into (2.21), the theoretical SNR for a B-bit ADC when stimulated by a full-scale sinusoid can be expressed as

$$\mathrm{SNR}_{\mathrm{dB}} = 10\log\frac{\left(\dfrac{\Delta^2 2^{2B}}{8}\right)}{\left(\dfrac{\Delta^2}{12}\right)} = 10\log\Big(2^{2B}\Big(3/2\Big)\Big) \tag{2.27}$$

or equivalently as

$$\mathrm{SNR}_{\mathrm{dB}} = 6.02B + 1.76. \tag{2.28}$$

2.2.3 Dynamic Range (DR)

Recall that dynamic range is defined as the ratio of the maximum to the minimum detectable signal levels. If the noise spectrum is constant for all frequencies, DR is equivalent to SNR as given in (2.28); that is, for a flat, or white, noise floor, an ADC's SNR and DR are identical. However, in practice, the noise floor is not always flat and in such cases, the peak of the noise floor limits the usable dynamic range of the ADC to less than the SNR. As a result, the peak of the noise floor limits the effective resolution of the ADC to less than the value predicted by (2.28).

To illustrate, consider an ADC that has the output spectrum shown in Figure 2.4. Because the noise spectrum is not constant for all frequencies, the peak of the noise floor is larger than the average noise floor. The difference, $\Delta\Gamma$, between the peak of the noise floor and the average noise floor results in a loss of effective resolution

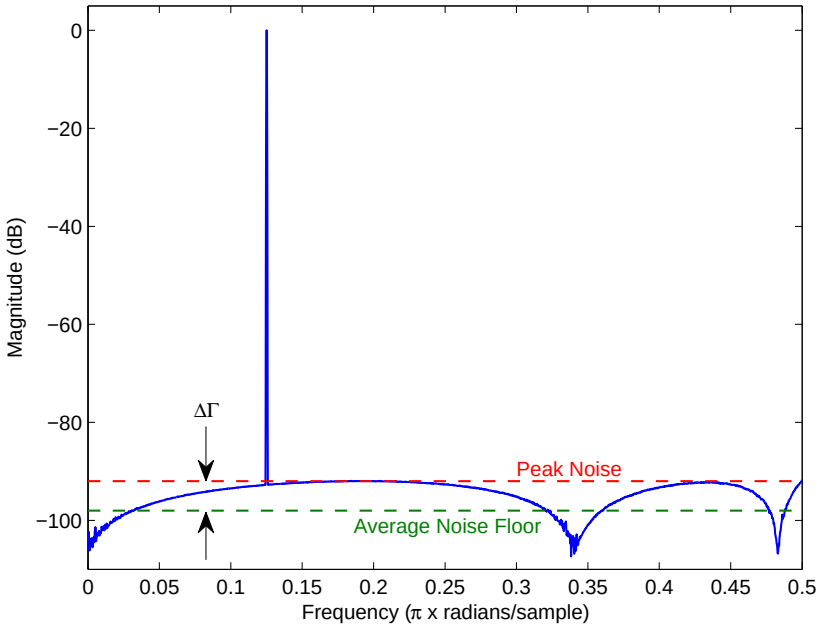

Figure 2.4: 16-bit ADC Output Spectrum

across much of the spectrum.

2.3 Architectures

Many different types of ADC architectures exist. Selecting the appropriate architecture for a given application is often a trade-off between size, power consumption, operational bandwidth, conversion latency, resolution, and sampling rate. For example, a Nyquist rate converter's resolution is often limited by its inherent technology. However, system design methods which incorporate signal processing techniques such as oversampling can increase the effective resolution of a Nyquist rate converter by reducing its operational bandwidth. Pipelined ADCs use subranging techniques to parse the sample conversion over several iterations thereby improving conversion accuracy. Typically, such architectures offer high resolutions over reasonable operational

bandwidths often without the need for additional post-processing. However, the iterative conversion process requires additional time thereby increasing conversion latency. For moderate bandwidth applications, specialized architectures such as $\Delta\Sigma$ modulators are available. Such architectures use feedback and oversampling to achieve high resolution from relatively simple, low power hardware.

2.3.1 Nyquist Rate Converters

Recall that Nyquist rate converters are ADCs which sample the input at or near its Nyquist rate, $2f_0$, as defined in (2.1). As such, the sampling frequency, f_s, must be must be at least twice the input signal's Nyquist bandwidth, f_0. To minimize out of band signal energy from aliasing into the operational bandwidth of the Nyquist rate converter, the input signal is typically bandlimited to $f_s/2$ by an an anti-aliasing filter prior to being sampled. However, due to practical limitations of this anti-aliasing filter, Nyquist converters often sample the input signal at a frequency that is slightly higher than the Nyquist rate.

Nyquist converter bandwidths are typically limited by the electrical properties of the fabrication process in which they are implemented. With current fabrication processes capable of supporting signal bandwidths in the GHz range, Nyquist converters are capable of processing bandlimited signals with bandwidths well in excess of 500 MHz [25]. As a result, Nyquist converters offer the widest range of usable bandwidth when compared to other ADC architectures. However, the effective resolution of Nyquist converters is typically limited by achievable device density and electronic component matching. As device geometries decrease, the inherent mismatch of components increases which limits the converter's achievable resolution.

To illustrate, consider the B-bit Flash ADC in Figure 2.5. Such implementations require $2^B - 1$ comparators and 2^B resistors. Because the total area required in an IC to implement a design is proportional to the overall device count and because

the device count of a B-bit Flash ADC increases exponentially with B, Flash ADCs with practical dimensions are limited to a resolution of 8-bits for current process technologies [17]. Also, Flash ADCs require components to be matched perfectly to maintain uniform quantization step sizes, Δ. Because nominal components do not match well over large device geometries, the effective resolution of Flash based ADCs are typically less than the ideal.

2.3.2 Oversampling Converters

Oversampling ADCs are ADCs which increase their effective resolution by sampling their input signals at much higher rates than its Nyquist rate and then bandlimiting the quantization noise to the Nyquist bandwidth of the input signal. The ratio of the sampling frequency, f_s, to the input signal's Nyquist rate, $2f_0$, is referred to as the oversampling-rate (OSR) and in this thesis is denoted as M; that is,

$$M = \frac{f_s}{2f_0}. \tag{2.29}$$

To illustrate, consider a Nyquist ADC with a sampling frequency, f_s, and an input signal with a Nyquist bandwidth, f_0. As illustrated in Figure 2.6(a), if $f_s = 2f_0$, then the quantization noise is uniformly distributed over the operational bandwidth, f_{NY}, where $f_{\text{NY}} \in [-f_0, f_0]$. Alternatively, consider an oversampling ADC with a sampling frequency, f_s, such that $f_s = M2f_0$ where M is the OSR. For such an ADC, the quantization noise is uniformly distributed over the operational bandwidth, f_{OS}, where $f_{\text{OS}} \in [-f_s/2, f_s/2]$. As illustrated in Figure 2.6(b), if the ADC's output is filtered so that it is bandlimited to the input signal's Nyquist bandwidth, f_0, then the quantization noise power distributed over the remaining frequencies, $f_0 \leq |f| \leq Mf_0$, is effectively removed from the output. Thus, the average quantization noise power is decreased by a factor of M and the SNR is increased by a factor of M thereby

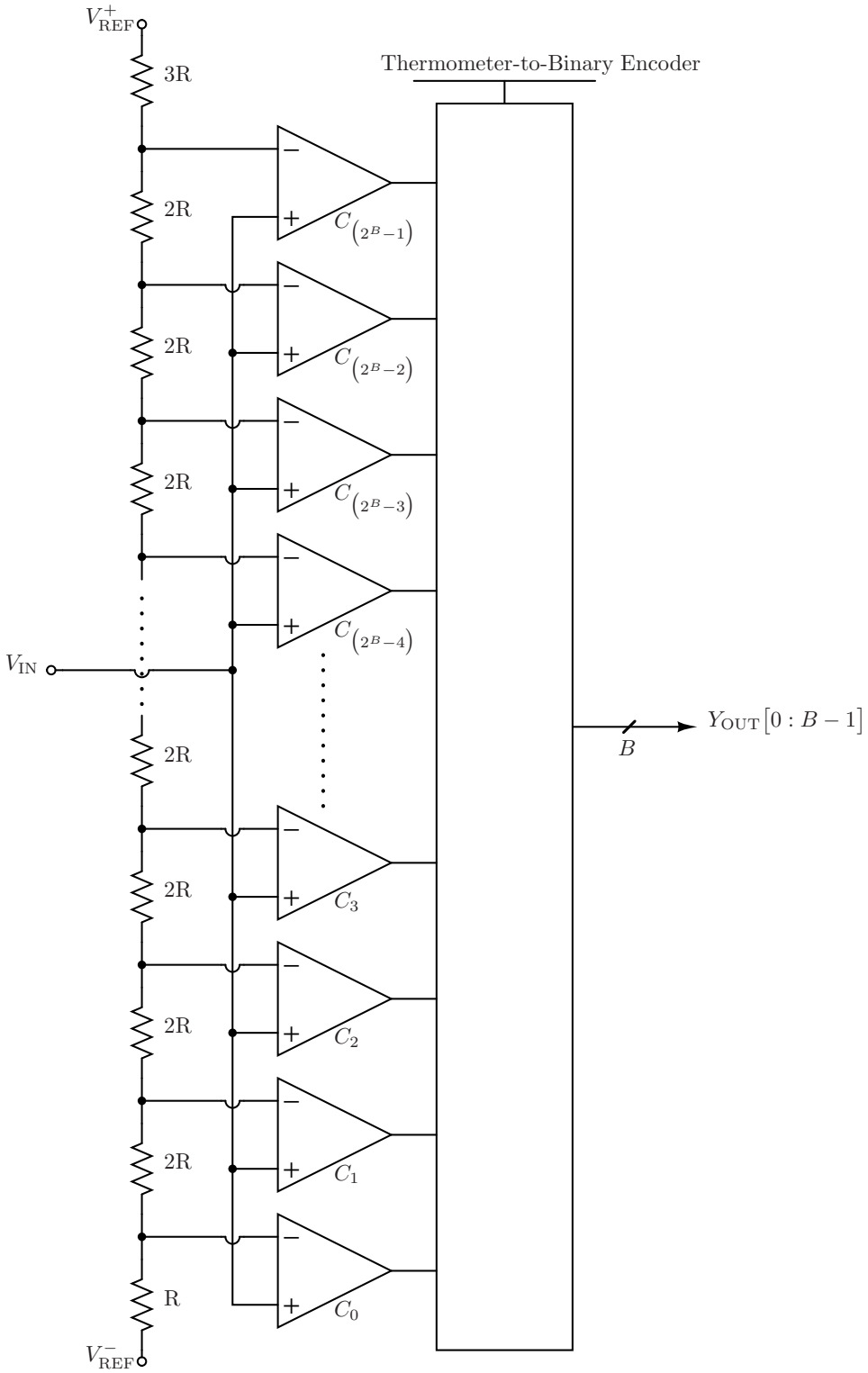

Figure 2.5: Flash ADC System Block Diagram

increasing the ADC's effective resolution. From observation of Figure 2.6, the in-band quantization noise for the oversampling converter is significantly less than the Nyquist converter.

To further illustrate, consider a B-bit oversampling ADC that has an OSR, M, with a quantization noise power, P_e, that is uniformly distributed over its operational bandwidth $[-f_s/2, f_s/2]$, where f_s denotes the sampling frequency. If the ADC's output is filtered so that the quantization noise power is bandlimited to the input signal's Nyquist bandwidth, $[-f_0, f_0]$, the filtered output quantization noise power, $P_{e,\mathrm{OS}}$, can be expressed as

$$P_{e,\mathrm{OS}} = \frac{1}{f_s}\int_{-f_0}^{f_0} P_e df = P_e \frac{2f_0}{f_s} = \frac{P_e}{M}. \tag{2.30}$$

Thus, the average quantization noise power, $P_{e,\mathrm{OS}}$, for an oversampling ADC with an OSR, M, can be calculated by substituting (2.26) into (2.30) which results in

$$P_{e,\mathrm{OS}} = \frac{\Delta^2}{12}\left(\frac{1}{M}\right) \tag{2.31}$$

where Δ corresponds to the quantization step size. Substituting (2.31) into (2.21) and solving for the theoretical SNR for an oversampled Nyquist rate converter which is stimulated by a full-scale sinusoid as defined by (2.25) yields

$$\begin{aligned} \mathrm{SNR}_{\mathrm{dB,OS}} &= 10\log\frac{\left(\frac{\Delta^2 2^{2B}}{8}\right)}{\left(\frac{\Delta^2}{12M}\right)} \\ &= 10\log\Big(2^{2B}\big(3/2\big)M\Big) \\ &= 6.02B + 1.76 + 10\log(M) \end{aligned} \tag{2.32}$$

where B is the number of quantization bits and M is the OSR. Because the maximum OSR is a function of the ADC's maximum sampling frequency, M is typically selected between 8 and 256. As such, oversampling converters can typically achieve a 9 to 24 dB increase in SNR which is equivalent to an increase of 1 to 3 bits in effective resolution.

2.3.3 $\Delta\Sigma$ Modulators

$\Delta\Sigma$ modulators are an ADC architecture that uses oversampling and a feedback loop to achieve high signal to noise ratios (SNRs) and large dynamic ranges (DRs). Because of the simplicity of the architecture, $\Delta\Sigma$ modulators can be implemented using relatively simple analog circuitry in standard CMOS processes which offer low-power performance and high levels of integration for mixed-signal electronics [20].

To achieve a large DR and high SNR, $\Delta\Sigma$ modulators are designed so that the quantization noise's feedback loop filter, or noise transfer function (NTF), attenuates the quantization noise within the frequency band of interest. Additionally, as with other oversampling ADCs, the $\Delta\Sigma$ modulator's output is bandlimited to the input signal's Nyquist bandwidth, f_0. A comparison of the output spectra for a Nyquist rate converter, an oversampling converter, and a $\Delta\Sigma$ modulator is shown in Figure 2.6 (adapted from [19]) which illustrates the relative amount of in-band noise power for each architecture. As illustrated in Figure 2.6(c), the amount of in-band noise power for $\Delta\Sigma$ modulators is largely determined by the shape of the NTF.

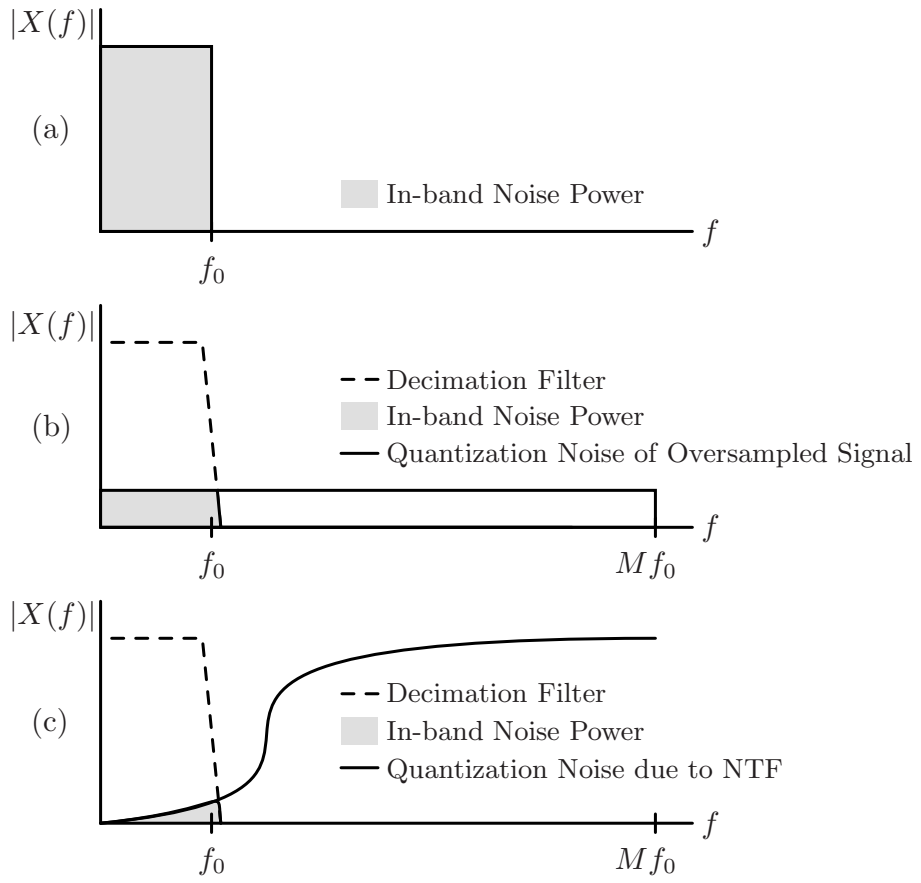

Figure 2.6: ADC Output Noise Spectrum Comparison

(a) Nyquist Rate Converter (b) Oversampling Converter (c) $\Delta\Sigma$ Modulator

2.3.3.1 Noise Shaping

Figure 2.7 illustrates a generic system structure of a discrete-time $\Delta\Sigma$ modulator. The input block, $F(z)$, is a discrete system that samples the analog input signal, $x_a(t)$, and processes the resulting discrete-time, continuous amplitude signal, $x_a(nT_s)$. The ADC block quantizes, or digitizes, $x_a(nT_s)$ to one of 2^B quantization levels, where B denotes the number of bits in the digital output, $x(n)$. The feedback DAC then converts the digital output signal, $x(n)$, into a discrete signal that is fedback through

$H(z)$ and into $G(z)$.

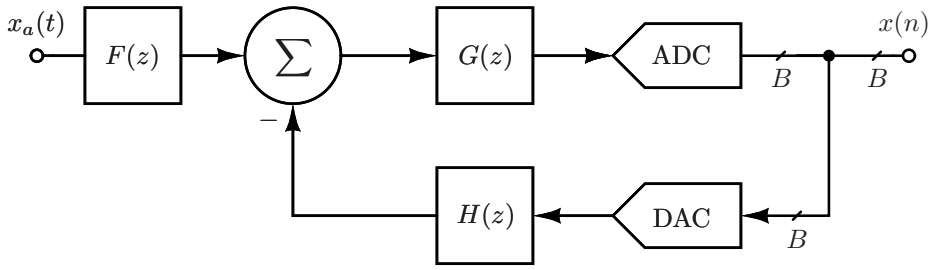

Figure 2.7: $\Delta\Sigma$ Modulator Block Diagram

Figure 2.8 illustrates Figure 2.7's generic $\Delta\Sigma$ modulator where the ADC's quantizer is modeled as an additive white noise source. For this thesis, only single-bit quantizers are considered. As such, the time required for data conversion, or latency through the ADC and the DAC, can be modeled as a unit delay in the feedback path.

Because the blocks, $F(z)$, $G(z)$, and $H(z)$ are typically implemented as linear time invariant (LTI) subsystems, the NTF and STF can be expressed as

$$\mathrm{STF}(z) = \frac{Y(z)}{X(z)} = \frac{F(z)G(z)}{1 + z^{-1}G(z)H(z)} \tag{2.33}$$

and

$$\mathrm{NTF}(z) = \frac{Y(z)}{E(z)} = \frac{1}{1 + z^{-1}G(z)H(z)}. \tag{2.34}$$

Figure 2.9 illustrates a STF and NTF for a lowpass $\Delta\Sigma$ modulator where the quantization noise is attenuated by a highpass NTF and the input signal is filtered by a lowpass STF. If the NTF and STF are modeled as LTI systems, the output,

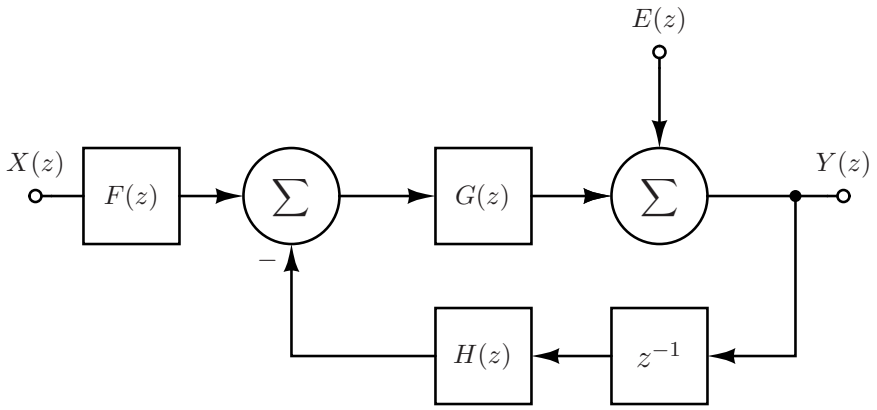

Figure 2.8: $\Delta\Sigma$ Modulator Linear Model Block Diagram

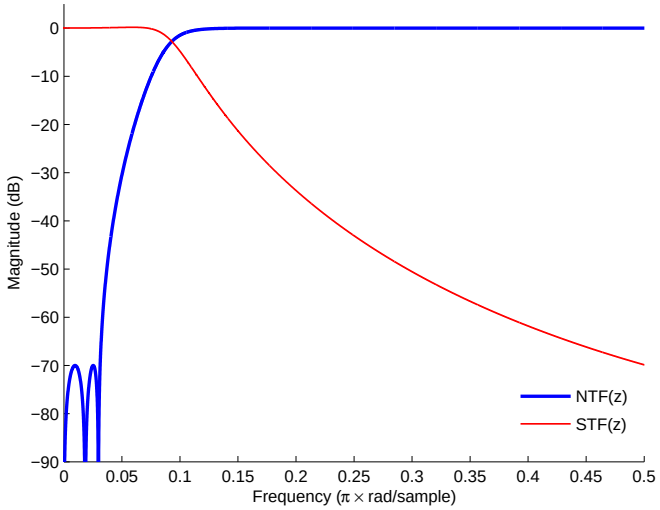

Figure 2.9: Examples of a STF and NTF Magnitude Responses for a Lowpass Discrete-Time $\Delta\Sigma$ Modulator

$Y(z)$, of a $\Delta\Sigma$ modulator can be expressed as

$$Y(z) = \mathrm{STF}(z)X(z) + \mathrm{NTF}(z)E(z) \tag{2.35}$$

where $X(z)$ and $E(z)$ correspond to the ‡-transforms of the input signal and quantization noise respectively.

2.3.4 $\Delta\Sigma$ Modulator Implementations: Linear Models

The $\Delta\Sigma$ modulator that has been mathematically modeled by the block diagram shown in Figure 2.7 can be implemented using many different structures. Common hardware implementations include cascade-of-resonators-feedback (CRFB), cascade-of-resonators-feedforward (CRFF), cascade-of-integrators-feedback (CIFB), and cascade-of-integrators-feedforward [35]. For this thesis, a CRFB implementation was implemented.

2.3.4.1 First Order System

$\Delta\Sigma$ modulator implementations which utilize analog hardware (e.g. integrators) to realize their loop filters are referred to as continuous-time $\Delta\Sigma$ modulators. For example, Figure 2.10 illustrates a 1st order, continuous-time $\Delta\Sigma$ modulator. Similarly,

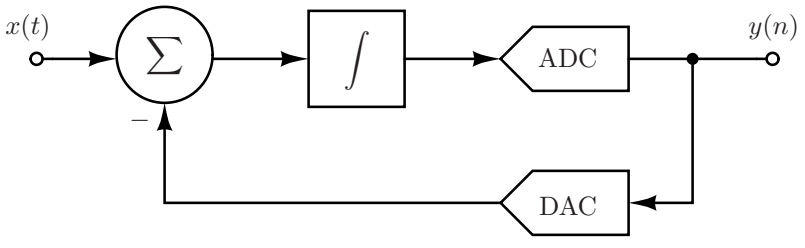

Figure 2.10: First-Order Continuous-Time $\Delta\Sigma$ Modulator

implementations which utilize discrete-time hardware (e.g. accumulators) to realize their loop filters are referred to as discrete-time $\Delta\Sigma$ modulators. As illustrated in Figures 2.10 and 2.11, discrete-time $\Delta\Sigma$ modulators typically use an accumulator in place of the integrators. Because discrete-time $\Delta\Sigma$ modulators use discrete-time

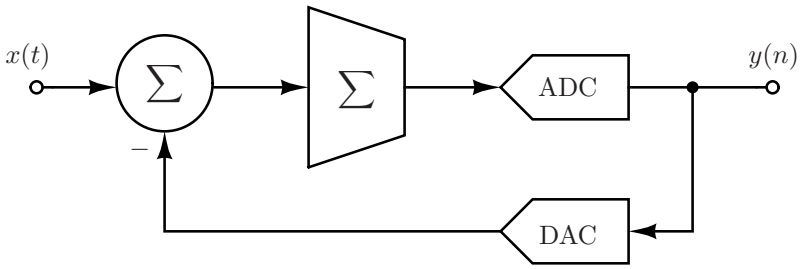

Figure 2.11: First-Order Discrete-Time $\Delta\Sigma$ Modulator

hardware to realize their loop filters, traditional discrete-time design and analysis techniques can be used [14] [6].

Figure 2.12 shows a 1st order, discrete-time $\Delta\Sigma$ modulator where the quantization noise is modeled as an additive white noise source. Recall that a $\Delta\Sigma$ modulator's output can be modeled as the sum of the quantization error and the input signal. For lowpass architectures, the quantization error is highpass filtered by the NTF and the input signal is lowpass filtered by the STF as described by (2.35). From observation of Figure 2.12, the $\Delta\Sigma$ modulator's output, $Y(z)$, is given as

$$Y(z) = E(z) + A(z) \tag{2.36}$$

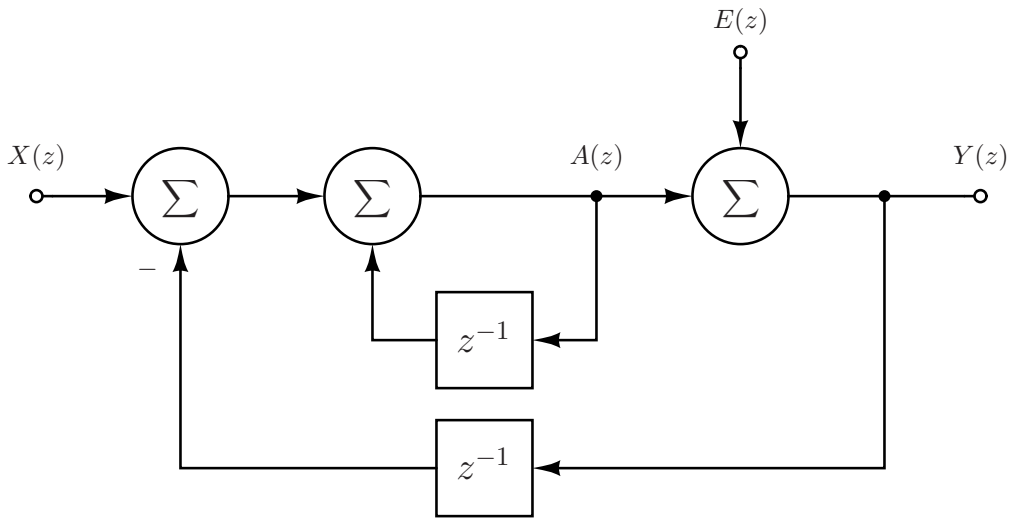

Figure 2.12: First Order Linear Model

where $A(z)$ is the accumulator output which is given as

$$A(z) = z^{-1}A(z) + X(z) - z^{-1}Y(z). \tag{2.37}$$

Substituting (2.37) into (2.36), the output, $Y(z)$, can be expressed as

$$\begin{aligned} Y(z) &= E(z) + z^{-1}A(z) + X(z) - z^{-1}Y(z) \\ &= X(z) + E(z) - z^{-1}\Big(Y(z) - A(z)\Big) \\ &= X(z) + \Big(1 - z^{-1}\Big)E(z). \end{aligned} \tag{2.38}$$

Comparing (2.38) and (2.35), it can be seen that

$$\mathrm{STF}(z) = 1 \tag{2.39}$$

and

$$\mathrm{NTF}(z) = \Big(1 - z^{-1}\Big) \tag{2.40}$$

which implies that the input signal, $X(z)$, is unaltered at the output and the quantization noise, $E(z)$, is lowpass filtered by the first order expression $(1 - z^{-1})$.

The location of the poles and zeros of the NTF and STF determine the characteristics of the NTF's and STF's frequency response. However, the $\Delta\Sigma$ modulator shown in Figure 2.12 does not allow the pole and zero locations to be adjusted. The pole locations can be adjusted by adding feedback coefficients to the $\Delta\Sigma$ modulator as illustrated in Figure 2.13. From observation of Figure 2.13, the $\Delta\Sigma$ modulator's

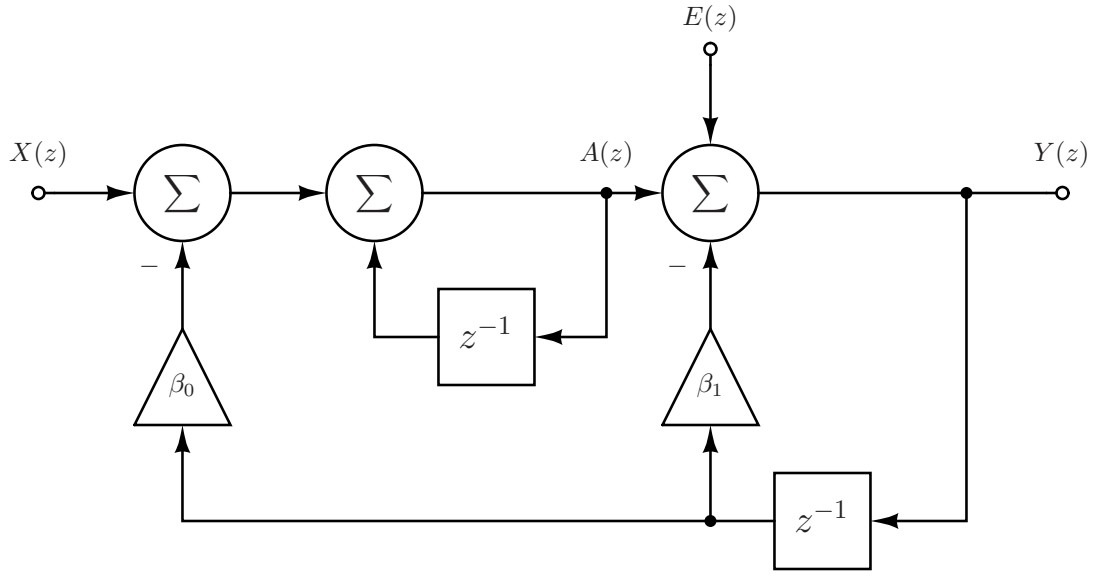

Figure 2.13: Generalized First Order Linear Model

output, $Y(z)$, is given as

$$Y(z) = E(z) + A(z) - \beta_1 z^{-1} Y(z) \tag{2.41}$$

where $A(z)$ is the accumulator output which is given as

$$\begin{aligned} A(z) &= z^{-1}A(z) + X(z) - \beta_0 z^{-1}Y(z) \\ &= \frac{X(z) - \beta_0 z^{-1}Y(z)}{(1 - z^{-1})}. \end{aligned} \tag{2.42}$$

Substituting (2.42) into (2.41) the output, $Y(z)$, can be expressed as

$$\begin{aligned} Y(z) &= E(z) + \frac{X(z) - \beta_0 z^{-1}Y(z)}{(1 - z^{-1})} - \beta_1 z^{-1}Y(z) \\ &= \frac{\Big(1 - z^{-1}\Big)E(z) + X(z)}{\Big(1 + \Big(\beta_0 + \beta_1 - 1\Big)z^{-1} - \beta_1 z^{-2}\Big)} \\ &= \frac{X(z)}{\Big(1 + \Big(\beta_0 + \beta_1 - 1\Big)z^{-1} - \beta_1 z^{-2}\Big)} + \frac{\Big(1 - z^{-1}\Big)E(z)}{\Big(1 + \Big(\beta_0 + \beta_1 - 1\Big)z^{-1} - \beta_1 z^{-2}\Big)}. \end{aligned} \tag{2.43}$$

Comparing (2.43) and (2.35), it can be seen that

$$\mathrm{STF}(z) = \frac{1}{\Big(1 + \Big(\beta_0 + \beta_1 - 1\Big)z^{-1} - \beta_1 z^{-2}\Big)} \tag{2.44}$$

and

$$\mathrm{NTF}(z) = \frac{\Big(1 - z^{-1}\Big)}{\Big(1 + \Big(\beta_0 + \beta_1 - 1\Big)z^{-1} - \beta_1 z^{-2}\Big)}. \tag{2.45}$$

For most applications, $\beta_1 = 0$. For such applications the transfer functions described by (2.44) and (2.45) can be written as

$$\mathrm{STF}(z) = \frac{1}{1 + (\beta_0 - 1)\, z^{-1}} \tag{2.46}$$

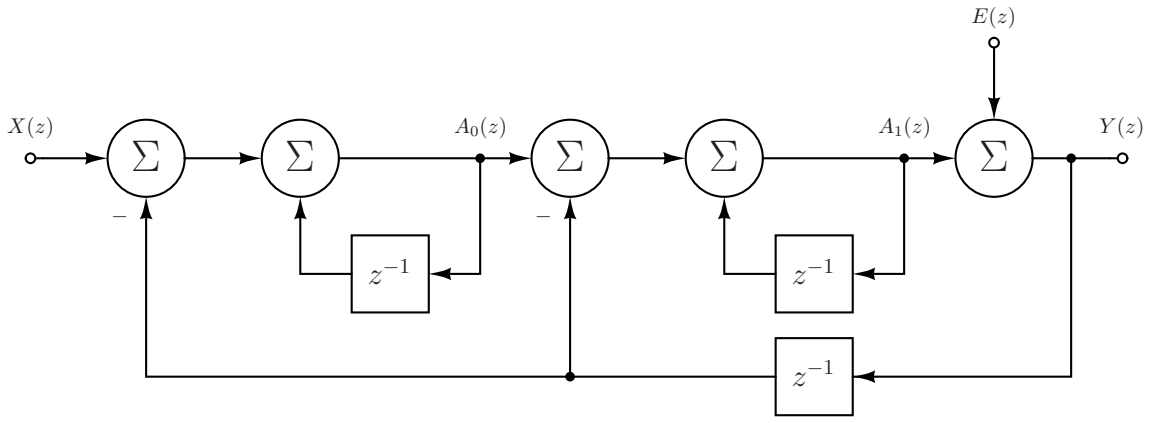

Figure 2.14: Second Order Linear Model

and

$$\mathrm{NTF}(z) = \frac{(1 - z^{-1})}{1 + (\beta_0 - 1)\, z^{-1}} \tag{2.47}$$

respectively. Thus, (2.46) and (2.47) are equivalent to (2.39) and (2.40) when $\beta_0 = 1$.

2.3.4.2 Second Order System

Because the NTFs of 1st order $\Delta\Sigma$ modulators have a limited amount of quantization noise attenuation, higher order systems are generally used. Consider the second order system illustrated in Figure 2.14. This system can be formed by cascading two first order systems together. From observation of Figure 2.14, the second order $\Delta\Sigma$ modulator's output, $Y(z)$, is given as

$$Y(z) = E(z) + A_1(z) \tag{2.48}$$

where $A_1(z)$ corresponds to the output of the second accumulator. The accumulator outputs, A_0 and A_1, can be expressed as

$$A_0(z) = \frac{X(z) - z^{-1}Y(z)}{1 - z^{-1}} \tag{2.49}$$

and

$$A_1(z) = \frac{A_0(z) - z^{-1}Y(z)}{1 - z^{-1}}. \tag{2.50}$$

Substituting (2.49) into (2.50) and the result into (2.48), the output, $Y(z)$, can be expressed as

$$Y(z) = X(z) + \left(1 - z^{-1}\right)^2 E(z). \tag{2.51}$$

Comparing (2.51) and (2.35), it can be seen that

$$\mathrm{STF}(z) = 1 \tag{2.52}$$

and

$$\mathrm{NTF}(z) = \left(1 - z^{-1}\right)^2 \tag{2.53}$$

which implies that the input signal, $X(z)$, is unaltered at the output and the quantization noise, $E(z)$, is lowpass filtered by the second order expression $(1 - z^{-1})^2$.

To adjust the pole locations of the 2nd order $\Delta\Sigma$ modulator shown in Figure 2.14 feedback coefficients, denoted as β_0, β_1, and β_2, can be added as shown in Figure 2.15. From observation of Figure 2.15, the $\Delta\Sigma$ modulators's output, $Y(z)$, is given as

$$Y(z) = E(z) + A_1(z) - \beta_2 z^{-1} Y(z) \tag{2.54}$$

where $A_1(z)$ corresponds to the output of the second accumulator. The accumulator outputs, A_0 and A_1, can be expressed as

$$A_0(z) = \frac{X(z) - \beta_0 z^{-1} Y(z)}{1 - z^{-1}} \tag{2.55}$$

and

$$A_1(z) = \frac{A_0(z) - \beta_1 z^{-1} Y(z)}{1 - z^{-1}} = \frac{X(z) - (\beta_0 + \beta_1) z^{-1} Y(z)}{(1 - z^{-1})^2} \tag{2.56}$$

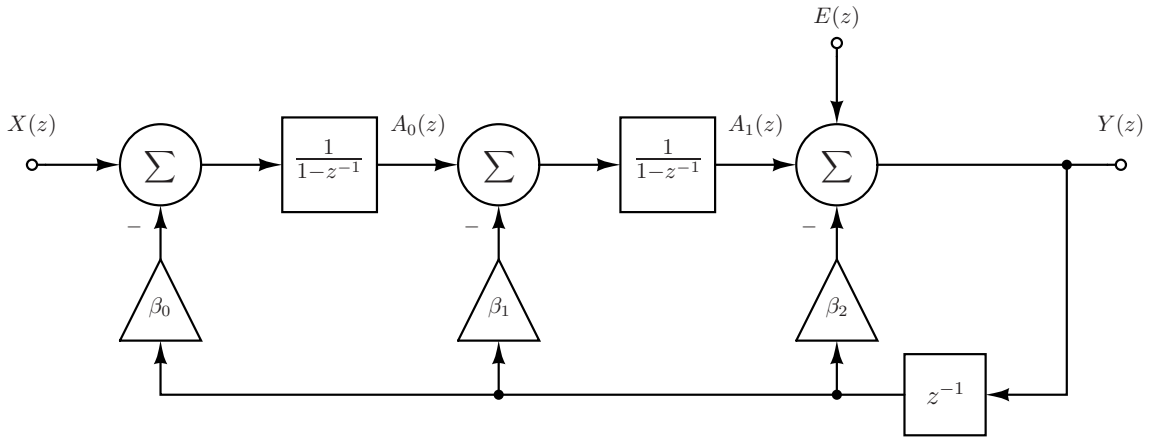

Figure 2.15: Generalized Second Order Linear Model

Substituting (2.55) into (2.56) and the result into (2.54), the output, $Y(z)$, can be expressed as

$$\begin{aligned} Y(z) &= E(z) + \frac{X(z) - (\beta_0 + \beta_1)z^{-1}Y(z)}{(1 - z^{-1})^2} - \beta_2 z^{-1}Y(z) \\ &= \frac{X(z) + \left(1 - z^{-1}\right)^2 E(z)}{1 + z^{-1}\left(-2 + \beta_0 + \beta_1 + \beta_2\right) + z^{-2}\left(1 - \beta_1 - 2\beta_2\right) + z^{-3}\beta_2}. \end{aligned} \tag{2.57}$$

Comparing (2.57) and (2.35), it can be seen that

$$\mathrm{STF}(z) = \frac{1}{1 + z^{-1}\left(-2 + \beta_0 + \beta_1 + \beta_2\right) + z^{-2}\left(1 - \beta_1 - 2\beta_2\right) + z^{-3}\beta_2} \tag{2.58}$$

and

$$\mathrm{NTF}(z) = \frac{\left(1 - z^{-1}\right)^2}{1 + z^{-1}\left(-2 + \beta_0 + \beta_1 + \beta_2\right) + z^{-2}\left(1 - \beta_1 - 2\beta_2\right) + z^{-3}\beta_2}. \tag{2.59}$$

For most applications, $\beta_2 = 0$. For such applications the transfer functions de-

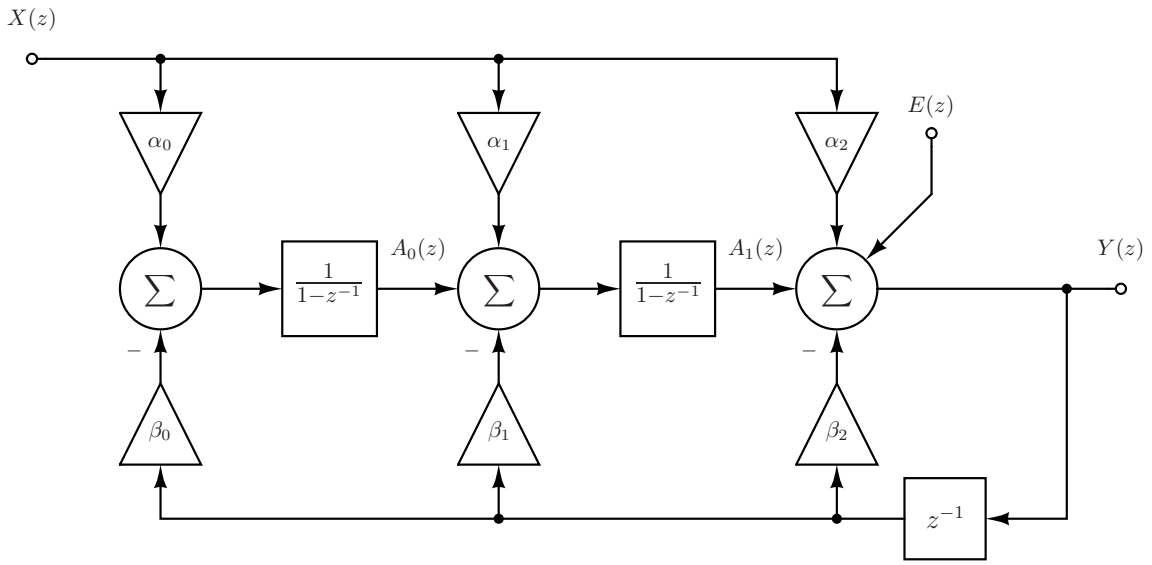

Figure 2.16: Generalized Second Order Linear Model with Feedforward Coefficients

scribed by (2.58) and (2.59)) can be written as

$$\mathrm{STF}(z) = \frac{1}{1 + z^{-1}\Big(-2 + \beta_0 + \beta_1\Big) + z^{-2}\Big(1 - \beta_1\Big)} \tag{2.60}$$

and

$$\mathrm{NTF}(z) = \frac{(1 - z^{-1})^2}{1 + z^{-1}\Big(-2 + \beta_0 + \beta_1\Big) + z^{-2}\Big(1 - \beta_1\Big)} \tag{2.61}$$

respectively. Thus, (2.60) and (2.61) are equivalent to (2.52) and (2.53) when $\beta_0 = \beta_1 = 1$.

Because a $\Delta\Sigma$ modulator's NTF is designed first, the feedback coefficients, $\{\beta_n\}$, are chosen to optimize the NTF's characteristics. As such, the STFs in (2.58) and (2.60) are fixed by the NTF's design. To shape the STF, feedforward coefficients can be added to Figure 2.15 as illustrated in Figure 2.16. From observation of Figure

2.16, the $\Delta\Sigma$ modulator's output, $Y(z)$, is given as

$$Y(z) = E(z) + \alpha_2 X(z) + A_1(z) - \beta_2 z^{-1} Y(z) \tag{2.62}$$

where $A_1(z)$ corresponds to the output of the second accumulator. The accumulator outputs, A_0 and A_1, can be expressed as

$$A_0(z) = \frac{\alpha_0 X(z) - \beta_0 z^{-1} Y(z)}{1 - z^{-1}} \tag{2.63}$$

and

$$A_1(z) = \frac{\alpha_1 X(z) + A_0(z) - \beta_1 z^{-1} Y(z)}{1 - z^{-1}}. \tag{2.64}$$

Substituting (2.63) into (2.64) and the result into (2.62), the output, $Y(z)$, can be expressed as

$$\begin{aligned} Y(z) = &\left(\frac{\Big(\alpha_0 + \alpha_1 + \alpha_2\Big) - z^{-1}\Big(\alpha_1 + 2\alpha_2\Big) + z^{-2}\alpha_2}{1 + z^{-1}\Big(-2 + \beta_0 + \beta_1 + \beta_2\Big) + z^{-2}\Big(1 - \beta_1 - 2\beta_2\Big) + z^{-3}\beta_2} \right) X(z) \\ &+ \left(\frac{\Big(1 - z^{-1}\Big)^2}{1 + z^{-1}\Big(-2 + \beta_0 + \beta_1 + \beta_2\Big) + z^{-2}\Big(1 - \beta_1 - 2\beta_2\Big) + z^{-3}\beta_2} \right) E(z). \end{aligned} \tag{2.65}$$

Comparing (2.65) and (2.35), it can be seen that

$$\mathrm{STF}(z) = \frac{\Big(\alpha_0 + \alpha_1 + \alpha_2\Big) - z^{-1}\Big(\alpha_1 + 2\alpha_2\Big) + z^{-2}\alpha_2}{1 + z^{-1}\Big(-2 + \beta_0 + \beta_1 + \beta_2\Big) + z^{-2}\Big(1 - \beta_1 - 2\beta_2\Big) + z^{-3}\beta_2} \tag{2.66}$$

and

$$\mathrm{NTF}(z) = \frac{\Big(1 - z^{-1}\Big)^2}{1 + z^{-1}\Big(-2 + \beta_0 + \beta_1 + \beta_2\Big) + z^{-2}\Big(1 - \beta_1 - 2\beta_2\Big) + z^{-3}\beta_2}. \tag{2.67}$$

It can be observed from (2.66) and (2.67) that the feedforward coefficients, α_0, α_1, and α_2, only affect the STF and not the NTF. As such, this allows the shape of the STF to be changed independently from the NTF. It also can be seen that (2.66) and (2.67) are equivalent to (2.58) and (2.59) for $\alpha_1 = \alpha_2 = 0$ and $\alpha_0 = 1$.

2.3.4.3 High Order Systems

Theoretically, a $\Delta\Sigma$ modulator's order, n, has no upper bound and thus, the NTF's stopband attenuation can be increased to any arbitrarily large level. This in turn would allow the effective resolution of the $\Delta\Sigma$ modulator to increase without bound. However, physical phenomena such as thermal noise and clock jitter typically limit a $\Delta\Sigma$ modulator's achievable effective resolution. Therefore, in practice, $\Delta\Sigma$ modulators are typically designed such that $n \leq 8$.

Additionally, because the internal voltage swings of a $\Delta\Sigma$ modulator are limited by the electrical characteristics of its process technology, scaling coefficients are typically placed between adjacent integrators to avoid saturation. Saturation, or clipping, can cause instability and introduces nonlinear distortion thereby decreasing the effective resolution of the $\Delta\Sigma$ modulator. Figure 2.17 illustrates a generalized nth order converter topology with scaling coefficients, denoted c_x, between adjacent integrators where x corresponds to the coefficient's respective integrator number.

2.3.5 $\Delta\Sigma$ Modulator Theoretical Performance

The theoretical effective resolution of a $\Delta\Sigma$ modulator can be calculated if the shape of in-band NTF is known. That is, stochastic system theory can be used to predict the performance of a deterministic system function for a particular input, $x(n)$, and the randomly modeled quantization noise, $q(n)$.

Stochastic system theory states that for an LTI system, an input signal, $x(n)$, which can be modeled as a discrete-time random process, will produce an output signal, $y(n)$, which can also be modeled as a discrete-time random process [27]. As

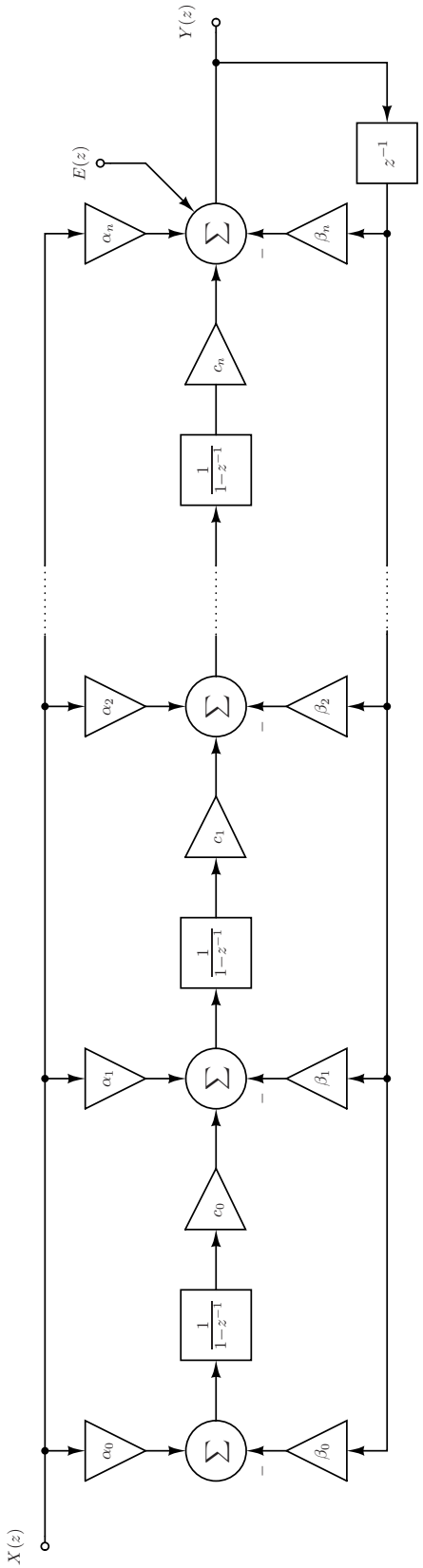

Figure 2.17: Generalized nth Order Linear Model

such, descriptive statistics (e.g. mean, variance, and autocorrelation) are often used to analyze LTI system's output signals when the input signals can be modeled as random processes.

The autocorrelation, $R_{x(n)}(k)$, of a discrete-time random process, $x(n)$, can be expressed as

$$R_{x(n)}(k) = E\left[x(n)x(n+k)\right] \tag{2.68}$$

where $E[\cdot]$ denotes the expectation operator as defined in (2.7) [16]. If $x(n)$ is a zero mean, random process then the autocorrelation of $x(n)$ for $k = 0$, $R_{x(n)}(0)$, is equivalent to the variance, $\sigma^2_{x(n)}$, of $x(n)$, as described by (2.9); that is,

$$R_{x(n)}(0) = E\left[x(n)x(n+0)\right] = E\left[x^2(n)\right] = \sigma^2_{x(n)}. \tag{2.69}$$

As such, the average power, $P_{x(n)}$, of a zero mean, random process, $x(n)$, can be expressed as

$$P_{x(n)} = R_{x(n)}(0). \tag{2.70}$$

It has been shown [26] that the power spectral density, $\boldsymbol{S}_{x(n)}(e^{j\omega})$, of a discrete-time random process, $x(n)$, can be calculated by taking the Fourier transform of its autocorrelation, $R_{x(n)}(k)$; that is,

$$\boldsymbol{S}_{x(n)}(e^{j\omega}) = \sum_{k=-\infty}^{\infty} R_{x(n)}(k)e^{-j\omega k} \tag{2.71}$$

where ω denotes frequency in radians per sample. Conversely, the autocorrelation of a discrete-time random process, $R_{x(n)}(k)$, can be calculated by taking the inverse Fourier transform of its power spectral density, $\boldsymbol{S}_{x(n)}(e^{j\omega})$, which implies that

$$R_{x(n)}(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \boldsymbol{S}_{x(n)}(e^{j\omega})e^{j\omega k}d\omega. \tag{2.72}$$

It has also been shown [16] [26] that the output power spectral density, $\boldsymbol{S}_{y(n)}(e^{j\omega})$, of a LTI system that has a frequency response, $H(e^{j\omega})$, can be calculated by taking the product of the input power spectral density, $\boldsymbol{S}_{x(n)}(e^{j\omega})$, and the magnitude-squared of the system's frequency response; that is,

$$\boldsymbol{S}_{y(n)}(e^{j\omega}) = |H(e^{j\omega})|^2 \boldsymbol{S}_{x(n)}(e^{j\omega}). \tag{2.73}$$

Therefore using (2.72) and (2.73), the autocorrelation, $R_{y(n)}(k)$, of the output, $y(n)$, of an LTI system that has the input $x(n)$ can be written as

$$\begin{aligned} R_{y(n)}(k) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \boldsymbol{S}_{y(n)}(e^{j\omega}) e^{j\omega k} d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |H(e^{j\omega})|^2 \boldsymbol{S}_{x(n)}(e^{j\omega}) e^{j\omega k} d\omega. \end{aligned} \tag{2.74}$$

Using (2.70) and (2.74), the average output power, $P_{y(n)}$, of an LTI system can be calculated as

$$P_{y(n)} = R_{y(n)}(0) = \frac{1}{2\pi}\int_{-\pi}^{\pi} |H(e^{j\omega})|^2 \boldsymbol{S}_{x(n)}(e^{j\omega}) d\omega. \tag{2.75}$$

Because a $\Delta\Sigma$ modulator's NTF is modeled as a LTI system, its output quantization noise power, $P_{q(n)}$, can be calculated using (2.75); that is,

$$P_{q(n)} = \frac{1}{2\pi}\int_{-\omega_0}^{\omega_0} |\mathrm{NTF}(e^{j\omega})|^2 \boldsymbol{S}_{e(n)}(e^{j\omega}) d\omega \tag{2.76}$$

where ω_0 corresponds to the Nyquist frequency of the input signal in radians per sample.

To determine the quantization noise power spectral density, $\boldsymbol{S}_{e(n)}(e^{j\omega})$, the quantization noise is assumed to be a zero mean, uncorrelated white noise process, which

implies that its autocorrelation, $R_{e(n)}(k)$, can be written as

$$R_{e(n)}(k) = E\left[e(n)e(n+k)\right] = E\left[e^2(n)\right]\delta(k) = \sigma^2_{e(n)}\delta(k). \tag{2.77}$$

Therefore, the power spectral density of the quantization noise, $\boldsymbol{S}_{e(n)}(e^{j\omega})$, can be written as

$$\boldsymbol{S}_{e(n)}(e^{j\omega}) = \sum_{k=-\infty}^{\infty} \sigma^2_{e(n)}\delta(k)e^{-j\omega k} = \sigma^2_{e(n)} = P_{e(n)}. \tag{2.78}$$

Thus, substituting (2.26) into (2.78), it can be seen that

$$\boldsymbol{S}_{e(n)}(e^{j\omega}) = \frac{\Delta^2}{12} \tag{2.79}$$

where Δ is the quantization interval. Substituting (2.79) into (2.76), the $\Delta\Sigma$ modulator's output quantization noise power, $P_{q(n)}$, can be expressed as

$$\begin{aligned} P_{q(n)} &= \frac{1}{2\pi}\int_{-\omega_0}^{\omega_0} |\mathrm{NTF}(e^{j\omega})|^2 \boldsymbol{S}_{e(n)}(e^{j\omega})d\omega \\ &= \frac{1}{2\pi}\left(\frac{\Delta^2}{12}\right)\int_{-\omega_0}^{\omega_0} |\mathrm{NTF}(e^{j\omega})|^2 d\omega \end{aligned} \tag{2.80}$$

where ω_0 corresponds to the input signal's Nyquist bandwidth.

For the nth order discrete-time $\Delta\Sigma$ modulator architecture shown in Figure 2.17 where $a_n = b_n = c_n = 1$, the NTF can be written as

$$\mathrm{NTF}(z) = (1 - z^{-1})^n \tag{2.81}$$

which implies that

$$\mathrm{NTF}(e^{j\omega}) = (1 - e^{-j\omega})^n = \left(j2e^{-j\frac{\omega}{2}}\left(\frac{e^{j\frac{\omega}{2}} - e^{-j\frac{\omega}{2}}}{j2}\right)\right)^n. \tag{2.82}$$

Using Euler's identity [39], (2.82) can be written as

$$\mathrm{NTF}(e^{j\omega}) = \left(j2e^{-j\frac{\omega}{2}} \sin\left(\frac{\omega}{2}\right)\right)^n. \tag{2.83}$$

Thus,

$$|\mathrm{NTF}(e^{j\omega})|^2 = \left| \left(j2e^{-j\frac{\omega}{2}} \sin\left(\frac{\omega}{2}\right)\right)^n \right|^2 = \left(2\sin\left(\frac{\omega}{2}\right)\right)^{2n}. \tag{2.84}$$

Substituting (2.84) into (2.80), the quantization noise power, $P_{q(n)}$, can be written as

$$P_{q(n)} = \frac{1}{2\pi}\left(\frac{\Delta^2}{12}\right) \int_{-\omega_0}^{\omega_0} \left(2\sin\left(\frac{\omega}{2}\right)\right)^{2n} d\omega. \tag{2.85}$$

For large OSRs, $\omega_0 \ll \pi$, and therefore,

$$\sin\left(\frac{\omega}{2}\right) \approx \frac{\omega}{2}.$$

[35] [39] [21]. Substituting this approximation into (2.85), the output quantization noise power, $P_{q(n)}$, can be given as

$$\begin{aligned} P_{q(n)} &= \frac{1}{2\pi}\left(\frac{\Delta^2}{12}\right) \int_{-\omega_0}^{\omega_0} \omega^{2n} d\omega \\ &= \frac{1}{2\pi}\left(\frac{\Delta^2}{12}\right) \left(\frac{\omega^{2n+1}}{2n+1}\Big|_{-\omega_0}^{\omega_0} \right) \\ &= \frac{\Delta^2}{12\pi}\left(\frac{1}{2n+1}\right)\omega_0^{2n+1}. \end{aligned} \tag{2.86}$$

Substituting $\omega_0 = \pi/M$ into (2.86), the output quantization noise power, $P_{q(n)}$, can be expressed as

$$P_{q(n)} = \frac{\Delta^2}{12\pi}\left(\frac{1}{2n+1}\right)\left(\frac{\pi}{M}\right)^{2n+1} \tag{2.87}$$

where M denotes the OSR.

The theoretical $\mathrm{SNR_{dB}}$, $\mathrm{SNR_{dB,LPDSM}}$, for a lowpass $\Delta\Sigma$ modulator that has a

full-scale sinusoidal input as defined by (2.25), can be derived by substituting (2.87) into (2.21) such that

$$\begin{aligned}\mathrm{SNR}_{\mathrm{dB,LPDSM}} &= 10\log\frac{\left(\frac{\Delta^2 2^{2B}}{8}\right)}{\frac{\Delta^2}{12\pi}\left(\frac{1}{2n+1}\right)\left(\frac{\pi}{M}\right)^{2n+1}} \\ &= 10\log\left(\frac{3}{2}2^{2B}\right) + 10\log\left(2n+1\right) - 2n10\log(\pi) + (2n+1)10\log(M) \\ &= 6.02B + 10\log(2n+1) - 20n\log(\pi) + (20n+10)\log(M)\end{aligned} \tag{2.88}$$

where M corresponds to the OSR and B corresponds to the number of quantization bits.

From observation of (2.88), it can be seen that the dominant term, that is the term which has the greatest impact on SNR, in (2.88) is $(20n+10)\log(M)$ for $M \gg 1$ which implies that the effective resolution for a $\Delta\Sigma$ modulator is largely determined by the OSR and the order of the loop filter.

CHAPTER 3

OPTIMIZATION AND GENETIC ALGORITHMS

Global optimization of multimodal and non-differentiable objective functions continues to be an open research topic in the field of numerical optimization. Traditional numerical techniques, such as linear programming and the simplex method, have been applied with great success to linearly constrained objective functions [9]. However, for certain types of problems they have been shown to determine suboptimal solutions [32]. Also, when linear programming techniques are applied to multimodal performance surfaces, constraints must be selected according to detailed knowledge of the performance surface topology. On the other hand, genetic algorithms, which are a class of optimization algorithms that are loosely based on the principles of evolution and genetics, have successfully determined globally optimal solutions of multimodal and non-differentiable objective functions for which linear programming algorithms could not determine the global optimum [11]. Genetic algorithms search a solution space by using genetic operators and cumulative information to reduce a solution space and generate a set of viable solutions. Some more recent genetic algorithms use orthogonal crossover operators and have been shown to perform remarkably well for classical challenging problems [24]. In this chapter a new algorithm called a hybrid orthogonal genetic (HOG) algorithm that uses customized genetic operators to improve algorithm performance and accuracy is applied to multimodal, non-differentiable performance surfaces.

3.1 Genetic Algorithms

Figure 1 illustrates the flow chart of a typical genetic algorithm (GA). In GAs, a population is defined as a set of N individuals where individuals represent possible solutions. At the outset, the population of N individuals is initialized and represents the first generation, denoted G_0, of the population's existence. If the solution space topology is unknown, the rationale of the genetic algorithm is referred to as an exploratory effort. Conversely, if the solution is known to exist in a localized area of the solution space, the rationale is referred to as an exploitative effort. For exploration of the performance surface, it is common to initialize the population with individuals whose elements, characteristics, or traits, are uniformly distributed over the solution space. However, to exploit localized areas of the performance surface, the population is initialized with individuals whose traits are known to be near optimal within some acceptable range of misadjustment.

Subsequent to initialization, each member of the population is evaluated by the objective function, which is a metric of solution quality or fitness. Thus, an individual's fitness is represented by the value, or cost, returned by the objective function evaluation. Individuals are then selected according to their relative fitness for placement into a mating pool which is a subset of the population. Individuals with a higher relative fitness have a higher likelihood of selection for mating eligibility while individuals with a lower relative fitness are more likely to be discarded. Individuals which have been selected and placed into the mating pool then reproduce via a crossover operator where reproduction is defined as the random exchange of traits between selected individuals (progenitors) who produce offspring (progeny) and the crossover operator is the algorithmic mechanism by which reproduction occurs. Following reproduction, genetic diversity of the population is generated by introducing new genetic information into the population via a mutation operator where mutation

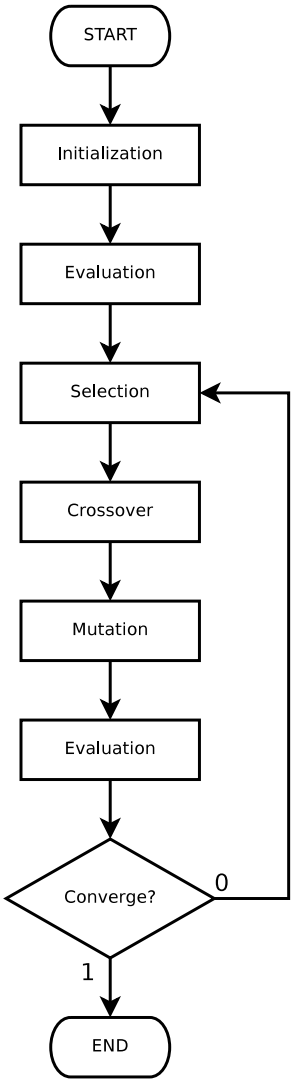

Figure 3.1: Traditional Genetic Algorithm Flow Chart

is defined as the random alteration of a selected individual's traits.

Each member of the new population, G_1, which is now comprised of the mating pool and their newly generated offspring, is evaluated for fitness. The new generation's fitness is then analyzed to see if the convergence criteria has been met. If the convergence criteria has not been met then the population must undergo selection, reproduction, and mutation again and be reevaluated for convergence. This cycle continues until the convergence criteria has been met.

3.2 Hybrid Orthogonal Genetic (HOG) Algorithm

Figure 3.2 shows a flow chart of the HOG algorithm which begins by initializing the population with a random selection of viable solutions, referred to as individuals or chromosomes. Structurally, each chromosome is represented by a vector of length K where each element or allele of the vector represents a trait which can be viewed as genetic information in the chromosome. A population of size N can be represented by aggregating the chromosomes into a $K \times N$ matrix. Following population initialization, each chromosome is evaluated for fitness by the objective function. The chromosomes are then sorted and linearly ranked according to their fitness and selected for placement into a mating pool according to their relative fitness. Once selected for mating eligibility, pairs of chromosomes reproduce via a traditional crossover operator producing a pair of offspring. The offspring and mating pool then reproduce a second time via a hybrid orthogonal crossover operator. Genetic diversity is then generated through the use of a mutation operator. Finally, the new generation's fitness is evaluated and the results are examined for convergence. If convergence conditions are not met then the process repeats itself until the convergence conditions are satisfied.

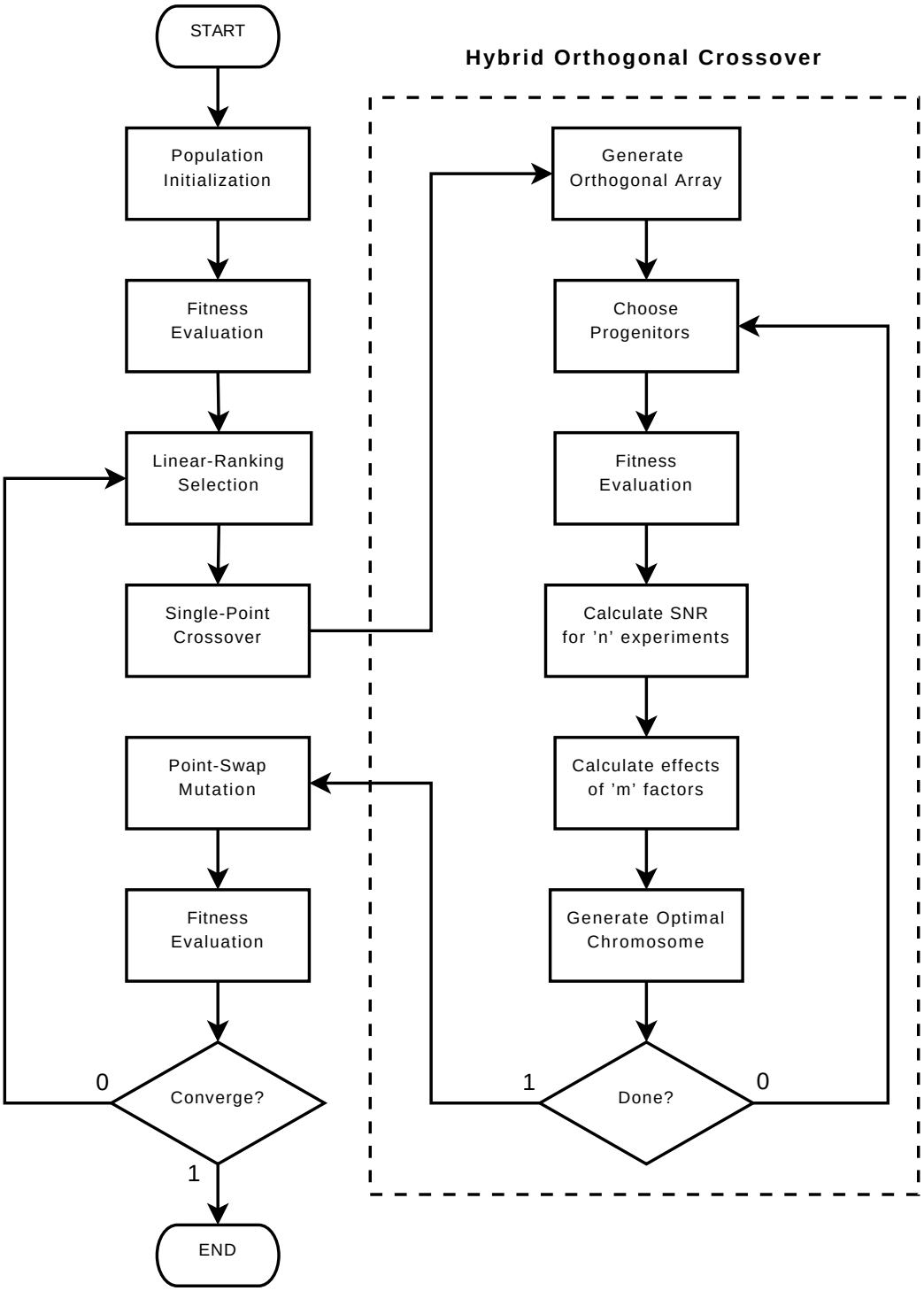

Figure 3.2: Hybrid Orthogonal Genetic Algorithm Flow Chart

3.2.1 Population Initialization

In the HOG algorithm, the initial population can be chosen such that the chromosomes are uniformly distributed over some subset of the solution space. Typically, the initial population is determined by the nature of the objective function. If the boundaries of the objective function's feasible solution space are well understood then the initial population can be selected over the known subset of this feasible solution space. However, if the boundaries of the feasible solution space are unknown or poorly understood then it may become necessary to iteratively modify the problem space as knowledge of the performance surface is gained through trial and error.

3.2.2 Fitness Evaluation

As mentioned previously, the fitness of an individual is determined by evaluating the objective function for that individual. Because the HOG algorithm is a minimization based evolutionary strategy, individuals with lower cost are considered more fit than individuals with a higher cost. Thus, the HOG algorithm is searching for the individual $\mathbf{x}$ that satisfies

$$\min_{\mathbf{x}\in\mathbb{R}^K}\Big\{J(\mathbf{x})\Big\}, \quad \mathbf{x}\in S \tag{3.1}$$

where $J(\mathbf{x})$ is the cost or objective function, $\mathbf{x}$ is a real vector in $\mathbb{R}^K$, and S is the set of feasible solutions.

3.2.3 Linear-Ranking Selection

After the individuals in the population are linearly ranked according to their fitness, each individual is assigned a probability of selection for reproduction such that more fit individuals have a higher likelihood of reproducing than less fit individuals. Individuals not selected for reproduction are discarded. In this thesis, the individual with the best fitness is always selected for mating eligibility ensuring that the most fit member of any generation is eligible for reproduction. Such algorithms are referred

to as elitist algorithms where elitist is defined as preserving the most fit individual for the next generation independent from the evolutionary process. The HOG algorithm implements an elitist, linear-ranking selection scheme to decrease stochastic selection variability and improve the representation of highly dynamic populations for mating eligibility.

In biological terms, the genotype of a population is defined as the total available genetic information belonging to that population. The phenotype of a population is defined as the expressed traits of the population where expression is defined as the observable display of characteristics related to a particular genetic composition. In evolutionary systems, phenotypic expression and genotypic content are often very loosely coupled and extremely difficult to characterize. As such, the observed fitness of an individual belonging to some population offers only a partial indication of possible reproductive optimality. Care must be taken that the evolutionary process does not inadvertently discard the genetic information contained in less-fit individuals which may be required to achieve the optimal chromosome. Prior to convergence, the optimal chromosome typically exists as some permutation of traits belonging to individuals which cannot be guaranteed to have the highest relative fitness. In fact, population dynamics typically exist such that the relative difference between the most fit and least fit individual is poorly reflected in its observed fitness even for well formed objective functions [3]. As such, for most objective functions, the fitness metric is not a good metric for determining an individual's breeding potential.

To mitigate the statistical selection bias inherent to the objective function, a linear ranking scheme can be implemented where the population is ordered according to its fitness prior to selection for mating eligibility [43]. In the HOG algorithm, the population of N individuals is sorted according to their fitness values. Once sorted, the individuals from the least fit (highest cost) to most fit (lowest cost) are assigned

consecutive integers from 1 to N; that is, the least fit individual is assigned one and the most fit individual is assigned N. The probability of selection for mating eligibility is then assigned according to the individual's rank within the greater population where the ith ranked individual has the probability, p_i, of selection, given as

$$p_i = \frac{1}{N}\left(\eta^- + \left(\eta^+ - \eta^-\right)\frac{i-1}{N-1}\right) \tag{3.2}$$

where $\left(\eta^-/N\right)$ and $\left(\eta^+/N\right)$ are the probabilities of selection for the least fit and most fit individual, respectively. Because the population is comprised of N disjoint elementary events,

$$\sum_{i=1}^{N} p_i = \sum_{i=1}^{N} \frac{1}{N}\left(\eta^- + \left(\eta^+ - \eta^-\right)\frac{i-1}{N-1}\right) = 1. \tag{3.3}$$

which implies that $\eta^- = 2 - \eta^+$ where $0 \leq \eta^+ \leq 2$.

Specific selection of η^- and η^+ allows for control over selection pressure which is defined as the relationship between the probability of selection of the most fit vs. least fit individual. It has been shown that fixing η^+ at 1.1 provides an adequate balance between exploration and exploitation of the performance surface [2].

Because standard operators which rely on stochastic selection where the probability of selection is proportional to the individual's relative fitness cannot guarantee that the most fit individuals will be considered for selection and subsequent reproduction, the possibility exists that good chromosomes may be randomly discarded. For the HOG algorithm, an elitist selection method is implemented to counter this phenomenon, where elitism is defined as the process of automatically selecting the best chromosome(s) for mating eligibility thereby ensuring the availability of their genetic information for subsequent reproduction. This technique has been shown to greatly increase convergence speeds, especially for applications where minimizing

steady-state misadjustment is significant [31].

3.2.4 Single Point Crossover

After randomly selecting a group of eligible progenitors, or a mating pool, using the selection probabilities, randomly paired individuals reproduce by exchanging alleles where alleles are the elements comprising the vector structure of the chromosome. In genetic algorithms, this process is referred to as crossover. This sharing of genetic information is the vehicle by which individuals propagate their beneficial traits.

For both single-point and hybrid orthogonal crossover, pairs of progenitors are randomly selected from the mating pool and are given the opportunity to exchange genetic information via the respective crossover operator. Each selected pair is assigned a random number, r that has a uniform distribution over $\Big[0, 1\Big]$. Selection for crossover is governed by the coefficient for crossover probability of, P_c, which typically ranges from 0.2 to 1.0. If $r < P_c$, crossover occurs and genetic information is exchanged between the pair of progenitors and offspring are produced. Conversely, if $r \geq P_c$, the pair is returned to the mating pool without producing offspring. Note that the process by which individuals are randomly selected for crossover is typically regulated. For this application, self replication is strictly forbidden. Thus, selected individuals must crossover with a different individual thereby ensuring the exchange of alleles and preventing a single anomalous individual from inadvertently dominating the greater population leading the algorithm to converge to a non-optimal solution.

Recall that reproduction is defined as the exchange of genetic information between two individuals belonging to a population and that the mechanism of exchange of vector elements between the individuals is referred to as the crossover operator. Many different crossover techniques have been analyzed and implemented successfully across a broad range of objective function types [28] [12]. Because the objective function both characterizes the problem space and evaluates specific chromosomes,

the physical structure of the chromosomes is determined by the characteristics of the objective function. Thus, the optimal crossover operator is related to the physical structure of the chromosome. For certain applications, it may be necessary to regulate which alleles can be exchanged. These regulations are dictated by the relationships which exist between adjacent groups of alleles. The power of evolutionary strategies lies in achieving a balance between exploring a solution space and exploiting its localized minima or maxima. As a result, a balance between preserving the stochastic nature of reproduction and the deterministic exchange of genetic information must be maintained. Thus, the structure of the chromosome should complement the crossover method selected. For the HOG algorithm, reproduction is achieved by using both a single-point crossover operator and a hybrid orthogonal crossover operator based on the Taguchi method.

The single-point crossover operator is an arithmetic operator derived from convex set theory which randomly selects a single crossover point for the selected chromosomes [41]. For two individuals represented by the real vectors $\mathcal{C}_\alpha$ and $\mathcal{C}_\beta$, all alleles subsequent to the crossover point are exchanged between the progenitors and two progeny are created. The single-point crossover process is illustrated in Figure 3.3 where the progenitor chromosomes are denoted as $\mathcal{C}_\alpha$ and $\mathcal{C}_\beta$ and the progeny chromosomes are denoted as $\mathcal{C}'_\alpha$ and $\mathcal{C}'_\beta$.

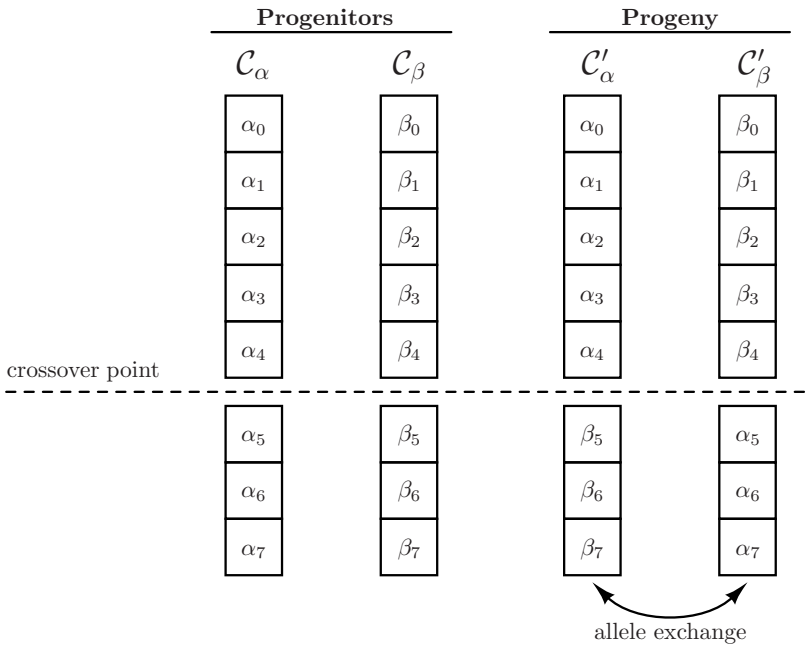

Figure 3.3: Single-Point Crossover

Single-point crossover has positional bias in that it favors continuous segments of genetic information. However, it does not contain distribution bias as the crossover point is a discrete random variable with uniform distribution over $\Big[0, m\Big]$, where m denotes the length of the respective chromosome [12].

3.2.5 Hybrid Orthogonal Crossover via The Taguchi Method

After traditional single-point crossover has been performed, the previously selected mating pool and newly generated progeny undergo an additional exchange of genetic information using a hybrid crossover technique which intelligently creates more fit offspring. Unlike the traditional crossover operator, the hybrid orthogonal crossover operator intelligently draws genetic information from each progenitor to create the best possible offspring given the traits that are available. Based on the

Taguchi method and using orthogonal-array based experimental design techniques, this method has been shown to produce the most robust offspring possible given the genetic information available from the current population [40]. It has also been shown to be significantly less sensitive to ill-formed objective functions and non-linear performance surfaces and to improve both solution accuracy and overall convergence speed [41].

The progenitors are randomly selected from the eligible mating pool as is done for the traditional crossover operator. However, unlike traditional crossover operators, all permutations of possible progeny for a selected pair of progenitors are considered with the hybrid operator. Each permutation of genetic exchange between the progenitors is treated as an experiment (trial) where the factors of the experiment correspond to the available traits. The subsequent fitness evaluation of the progeny can then be treated as the observation of an experimental trial of K factors where K is the length of each chromosome [23] [41]. This type of experiment is commonly referred to as a factorial experiment of K factors. As such, proven methods from the statistical design of experiments can be used to improve the experimental process. Specifically, statistical design of experiments is the process of planning experiments so that significant data can be collected with a minimum number of performed experiments. Subsequent to data collection, suitable statistical methods are then used to analyze the collected data and draw statistical inferences accordingly [15].

3.2.5.1 Design of Experiments and Orthogonal Arrays

For a factorial experiment of K factors, there exists N^K trials where N represents the number of levels for each factor. For the hybrid orthogonal crossover operator, $N = 2$ because only two progenitors are available from which to draw traits. Thus, 2^K experiments must be performed to determine the most optimal progeny from the full factorial experiment space where K corresponds to the number of traits or

experimental factors. Because each permutation must be evaluated by the objective function to determine the experimental results for that particular trial, performing all 2^K objective function evaluations becomes computationally prohibitive for large values of K. As such, fractional factorial design of experiments can be used to reduce the number of required experimental trials. The HOG algorithm uses orthogonal array based design of experiments to reduce the full factorial solution space to a small but representative number of sample trials [23].

The Taguchi method of design of experiments utilizes two-level orthogonal arrays which are special arrays derived from the Latin square. An M row and N column Latin square, denoted as L_M, has the form

$$L_M\left(Q^N\right) = \left[a_{i,j}\right]_{M\times N}, \quad a_{i,j} \in \{1, 2, \cdots, Q\} \tag{3.4}$$

where the M rows represent the experimental trials, the N columns represent the experimental factors, and Q corresponds to the number of levels of the experimental design [23]. Recall that the hybrid orthogonal crossover operator uses two-level orthogonal arrays. As such, substituting $Q = 2$ into (3.4) yields

$$L_n\left(2^{n-1}\right) \tag{3.5}$$

where n corresponds to the number of trials. The resulting Latin square then represents an orthogonal set of experiments where orthogonal is defined as the statistical independence of the columns representing experimental factors. The observations made over n trials of orthogonal experiments yields the effects of the experimental factors. In particular, the HOG algorithm uses this information to determine which factors have the most beneficial contribution to the progeny.

To illustrate, consider an experiment that has 3 factors (alleles) and 2 levels

(progenitors). For this experiment, 2^3, or 8, unique experiments exist in the full factorial experiment space. The Latin square which represents the experimental space is given as

$$L_4\left(2^3\right) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{pmatrix} \tag{3.6}$$

which is a 2-level orthogonal array of 4 trials. If the 2's in (3.6) are replaced with -1's, then the inner product of any two columns of the experimental matrix is null indicating that all columns of the orthogonal array are mutually orthogonal. Because the columns are orthogonal the effect of any factor across all trials is statistically independent from the effect of any other factor [10]. To illustrate, consider 2 progenitors α and β where $\boldsymbol{\alpha} = [\alpha_1 \quad \alpha_2 \quad \alpha_3]^T$ and $\boldsymbol{\beta} = [\beta_1 \quad \beta_2 \quad \beta_3]^T$. Using (3.6) to generate the orthogonal experimental matrix, the corresponding experimental trials and factors are represented by the rows and columns of Table 3.1 respectively.

Table 3.1: Two-Level Experimental Matrix for 3 Factors

Trial (n)	Factors		
1	α_1	α_2	α_3
2	α_1	β_2	β_3
3	β_1	α_2	β_3
4	β_1	β_2	α_3

The Latin square and the full factorial experiment space can be represented graph-

ically as depicted in Figure 3.4 where ϕ_x represent the independent factors and the vertices correspond to the space-representative trials. As illustrated in Figure 3.4,

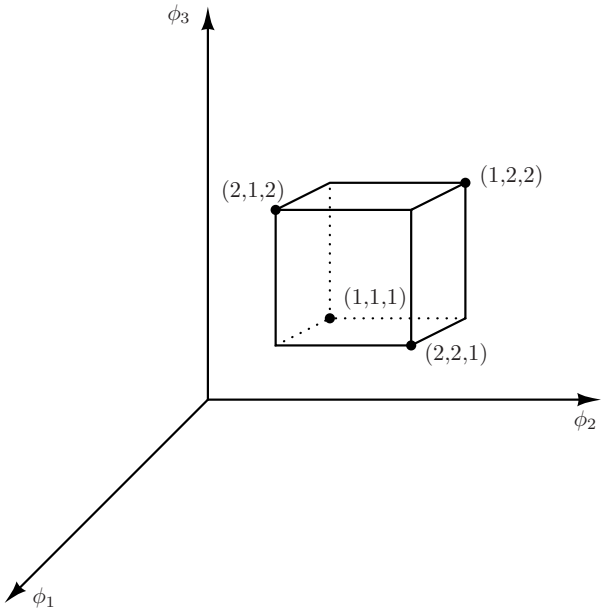

Figure 3.4: Graphical Representation of $L_4\left(2^3\right)$

the edges of the graph enclose the full factorial experiment space. On inspection, it is clear that the full factorial space of 8 experiments has been reduced to the evaluation of a subset of 4 vertices thereby decreasing the total number of trials necessary to observe the effects of the experimental factors by a factor of two.

3.2.5.2 Taguchi Method

To estimate an optimal set of alleles, or factors, from the reduced factorial experiment space, the Taguchi method determines the cost of each trial of the experimental matrix and processes these costs in a metric which is used to calculate the observed effects of all available alleles. Based on this calculation, a set of alleles are selected

from the available progenitors to produce an estimated optimal progeny.

This thesis uses the metric S_n which is defined as the cost function squared for the nth trial, that is,

$$S_n = J^2(\boldsymbol{x}_n) \tag{3.7}$$

where $\boldsymbol{x}_n$ is a vector containing the alleles, or factors, of the nth trial. This metric is then used to calculate each allele's observed effect which is subsequently used for determining an estimated optimal progeny.

The observed effect, $E_{x_k P_m}$, for an allele or factor x_k is defined as the sum of the metrics, S_n, for which the nth trial corresponds to the mth progenitor's factor contribution. For example, for the two-level experiment given in Table 3.1, the observed effect is given as

$$E_{x_k P_m} = \sum_{i \in \{k : x_k = P_m\}} S_i \tag{3.8}$$

where i corresponds to the trial number in which the mth progenitor P_m contributed its kth respective factor. For a global minimizer, the optimal contributing kth allele is defined as the allele from the progenitor which gives the lowest value for $E_{x_k P_m}$. To illustrate, consider an experiment with two progenitors which corresponds to a two-level orthogonal array. If $E_{x_k P_1} < E_{x_k P_2}$ for the factor x_k, the optimal contribution is given by progenitor 1 (P_1). Conversely, for $E_{x_k P_1} > E_{x_k P_2}$, progenitor 2 ($P_2$) would provide the optimal contribution [41]. The optimal chromosome is then assembled by taking the factor from the optimal progenitor for each of the K factors in the chromosome. This process is further illustrated in the following example.

3.2.5.3 Taguchi Method Example

To illustrate the Taguchi method, consider a cost function $J(\boldsymbol{x})$ given as

$$J(\boldsymbol{x}) = \sum_{i=1}^{7} x_i^2 \tag{3.9}$$

where $x_1, x_2, \ldots, x_7$ are real elements of the vector $\boldsymbol{x}$. Equation (3.9) describes an elliptic paraboloid that has a global minimum at zero. Consider two randomly selected progenitors, $\mathcal{C}_1$ and $\mathcal{C}_2$, where $\mathcal{C}_1 = [1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0]$ and $\mathcal{C}_2 = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1]$. Because the experiment has 7 factors, or alleles, and 2 progenitors, or levels, the orthogonal array, $L_8\left(2^7\right)$, where

$$L_8\left(2^7\right) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 2 & 1 & 1 & 2 & 2 \\ 1 & 2 & 2 & 2 & 2 & 1 & 1 \\ 2 & 1 & 2 & 1 & 2 & 1 & 2 \\ 2 & 1 & 2 & 2 & 1 & 2 & 1 \\ 2 & 2 & 1 & 1 & 2 & 2 & 1 \\ 2 & 2 & 1 & 2 & 1 & 1 & 2 \end{pmatrix} \tag{3.10}$$

can be used to form an appropriate experimental matrix. Using the orthogonal array shown in (3.10), the experimental matrix illustrated in Table 3.2 can be created where n denotes the trial number, x_k denotes the kth experimental factor, $\boldsymbol{x}_n$ represents the chromosome for the nth trial, $J(\boldsymbol{x}_n)$ corresponds to the evaluated cost for the nth experimental trial, and S_n corresponds to the calculated metric for the nth experimental trial.

Table 3.2 provides all the necessary information to calculate the observed effects, $E_{x_k P_m}$, for the K alleles which are available from the two progenitors P_1 and P_2. For

Table 3.2: Taguchi Method Example Experimental Matrix

Trial (n)	Chromosome	Experimental Factors							$J_n(\boldsymbol{x}_n)$	S_n
		x_1	x_2	x_3	x_4	x_5	x_6	x_7		
1	$\boldsymbol{x}_1$	1	1	1	1	0	0	0	4	16
2	$\boldsymbol{x}_2$	1	1	1	0	1	1	1	6	36
3	$\boldsymbol{x}_3$	1	0	0	1	0	1	1	4	16
4	$\boldsymbol{x}_4$	1	0	0	0	1	0	0	2	4
5	$\boldsymbol{x}_5$	0	1	0	1	1	0	1	4	16
6	$\boldsymbol{x}_6$	0	1	0	0	0	1	0	2	4
7	$\boldsymbol{x}_7$	0	0	1	1	1	1	0	4	16
8	$\boldsymbol{x}_8$	0	0	1	0	0	0	1	2	4

this example, the observed effects are calculated as follows:

$$E_{x_1P_1} = \sum_{i\in\{n:x_1=1\}=\{1,2,3,4\}} S_i = 72 \qquad E_{x_1P_2} = \sum_{i\in\{n:x_1=0\}=\{5,6,7,8\}} S_i = 40$$

$$E_{x_2P_1} = \sum_{i\in\{n:x_2=1\}=\{1,2,5,6\}} S_i = 72 \qquad E_{x_2P_2} = \sum_{i\in\{n:x_2=0\}=\{3,4,7,8\}} S_i = 40$$

$$E_{x_3P_1} = \sum_{i\in\{n:x_3=1\}=\{1,2,7,8\}} S_i = 72 \qquad E_{x_3P_2} = \sum_{i\in\{n:x_3=0\}=\{3,4,5,6\}} S_i = 40$$

$$E_{x_4P_1} = \sum_{i\in\{n:x_4=0\}=\{1,3,5,7\}} S_i = 64 \qquad E_{x_4P_2} = \sum_{i\in\{n:x_4=1\}=\{2,4,6,8\}} S_i = 48$$

$$E_{x_5P_1} = \sum_{i\in\{n:x_5=0\}=\{1,3,6,8\}} S_i = 40 \qquad E_{x_5P_2} = \sum_{i\in\{n:x_5=1\}=\{2,4,5,7\}} S_i = 72$$

$$E_{x_6P_1} = \sum_{i\in\{n:x_6=0\}=\{1,4,5,8\}} S_i = 40 \qquad E_{x_6P_2} = \sum_{i\in\{n:x_6=1\}=\{2,3,6,7\}} S_i = 72$$

$$E_{x_7P_1} = \sum_{i\in\{n:x_7=0\}=\{1,4,6,7\}} S_i = 40 \qquad E_{x_7P_2} = \sum_{i\in\{n:x_7=1\}=\{2,3,5,8\}} S_i = 72$$

The estimated optimal chromosome, $\mathcal{C}^*$, is given as

$$\mathcal{C}^* = [x_1^*, x_2^*, \ldots, x_k^*] \tag{3.11}$$

where x_k^* represents the optimal kth allele. For example, because $E_{x_1 P_2} < E_{x_1 P_1}$, the first optimal allele is given as $x_1^* = 0$ which corresponds to the first allele of the second progenitor (P_2). The remaining results are summarized in Table 3.3. Based on these results, the estimated optimal chromosome is given as $\mathcal{C}^* = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$. Evaluating $\mathcal{C}^*$ by (3.9) yields a result of zero verifying that the result is globally optimal.

Table 3.3: Taguchi Method Example Results

Description	**Expression**	**Experimental Factors**						
		x_1	x_2	x_3	x_4	x_5	x_6	x_7
Observed Effects	$E_{x_k P_1}$	72	72	72	64	40	40	40
	$E_{x_k P_2}$	40	40	40	48	72	72	72
Chromosome 1	$\mathcal{C}_1$	1	1	1	1	0	0	0
Chromosome 2	$\mathcal{C}_2$	0	0	0	0	1	1	1
Optimal Progenitor	–	$\mathcal{C}_2$	$\mathcal{C}_2$	$\mathcal{C}_2$	$\mathcal{C}_2$	$\mathcal{C}_1$	$\mathcal{C}_1$	$\mathcal{C}_1$
Optimal Progeny	$\mathcal{C}^*$	0	0	0	0	0	0	0

3.2.6 Single Point-Swap Mutation

Whether stochastic or guided via the Taguchi method, the process of transferring genetic information between members of any given population through crossover potentially evolves more fit individuals from the currently available genotype. The genetic composition of the more fit offspring is limited by the genetic diversity within the current generational genotype. If the optimal chromosome is to be evolved through crossover alone, the optimal genes must exist within the genotype. Because this cannot be guaranteed, a single-point mutation operator is implemented to introduce new genetic information into the otherwise static genotype. This process is illustrated in

Figure 3.5.

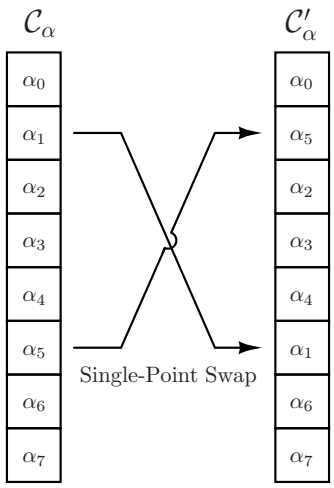

Figure 3.5: Single-Point Swap Mutation

Each individual is assigned a random number s that has a uniform distribution over $\Big[0, 1\Big]$. Similar to crossover, mutation is governed by a coefficient for probability of mutation, P_m, which typically ranges from 0.01 to 0.1. If $s < P_m$, mutation occurs and genetic diversity is introduced through single point-swap mutation. Conversely, if $s \geq P_m$, the individual is returned to the population unaltered. For this thesis, the best or elite chromosome is also eligible for mutation, thereby avoiding premature convergence and increasing the robust nature of the algorithm.

3.2.7 Convergence

Convergence is characterized by a stall in evolutionary progress which can be detected by a lack of improvement of objective function performance over some number of generations of the best chromosome. The stochastic introduction of new genetic information via the mutation operator ensures that there will always be some steady

state misadjustment. However, the elitist nature of the algorithm ensures that the best chromosome a member of the mating pool.

Because the selection operator favors the most fit individuals, the population will converge to the most fit individual as evolutionary stall persists. As such, another method for detecting convergence is to compare the population variance against a threshold value. Both methods have been used in the development of the HOG algorithm. However, it was observed that selecting a minimum of 1000 generations and a stall of 50 generations was sufficient for determining convergence in most cases which is consistent with results shown in [23].

3.2.8 HOG Algorithm Application Examples

To demonstrate the HOG algorithm, three different cost functions were selected which are considered canonical exercises for determining the robustness of optimization algorithms [29] [23] [41].

3.2.8.1 Example 1: MATLAB® Peaks Function

The MATLAB® *Peaks* function, denoted as $P(x, y)$, where

$$P(x,y) = 3\Big(1-x\Big)^2 e^{\left(-x^2-(y+1)^2\right)} - 10\Big(\frac{x}{5} - x^3 - y^5\Big)e^{\left(-x^2-y^2\right)} - \frac{1}{3}e^{\left(-(x+1)^2-y^2\right)} \quad (3.12)$$

and x and y are real continuous variables is a function that is obtained by translating and scaling Gaussian distributions. The *Peaks* function in (3.12) exhibits two local minima and one global minimum near $(0.25, -1.625)$ as illustrated in Figure 3.6.

3.2.8.1.1 Algorithm Setup and Initialization

To find the global minimum of the the *Peaks* function, the HOG algorithm was initially configured with a population size of 200 chromosomes. The nth chromosome was seeded with two alleles, x_n and y_n, where x_n and y_n are continuous random variables uniformly distributed over the feasible solution space,

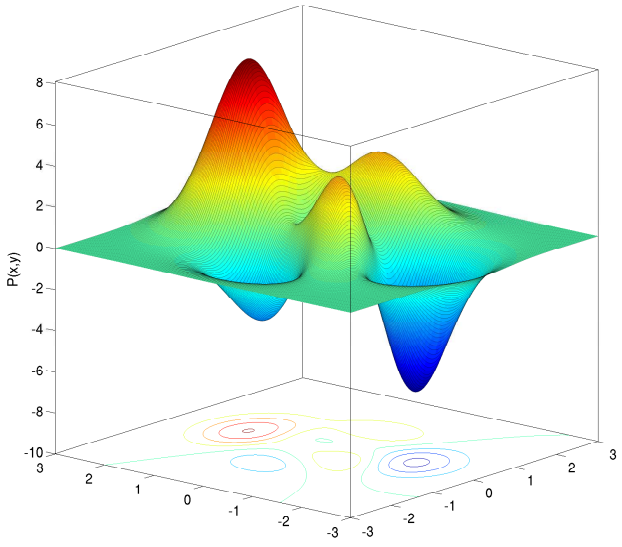

Figure 3.6: MATLAB® *Peaks* Function

$\{(x, y) : -3 \leq x \leq 3, -3 \leq y \leq 3\}$. To illustrate average performance metrics, the HOG algorithm was applied 50 times to minimizing the cost function given in (3.12). For each trial, convergence was defined as a lack of further improvement over 50 successive generations following a minimum of 1000 generations. The remaining HOG algorithm parameters are summarized in Table 3.4.

3.2.8.1.2 Results

In all 50 cases, the HOG algorithm successfully reached the global minimum. Figure 3.7 illustrates the average learning curve over all 50 runs of the HOG algorithm when applied to the *Peaks* function.

Table 3.4: MATLAB® *Peaks* Function Minimization: HOG Algorithm Parameters

Description	**Value**
Problem Dimension	2
Population Size	200×2
Feasible Solution Space	$-3 \leq \{x_n, y_n\} \leq 3$
Probability of Crossover	$P_c = 0.2$
Probability of Mutation	$P_m = 0.02$
Selection Pressure	$\eta^+ = 1.1$

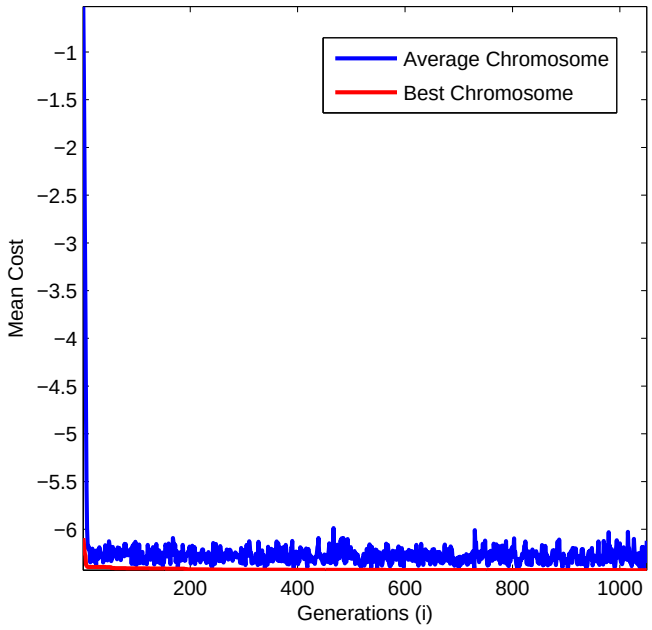

Figure 3.7: MATLAB® *Peaks* Function Minimization: Average Cost

Figure 3.8 shows a parametric plot of the *Peaks* function and plots the average chromosome for each generation of a population for one of the 50 cases. Figure 3.8 shows that the average chromosome converges to the optimum value located at the

global minimum of 6.5 at $(x, y) = (0.25, -1.625)$.

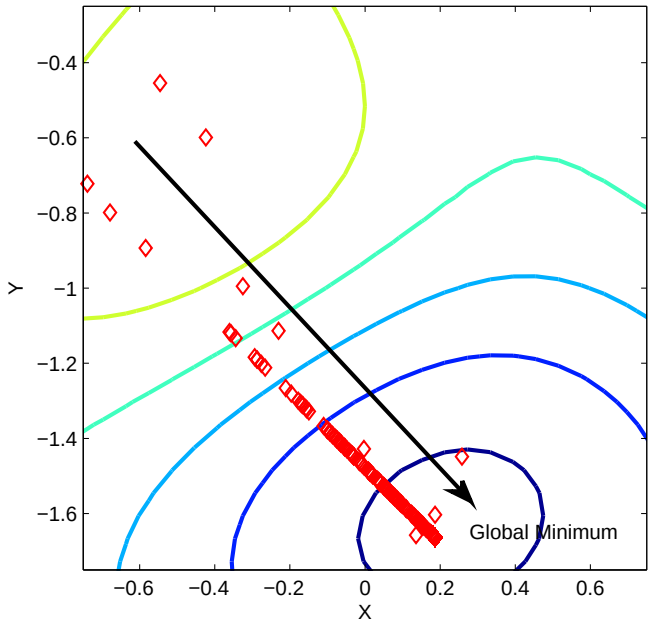

Figure 3.8: MATLAB® *Peaks* Function Minimization: Average Chromosome Parametric Plot

3.2.8.2 Example 2: Rastrigin Function

The Rastrigin function is a highly multimodal nonlinear function that is often used to test global optimizers and specifically genetic algorithms. The Rastrigin function, $f(\boldsymbol{x})$ is defined as

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \Big[x_i^2 - 10\cos\big(2\pi x_i\big) + 10 \Big] \tag{3.13}$$

and the two dimensional case is plotted in Figure 3.9. As illustrated in Figure 3.9, the two dimensional Rastrigin function is characterized by its numerous local minima

and a global minimum at $(0, 0)$.

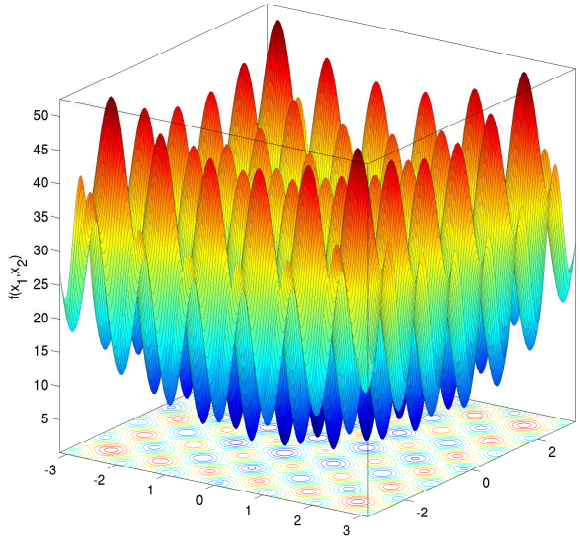

Figure 3.9: Two-Dimensional Rastrigin Function

3.2.8.2.1 Algorithm Setup and Initialization

To find the global minimum of the the Rastrigin function, the HOG algorithm was configured with a population size of 200 chromosomes. Each chromosome was seeded with two alleles, x_{1n} and x_{2n}, where x_{1n} and x_{2n} are continuous random variables which are uniformly distributed over the feasible solution space, $\{(x_1, x_2) : -5.12 \leq x_1 \leq 5.12, -5.12 \leq x_2 \leq 5.12\}$. To illustrate average performance metrics, the HOG algorithm was applied 50 times to minimizing the cost function given in (3.13). For each trial, convergence was defined as a lack of further improvement over 50 successive generations following a minimum of 1000 generations. The

remaining HOG algorithm parameters are summarized in Table 3.5.

Table 3.5: Rastrigin Function Minimization: HOG Algorithm Parameters

Description	**Value**
Problem Dimension	2
Population Size	200×2
Feasible Solution Space	$-5.12 \leq \{x_n, y_n\} \leq 5.12$
Probability of Crossover	$P_c = 0.2$
Probability of Mutation	$P_m = 0.02$
Selection Pressure	$\eta^+ = 1.1$

3.2.8.2.2 Results

In all 50 cases, the HOG algorithm successfully reached the global minimum. Figure 3.10 illustrates the average learning curve over all 50 runs of the HOG algorithm when applied to the Rastrigin function.

Figure 3.11 shows a parametric plot of the Rastrigin function and plots the average chromosome for each generation of a population for one of the 50 cases. As indicated by the arrows, the algorithm initially finds a local minima adjacent to the global minimum. However, the average chromosome converges to the global optimum value located at $(0, 0)$.

3.2.8.3 Example 3: Rosenbrock Function

The Rosenbrock function is a uni-modal non-linear function often used to evaluate the performance of optimization algorithms. The Rosenbrock function, $f(\boldsymbol{x})$, is defined as

$$f(\boldsymbol{x}) = \sum_{i=1}^{N-1} \Big[\Big(1 - x_i\Big)^2 + 100 \Big(x_{i+1} - x_i^2\Big)^2 \Big] \tag{3.14}$$

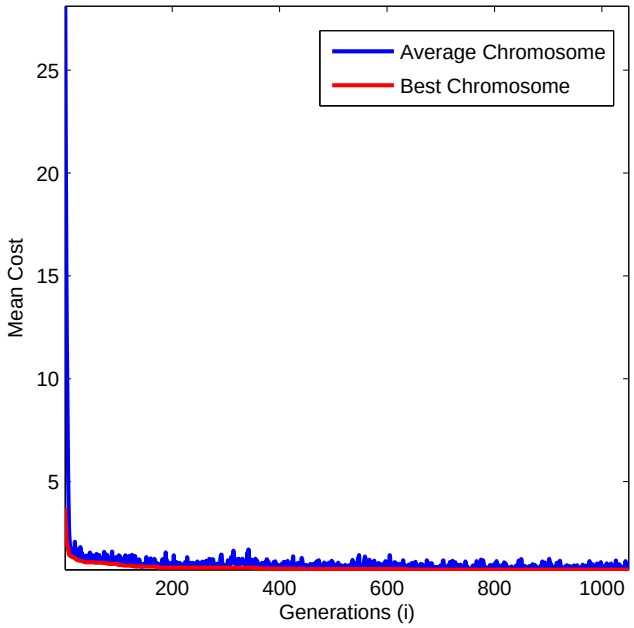

Figure 3.10: Rastrigin Function Minimization: Average Cost

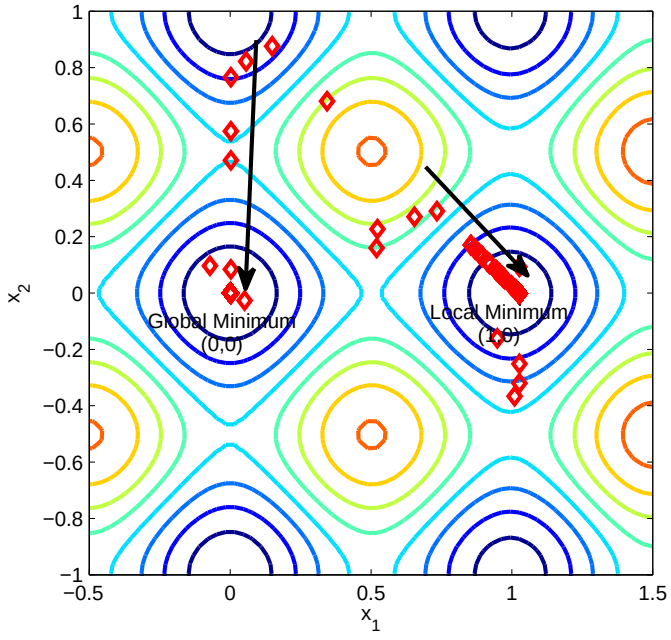

Figure 3.11: Rastrigin Function Minimization: Average Chromosome Parametric Plot

and the two dimensional case is plotted in Figure 3.12. As illustrated in Figure 3.12, the two dimensional Rosenbrock function is characterized by its long banana shaped valley with a global minimum at $(1, 1)$.

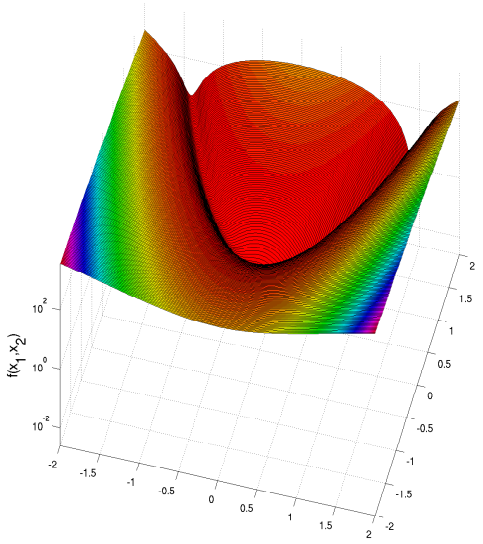

Figure 3.12: Two-Dimensional Rosenbrock Function

3.2.8.3.1 Algorithm Setup and Initialization

To find the global minimum of the the Rosenbrock function, the HOG algorithm was configured with a population size of 200 chromosomes. The nth chromosome was seeded with two alleles, x_{1n} and x_{2n}, where x_{1n} and x_{2n} are continuous random variables which are uniformly distributed over the feasible solution space, $\{(x_1, x_2) : -\pi \leq x_1 \leq \pi, -\pi \leq x_2 \leq \pi\}$. To illustrate average performance metrics, the HOG algorithm was applied 50 times to minimizing the cost function given in (3.14). For each trial, convergence was defined as a lack of further improvement over

50 successive generations following a minimum of 1000 generations. The remaining HOG algorithm parameters are summarized in Table 3.6.

Table 3.6: Rosenbrock Function Minimization: HOG Algorithm Parameters

Description	**Value**
Problem Dimension	2
Population Size	200×2
Feasible Solution Space	$-\pi \leq \{x_n, y_n\} \leq \pi$
Probability of Crossover	$P_c = 0.2$
Probability of Mutation	$P_m = 0.02$
Selection Pressure	$\eta^+ = 1.1$

3.2.8.3.2 Results

In all 50 cases, the HOG algorithm successfully reached the global minimum. Figure 3.13 illustrates the average learning curve over all 50 runs of the HOG algorithm when applied to the Rosenbrock function.

Figure 3.14 shows a parametric plot of the Rosenbrock function and plots the average chromosome for each generation of a population for one of the 50 cases. Figure 3.14 shows that the average chromosome converges to the optimum value located at the global minimum $(1, 1)$.

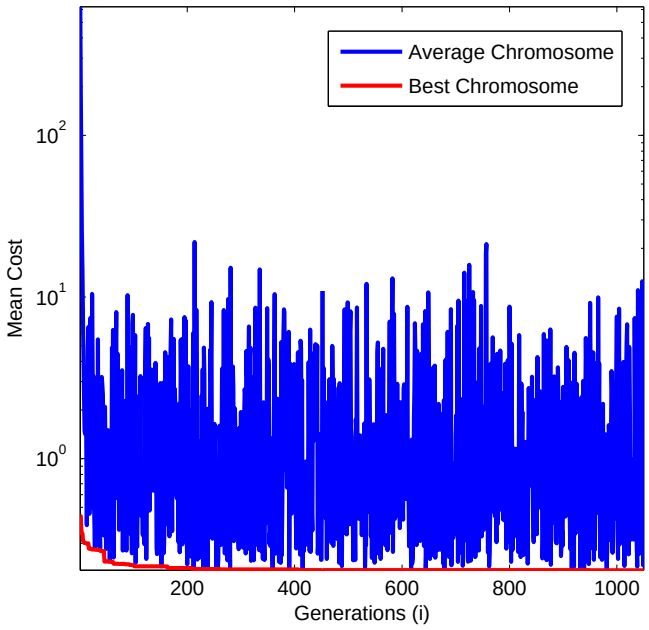

Figure 3.13: Rosenbrock Function Minimization: Average Cost

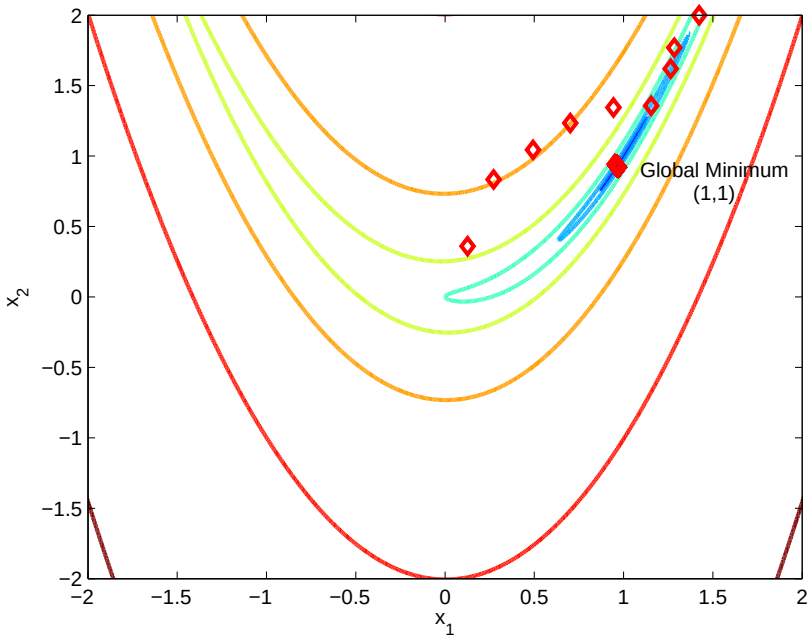

Figure 3.14: Rosenbrock Function Minimization: Average Chromosome Parametric Plot

CHAPTER 4

IMPLEMENTATION AND RESULTS

As discussed in Chapter 2, $\Delta\Sigma$ modulators use oversampling and feedback to attenuate inband quantization noise power. Because many discrete-time $\Delta\Sigma$ modulators can be modeled by the LTI system shown in Figure 4.1, the NTF and STF can be described by (2.34) and (2.33). As such, the NTF and STF represent discrete

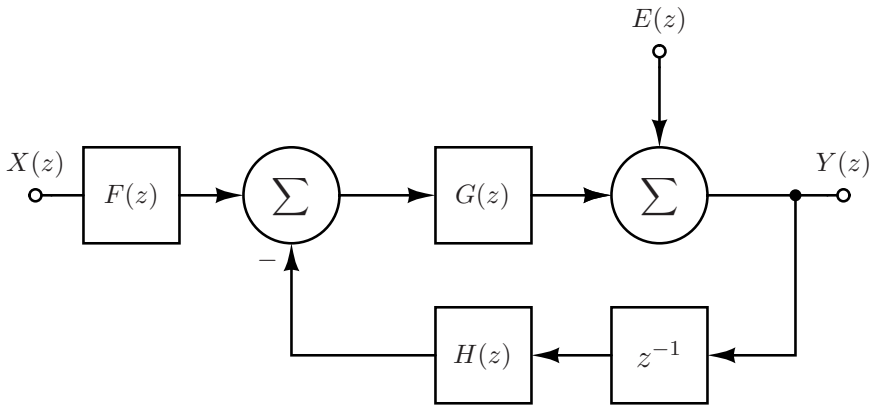

Figure 4.1: $\Delta\Sigma$ Modulator Linear Model Block Diagram

recursive filters, which are often designed using digital IIR filter design techniques. The structure of a $\Delta\Sigma$ modulator dictates that the STF and NTF share a common set of poles. As such, the design of either the STF or NTF has implications for the design of the other. Because a $\Delta\Sigma$ modulator's performance depends mostly on the

shape of the inband NTF, the NTF is typically designed first. The zeros of the STF are then designed to accommodate the poles that result from the preceding NTF design.

NTFs are often designed using traditional filters, such as Butterworth and Chebyshev filters [42]. These methods result in optimal designs for certain performance criterion, often without consideration for others. For example, a $\Delta\Sigma$ modulator's NTF that has been designed using a Chebyshev filter is optimal for dynamic range performance but not necessarily optimal for SNR performance. Traditional filters such as Chebyshev filters generate NTFs with well defined passbands which are not required for most NTFs. Also, these methods are not always easily adaptable to the frequency response specifications required by $\Delta\Sigma$ modulators for certain applications [36].

Other contemporary methods often use numerical methods to determine optimal NTFs. For example, the Delta Sigma Toolbox available for MATLAB® generates $\Delta\Sigma$ modulator NTFs by determining NTF zeroes which minimize the inband quantization noise power [35]. This technique determines these optimized zeroes by setting the first derivative of the inband power spectral density to zero and solving the resulting equations. After determining the optimized zeroes, the NTF's poles are optimized using an iterative approach. Because poles and zeros do not affect system functions independently and because this technique determines the NTF's zeros independently from its poles, this technique does not necessarily generate optimal designs. As such, it will be shown that this method offers only marginal improvement with respect to SNR over traditional polynomial based methods.

In this thesis, the Hybrid Orthogonal Genetic (HOG) algorithm, developed in Chapter 3, is used to determine optimal NTFs and STFs. The overall design methodology is illustrated in Figure 4.2. Specifically, this implementation of the HOG algo-

rithm evolves optimal NTFs by maximizing a weighted combination of inband SNR and DR. After determining an optimal NTF, the HOG algorithm then evolves an optimal STF by minimizing the passband magnitude error and attenuating out of band signal energy. The resulting NTFs and STFs are then rigorously simulated and analyzed to ensure stability and performance [37].

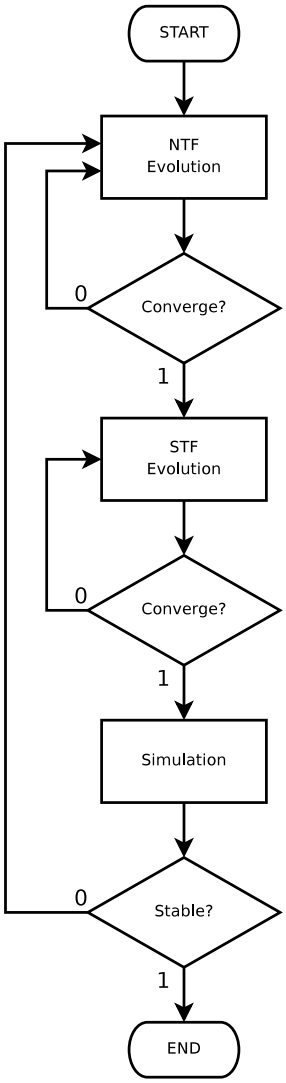

Figure 4.2: Optimal $\Delta\Sigma$ Modulator Design Flowchart

4.1 $\Delta\Sigma$ Modulator Design Objective Functions

For this thesis, optimal NTFs and STFs are determined by approximating the NTF and STF system functions so that the $\Delta\Sigma$ modulator's SNR and DR are maximized. That is, cost functions are minimized with respect to the NTF and STF coefficients such that the effective resolution of the $\Delta\Sigma$ modulator is maximized.

4.1.1 NTF Objective Function

Ideally, a NTF magnitude response, $|H(e^{j\omega})|$, would remove all signal energy in the stopband which implies that

$$|H(e^{j\omega})| = 0, \qquad \omega \in \omega_{\text{sb}} \tag{4.1}$$

where ω_{sb} is the set of stopband frequencies. For a lowpass $\Delta\Sigma$ modulator, ω_{sb}, can be defined such that $\omega_{\text{sb}} \in \{\omega : |\omega| \leq \omega_s\}$ for the stopband corner frequency, ω_s. However, to prevent the NTF from having all zero coefficients the NTF's passband is ideally 1. Therefore, the ideal NTF magnitude response can be written as

$$|H(e^{j\omega})| = \begin{cases} 0, & \omega \in \omega_{\text{sb}} \\ 1, & \omega \in \omega_{\text{pb}} \end{cases} \tag{4.2}$$

where ω_{pb} is the set of passband frequencies. For a lowpass $\Delta\Sigma$ modulator, ω_{pb} can be defined such that $\omega_{\text{pb}} \in \{\omega : \omega_p \leq |\omega| \leq \pi\}$ for the passband corner frequency, ω_p.

In practice, the ideal frequency response in (4.2) cannot be achieved, and therefore, $|H(e^{j\omega})|$ must be approximated. As such, the difference between the ideal magnitude response and the realized NTF magnitude response, $|\text{NTF}(e^{j\omega})|$, is referred to as the NTF magnitude response error, $|H_e(e^{j\omega})|$; that is,

$$|H_e(e^{j\omega})| = |H(e^{j\omega})| - |\text{NTF}(e^{j\omega})|. \tag{4.3}$$

Substituting (4.2) into (4.3), it can be seen that for a lowpass $\Delta\Sigma$ modulator

$$|H_e(e^{j\omega})| = \begin{cases} |\mathrm{NTF}(e^{j\omega})|, & \omega \in \omega_{\mathrm{sb}} \\ 1 - |\mathrm{NTF}(e^{j\omega})|, & \omega \in \omega_{\mathrm{pb}} \end{cases}. \tag{4.4}$$

4.1.1.1 SNR Optimization

To maximize the SNR over the NTF's stopband, the noise power in the NTF's stopband must be minimized. As demonstrated by (2.75) in Chapter 2, a $\Delta\Sigma$ modulator's output quantization noise power, P_q, over the stopband can be written as

$$P_q = \frac{1}{2\pi} \int_{\omega \in \omega_{\mathrm{sb}}} |\mathrm{NTF}(e^{j\omega})|^2 \boldsymbol{S}_e(e^{j\omega}) d\omega \tag{4.5}$$

where $\boldsymbol{S}_e(e^{j\omega})$ is the power spectral density of the quantization noise. As shown in (2.79),

$$\boldsymbol{S}_{e(n)}(e^{j\omega}) = \frac{\Delta^2}{12}$$

which implies that

$$P_q = \frac{\Delta^2}{12} \left(\frac{1}{2\pi}\right) \int_{\omega \in \omega_{\ \mathrm{sb}}} |\mathrm{NTF}(e^{j\omega})|^2 d\omega. \tag{4.6}$$

Because the p-norm, denoted $\|\cdot\|_p$, of a continuous-time signal, $x(t)$, is defined as

$$\|x(t)\|_p = \left(\int_{t \in T} |x(t)|^p dt \right)^{\frac{1}{p}} \tag{4.7}$$

where T denotes some fixed time interval, P_q can be written as

$$P_q = \frac{\Delta^2}{12} \left(\frac{1}{2\pi}\right) \int_{\omega \in \omega_{\ \mathrm{sb}}} |\mathrm{NTF}(e^{j\omega})|^2 d\omega = \frac{\Delta^2}{12} \left(\frac{1}{2\pi}\right) \|\mathrm{NTF}(e^{j\omega})\|_{2\ \omega \in \omega_{\mathrm{sb}}}^2. \tag{4.8}$$

Therefore, a $\Delta\Sigma$ modulator's SNR can be maximized by minimizing $\|\mathrm{NTF}(e^{j\omega})\|_{2\ \omega \in \omega_{\mathrm{sb}}}^2$.

4.1.1.2 DR Optimization

Recall that DR is defined as the ratio of the maximum to the minimum detectable signal levels. As such, the DR can be maximized my minimizing the largest noise component over the stopband of the NTF. Because the ∞-norm of a continuous-time signal, $x(t)$, can be written as

$$\|x(t)\|_\infty = \lim_{p\to\infty}\left(\int_{t\in T}|x(t)|^p dt\right)^{\frac{1}{p}} = \max\langle|x(t)|\rangle. \tag{4.9}$$

Therefore, a $\Delta\Sigma$ modulator's DR can be maximized by minimizing $\|\mathrm{NTF}(e^{j\omega})\|_{\omega\in\omega_{\mathrm{sb}}}^{\infty}$.

4.1.1.3 Passband Optimization

It has been observed that the likelihood of $\Delta\Sigma$ modulator stability can be increased by designing NTFs that have approximately unity gain at π radians/sample and that do not have passband peaking [34]. As such, minimizing the 1-norm of the magnitude error over the passband, $\|1 - |\,\mathrm{NTF}(e^{j\omega})|\|_{\omega\in\omega_{\mathrm{pb}}}^{1}$, can produce passband magnitude responses with approximately unity gain at π radians/sample and that do not have passband peaking.

4.1.1.4 Stability

Because the NTF is modeled as discrete-time LTI systems, its pole locations, $\{\gamma_k\}$, must lie within the unit circle to realize a stable system; that is, for the $\Delta\Sigma$ modulator to be stable,

$$|\gamma_k| < 1. \tag{4.10}$$

As such, the solution space for the approximated NTF must be constrained such that none of its poles lie outside the unit circle.

4.1.1.5 Cost Function

The NTF cost function is the weighted sum of the approximated magnitude response errors and can be written as

$$J_{\mathrm{NTF}} = \alpha \|\mathrm{NTF}(e^{j\omega})\|_{2,\,\omega\in\omega_{\mathrm{sb}}}^{2} + \beta \|\mathrm{NTF}(e^{j\omega})\|_{\infty,\,\omega\in\omega_{\mathrm{sb}}} + (1-\alpha-\beta)\,\|1-|\mathrm{NTF}(e^{j\omega})|\|_{1,\,\omega\in\omega_{\mathrm{pb}}} \tag{4.11}$$

where α and β are weighting coefficients such that $\{\alpha,\beta\} \geq 0$ and $\alpha+\beta \leq 1$, $\|\cdot\|_p$ denotes the p-norm, and all the pole locations are constrained such that they lie within the unit circle.

As discussed in Chapter 2, NTFs are typically modeled as LTI systems that can be described by rational functions. Because minimizing the p-norm of a rational function is a difficult analytical problem, the cost function is minimized numerically. Thus, the NTF is approximated by calculating the DFT over the stopband and the passband.

For example, the N_s-point DFT of the NTF can be written as

$$\mathrm{NTF}(k) = \mathrm{NTF}(e^{j\omega})\Big|_{\omega=\frac{2\pi}{N_s}k} \tag{4.12}$$

for $k \in \{I : 0 \leq k \leq N_s - 1\}$. Thus,

$$\|\mathrm{NTF}(e^{j\omega})\|_{2,\,\omega\in\omega_{\mathrm{sb}}}^{2} \approx \|\,\mathrm{NTF}(k)\|_{2,\,k\in k_{\mathrm{sb}}}^{2} \tag{4.13}$$

where $k_{\mathrm{sb}} \in \{k : 0 \leq k \leq N_s - 1 \text{ and } \frac{2\pi}{N_s}k \in \omega_{\mathrm{sb}}\}$ and where the p-norm of a discrete signal, $x(n)$, is defined as

$$\|x(n)\|_p = \left(\sum_{k=1}^{N} |x(k)|^p\right)^{\frac{1}{p}} \tag{4.14}$$

for a signal of length N. As such, the SNR can be maximized by minimizing the 2-norm squared of the approximated stopband frequency response, $\|\mathrm{NTF}(k)\|_{2,\,k\in k_{\mathrm{sb}}}^{2}$,

where $k_{\text{sb}} = \left\{k : \frac{2\pi}{N_s}k \in \omega_{\text{sb}}\right\}$. It can also be seen that

$$\|\text{NTF}(e^{j\omega})\|_{\underset{\omega\in\omega_{\text{sb}}}{\infty}} \approx \quad \|\,\text{NTF}(k)\|_{\underset{k\in k_{\text{sb}}}{\infty}} \tag{4.15}$$

where the ∞-norm of a discrete signal, $x(n)$, is defined as

$$\|x(n)\|_\infty = \lim_{p\to\infty}\left(\sum_{k=1}^{N}|x(k)|^p\right)^{\frac{1}{p}} = \max\langle|x(k)|\rangle. \tag{4.16}$$

for a signal of length N. As such, the DR can be maximized by minimizing the ∞-norm of the approximated stopband frequency response, $\|\text{NTF}(k)\|_{\underset{k\in k_{\text{sb}}}{\infty}}$, for

$$k_{\text{sb}} = \left\{k : \frac{2\pi}{N_s}k \in \omega_{\text{sb}}\right\}.$$

Similarly, the N_p-point DFT of the NTF can be written as

$$\text{NTF}(k) = \text{NTF}(e^{j\omega})\Big|_{\omega=\frac{2\pi}{N_p}k} \tag{4.17}$$

for $k \in \{I : 0 \le k \le N_p - 1\}$. Thus,

$$\|\text{NTF}(e^{j\omega})\|_{\underset{\omega\in\omega_{\text{pb}}}{1}} \approx \quad \|\,\text{NTF}(k)\|_{\underset{k\in k_{\text{pb}}}{1}} \tag{4.18}$$

where $k_{\text{pb}} = \{k : 0 \le k \le N_p - 1 \text{ and } \frac{2\pi}{N_p}k \in \omega_{\text{pb}}\}$. As such, the shape of the passband magnitude response can be determined by minimizing the 1-norm of the approximated passband magnitude error response, $\|1 - |\text{NTF}(k)|\|_{\underset{k\in k_{\text{pb}}}{1}}$, for $k_{\text{pb}} = \left\{k : \frac{2\pi}{N_p}k \in \omega_{\text{pb}}\right\}$.

Therefore, the NTF's objective function, J_{NTF}, in (4.11) can be approximated as

$$J_{\text{NTF}} = \alpha\|\text{NTF}(k)\|^2_{\underset{k\in k_{\text{sb}}}{2}} + \beta\|\text{NTF}(k)\|_{\underset{k\in k_{\text{sb}}}{\infty}} + (1-\alpha-\beta)\|1 - |\text{NTF}(k)|\|_{\underset{k\in k_{\text{pb}}}{1}} \tag{4.19}$$

where $\{\alpha, \beta\} \geq 0$ and $\alpha + \beta \leq 1$ and all the pole locations are constrained such that they lie within the unit circle. For this thesis, optimal NTFs were designed with $\alpha = \beta = 0.25$.

For example, the cost function illustrated in Figure 4.3 can be minimized to determine a highpass NTF for a lowpass $\Delta\Sigma$ modulator which is optimal with respect to a weighted combination of SNR and DR.

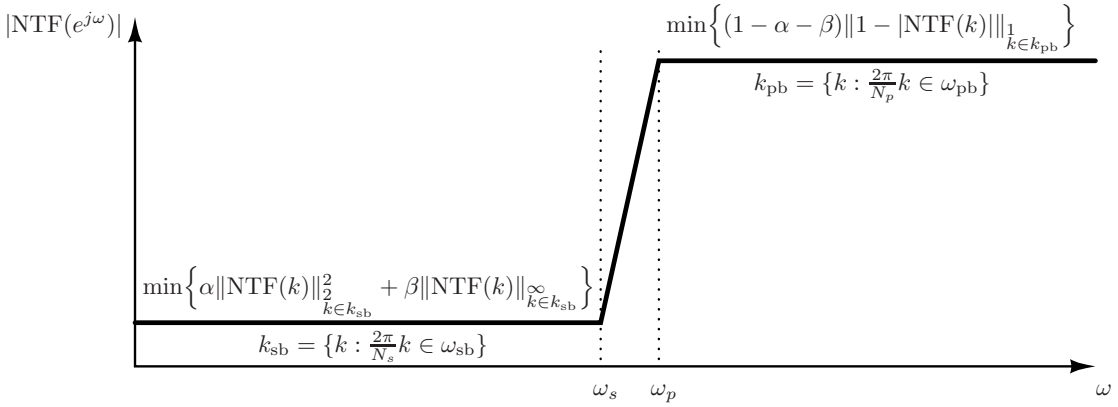

Figure 4.3: NTF Magnitude Response Objective Function

4.1.2 STF Objective Function

Ideally, a STF magnitude response, $|H(e^{j\omega})|$, would remove all signal energy in the stopband and the passband would be one; that is, an ideal STF magnitude response, $|H(e^{j\omega})|$, can be written as

$$|H(e^{j\omega})| = \begin{cases} 0, & \omega \in \omega_{\mathrm{sb}} \\ 1, & \omega \in \omega_{\mathrm{pb}} \end{cases} \tag{4.20}$$

where ω_{sb} is the set of stopband frequencies and ω_{pb} is the set of passband frequencies. In practice, the ideal frequency response in (4.20) cannot be achieved, and therefore, $|H(e^{j\omega})|$ must be approximated. As such, the difference between the ideal magnitude response and the realized STF magnitude response, $|\text{STF}(e^{j\omega})|$, is referred to as the STF magnitude response error, $H_e(e^{j\omega})$; that is,

$$H_e(e^{j\omega}) = |H(e^{j\omega})| - |\text{STF}(e^{j\omega})|. \tag{4.21}$$

Comparing (4.20) and (4.21), it can be seen that for a lowpass $\Delta\Sigma$ modulator

$$|H_e(e^{j\omega})| = \begin{cases} |\text{STF}(e^{j\omega})|, & \omega \in \omega_{\text{sb}} \\ 1 - |\text{STF}(e^{j\omega})|, & \omega \in \omega_{\text{pb}} \end{cases} \tag{4.22}$$

4.1.2.1 STF Optimization

Recall that the stopband signal energy can be minimized by minimizing the 2-norm squared of the frequency response, $\|\text{STF}(e^{j\omega})\|_2^2$, for $\omega \in \omega_{\text{sb}}$. Similarly, minimizing the 1-norm of the passband magnitude error, $\|1 - |\text{STF}(e^{j\omega})|\|_1$, for $\omega \in \omega_{\text{pb}}$ produces magnitude responses which are approximately unity gain and without passband peaking.

4.1.2.2 Cost Function

Therefore, the STF cost function, which is the weighted sum of the approximated magnitude response errors, can be written as

$$J_{\text{STF}} = \zeta \underset{\omega\in\omega_{\text{sb}}}{\|\text{STF}(e^{j\omega})\|_2^2} + (1-\zeta) \underset{\omega\in\omega_{\text{pb}}}{\|1 - |\text{STF}(e^{j\omega})|\|_1} \tag{4.23}$$

where ζ is a weighting coefficients such that $0 \leq \zeta \leq 1$ and $\|\cdot\|_p$ denotes the p-norm.

However, as discussed in Chapter 2, STFs are typically modeled as LTI systems

that can be described by rational functions. Because minimizing the p-norm of a rational function is a difficult analytical problem, the cost function is minimized numerically. Thus, the STF is approximated by calculating the DFT over the stopband and the passband.

As was done for the NTF, an N_s-point DFT and a N_p-point DFT can be used to approximate the cost function given in (4.23) such that

$$J_{\mathrm{STF}} = \zeta \|\mathrm{STF}(k)\|_{2}^{2}{}_{k\in k_{\mathrm{sb}}} + (1-\zeta)\|1 - |\mathrm{STF}(k)|\|_{1}{}_{k\in k_{\mathrm{pb}}} \tag{4.24}$$

where $0 \leq \zeta \leq 1$. For this thesis, optimal STFs were designed with a weighting coefficient of $\zeta = 0.5$.

For example, the cost function illustrated in Figure 4.4 can be minimized to determine an optimal STF for a lowpass $\Delta\Sigma$ modulator.

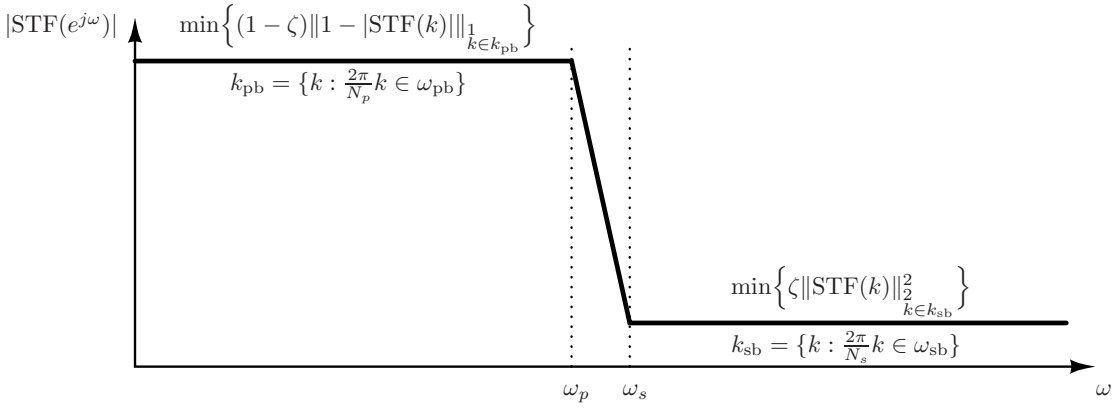

Figure 4.4: STF Magnitude Response Objective Function

4.1.3 Objective Function Minimization

The cost functions shown in (4.19) and (4.24) are known to be highly multimodal and non-differentiable equations [30]. As such, the HOG algorithm presented in Chapter 3 has been implemented as a constrained global optimizer to minimize the cost functions thereby producing $\Delta\Sigma$ modulator system functions which are optimal with respect to their cost functions.

The HOG algorithm constrains the solution space through penalization. For example, a large penalty value is returned if any of the evolved poles fall outside the unit circle. In addition, the general shape of the magnitude response is constrained such that proposed filter solutions which do not conform to the desired magnitude response shape (e.g. highpass or lowpass) are penalized. For example, the NTF magnitude response evaluated at DC must be sufficiently low or a large penalty value is returned. Similarly, the STF magnitude response evaluated at π radians/sample must be sufficiently low or a large penalty value is returned. In general, penalties are selected so that they do not restrict the solution space any more than necessary which has been shown to result in sub-optimal solutions [18].

4.2 HOG Algorithm Implementation

For this thesis, the HOG algorithm, developed in Chapter 3, is used to evolve a population of chromosomes which contain the polynomial coefficients for the rational functions which represent the NTF and STF. Once the population has been initialized to the regions surrounding the solutions of traditional polynomials (e.g the Chebyshev polynomial), the HOG algorithm iteratively evolves better polynomial coefficients until the population converges to the optimum. For this implementation, the optimal polynomial coefficients are determined by minimizing the cost functions in (4.19) and (4.24) thereby producing system functions which are optimal with respect to SNR

and DR.

4.2.1 Chromosome and Population Structure

Recall that structurally, chromosomes are implemented as arrays of length K. As such, these arrays are populated with K alleles, that are iteratively modified during the evolutionary process. For filter design applications, the alleles can correspond to the polynomial coefficients of the NTF or STF which have the general form

$$H(z) = \frac{\sum_{n=0}^{N} a_n z^{-n}}{\sum_{n=0}^{N} b_n z^{-n}} \tag{4.25}$$

where $\{a_n\}$ and $\{b_n\}$ are the sets of real polynomial coefficients and N is the order of the filter. However, populating the chromosomes directly with the polynomial coefficients, $\{a_n\}$ and $\{b_n\}$, yields chromosomes which are very sensitive to perturbation during the evolutionary process. To reduce coefficient sensitivity, (4.25) can be written in its zero-pole-gain form,

$$H(z) = \psi \cdot \frac{\prod_{n=1}^{N} \left(1 - c_n z^{-1}\right)}{\prod_{n=1}^{N} \left(1 - d_n z^{-1}\right)} \tag{4.26}$$

where $\{c_n\}$ is the set containing the system function's zeros, $\{d_n\}$ is the set containing the system function's poles, and ψ represents the system function's gain. The alleles then correspond to the system function's zeros, poles, and gain. However, for filters with real coefficients, these algorithms must explicitly manage complex conjugate pairs of poles and zeros which can significantly lengthen run times. To reduce coefficient sensitivity and simplify the management of complex conjugate pairs of poles and zeros, the chromosomes are structured specifically so that the system function,

$H(z)$, has the form

$$H(z) = \psi\left(\frac{1 - a_1 z^{-1}}{1 - b_1 z^{-1}}\right)^m \prod_{n=1}^{M} \frac{\left(1 + c_{1n} z^{-1} + c_{2n} z^{-2}\right)}{\left(1 + d_{1n} z^{-1} + d_{2n} z^{-2}\right)} \tag{4.27}$$

where M is the number of second-order-sections, ψ represents the system function's gain, and m is 1 or 0 for odd or even ordered systems, respectively [38].

For this thesis, the NTF is equivalent to (4.27); that is

$$\mathrm{NTF}(z) = \psi_N\left(\frac{1 - a_1 z^{-1}}{1 - b_1 z^{-1}}\right)^m \prod_{n=1}^{M} \frac{\left(1 + c_{1n} z^{-1} + c_{2n} z^{-2}\right)}{\left(1 + d_{1n} z^{-1} + d_{2n} z^{-2}\right)}. \tag{4.28}$$

Thus, for a NTF described by (4.28), the structural chromosome array, $\mathcal{C}_{\mathrm{NTF}}$, of length, K_N, where

$$K_N = 4M + 2m + 1, \tag{4.29}$$

can be written as

$$\mathcal{C}_{\mathrm{NTF}} = [c_{11}, c_{12}, d_{11}, d_{12}, c_{21}, c_{22}, d_{21}, d_{22}, \ldots, c_{M1}, c_{M2}, d_{M1}, d_{M2}, \psi_N]^T \tag{4.30}$$

for even ordered systems and

$$\mathcal{C}_{\mathrm{NTF}} = [a_1, b_1, c_{11}, c_{12}, d_{11}, d_{12}, c_{21}, c_{22}, d_{21}, d_{22}, \ldots, c_{M1}, c_{M2}, d_{M1}, d_{M2}, \psi_N]^T \tag{4.31}$$

for odd ordered systems where the superscript T denotes the transpose operator. As such, for a NTF of order $2M + 1$, a population, $\mathbb{G}_{\mathrm{NTF}}$, of n chromosomes can be

written as

$$\mathbb{G}_{\text{NTF}} = \Big[\mathcal{C}_{\text{NTF},1} \vdots \mathcal{C}_{\text{NTF},2} \vdots \cdots \vdots \mathcal{C}_{\text{NTF},n}\Big] = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n-1} & a_{1,n} \\ b_{1,1} & b_{1,2} & \dots & b_{1,n-1} & b_{1,n} \\ c_{11,1} & c_{11,2} & \dots & c_{11,n-1} & c_{11,n} \\ c_{12,1} & c_{12,2} & \dots & c_{12,n-1} & c_{12,n} \\ d_{11,1} & d_{11,2} & \dots & d_{11,n-1} & d_{11,n} \\ d_{12,1} & d_{12,2} & \dots & d_{12,n-1} & d_{12,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{M1,1} & c_{M1,2} & \dots & c_{M1,n-1} & c_{M1,n} \\ c_{M2,1} & c_{M2,2} & \dots & c_{M2,n-1} & c_{M2,n} \\ d_{M1,1} & d_{M1,2} & \dots & d_{M1,n-1} & d_{M1,n} \\ d_{M2,1} & d_{M2,2} & \dots & d_{M2,n-1} & d_{M2,n} \\ \psi_{N,1} & \psi_{N,2} & \dots & \psi_{N,n-1} & \psi_{N,n} \end{pmatrix} \tag{4.32}$$

where $\mathbb{G}_{\text{NTF}}$ is a $K_N \times n$ matrix and $\mathcal{C}_{\text{NTF},n}$ is the nth chromosome.

Because NTFs and STFs share a common set of poles and the NTF is designed first, the STF system function can be written as

$$\text{STF}(z) = \psi_S \left(\frac{1-\rho_1 z^{-1}}{1-b_1 z^{-1}}\right)^m \prod_{n=1}^{M} \frac{\left(1+\nu_{1n} z^{-1}+\nu_{2n} z^{-2}\right)}{\left(1+d_{1n} z^{-1}+d_{2n} z^{-2}\right)}. \tag{4.33}$$

Because the pole locations are established by the design of the NTF, only the STF's zero locations are perturbated during the optimization process. As such, for a STF system function described by (4.33), the structural chromosome array, $\mathcal{C}_{\text{STF}}$, of length K_S, where

$$K_S = 2M + m + 1, \tag{4.34}$$

can be written as

$$\mathcal{C}_{\text{STF}} = [\nu_{11}, \nu_{12}, \nu_{21}, \nu_{22}, \ldots, \nu_{M1}, \nu_{M2}, \psi_S]^T \tag{4.35}$$

for even ordered systems and

$$\mathcal{C}_{\text{STF}} = [\rho_1, \nu_{11}, \nu_{12}, \nu_{21}, \nu_{22}, \ldots, \nu_{M1}, \nu_{M2}, \psi_S]^T \tag{4.36}$$

for odd ordered systems. As such, for a STF of order $2M+1$, a population, $\mathbb{G}_{\text{STF}}$, of n chromosomes can be written as

$$\mathbb{G}_{\text{STF}} = \left[\mathcal{C}_{\text{STF},1} \vdots \mathcal{C}_{\text{STF},2} \vdots \cdots \vdots \mathcal{C}_{\text{STF},n}\right] = \begin{pmatrix} \rho_{1,1} & \rho_{1,2} & \cdots & \rho_{1,n-1} & \rho_{1,n} \\ \nu_{11,1} & \nu_{11,2} & \cdots & \nu_{11,n-1} & \nu_{11,n} \\ \nu_{12,1} & \nu_{12,2} & \cdots & \nu_{12,n-1} & \nu_{12,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \nu_{M1,1} & \nu_{M1,2} & \cdots & \nu_{M1,n-1} & \nu_{M1,n} \\ \nu_{M2,1} & \nu_{M2,2} & \cdots & \nu_{M2,n-1} & \nu_{M2,n} \\ \psi_{S,1} & \psi_{S,2} & \cdots & \psi_{S,n-1} & \psi_{S,n} \end{pmatrix} \tag{4.37}$$

where $\mathbb{G}_{\text{STF}}$ is a $K_S \times n$ matrix and $\mathcal{C}_{\text{STF},n}$ is the nth chromosome.

4.2.2 Algorithm Parameters

The relative size of a population dictates the nature of the results. For example, large unconstrained populations offer a broad evaluation of the performance surface at the expense of convergence speed and steady-state misadjustment. Conversely, small constrained populations converge quickly at the expense of increased population variance. Further, selecting the appropriate size for a population is directly related to the complexity, or dimensionality, of the problem space. However, the relationship between problem space complexity and requisite population size is typically either

poorly defined or unknown [1]. As such, for constrained problem spaces, the most important concept is often minimum population size. However, because the minimum population size is tightly coupled with the problem space complexity and therefore poorly defined, it is typically refined or tuned a posteriori [1] [33].

For this thesis, the initial size of the population has been restricted to 200; that is, to determine optimal NTFs and STFs, the HOG algorithm was initially configured with a population size of 200 chromosomes. Because the poles of an IIR filter must lie within the unit circle for the filter to be stable, the feasible solutions space has been constrained such that each pole location, γ, lies within the unit circle which implies that

$$|\gamma| \leq 1.$$

Because the interaction between transfer function poles and zeros is implied in the desired magnitude response, constraining the pole locations also indirectly constrains the zero locations. The remaining HOG algorithm parameters are summarized in Table 4.1.

Table 4.1: Low-Pass $\Delta\Sigma$ Modulator Design: HOG Algorithm Parameters

Description	**Value**
Problem Dimension	$K : K = K_N \text{or} K_S$
Population Size	200
Feasible Solution Space	$\lvert\gamma\rvert \leq 1$
Probability of Crossover	$P_c = 0.2$
Probability of Mutation	$P_m = 0.02$
Selection Pressure	$\eta^+ = 1.1$

4.2.3 Population Initialization

Although global optimization algorithms do not require a priori knowledge of the performance surface to determine the optimum solution, generalized constraints on the objective function's solution space can greatly increase the speed of the algorithm's convergence. Seeding the population with known solutions is often employed as a test for conversion latency [32]. It has also been shown that seeding the population with solutions which are known to be near optimal can lead to optimal solutions [32].

Discrete-time $\Delta\Sigma$ modulator NTFs which are designed using both contemporary and traditional techniques have zeros which are distributed on or near segments of the unit circle which correspond with the NTF's stopband [42] [35]. Thus, it is plausible that the optimal zero locations will be distributed on or near segments of the unit circle which correspond with the NTF's stopband as well. Further, if it is assumed that the optimal zero locations are distributed on or near the unit circle, then the optimal pole locations will likely be proximal to the pole locations of both the traditional and contemporary NTF filter designs.

Similarly, discrete-time $\Delta\Sigma$ modulator STFs which are designed using traditional techniques have zeroes which are distributed on or near the segments of the unit circle corresponding to the STF's stopband. Thus, it is plausible that the optimal zero locations will be distributed on or near segments of the unit circle which correspond to the STF's stopband, as well. However, unlike the NTF, which is designed first, the pole locations of the STF are fixed by the NTF design. As such, it is assumed that the zero locations will be proximal to the zero locations for traditional STF filter designs with pole locations determined by the optimal design of the NTF.

Thus, the initial NTF and STF populations, $\mathbb{G}_{\mathrm{NTF},0}$ and $\mathbb{G}_{\mathrm{STF},0}$, are seeded with chromosomes whose alleles are selected randomly about the coefficients of a comparable NTF or STF, respectively. Specifically, their respective chromosomes are

determined by adding zero mean, standard normal random dither, represented by x, with a fixed variance of 10% (normalized with respect to the unit circle) to the each of the Chebyshev polynomial coefficients where the system function is described by (4.27) or (4.33); that is, the kth additive dither element, δ_k, where $\Big\{k \in I : 1 \leq k \leq (K_N \text{ or } K_S)\Big\}$, can be expressed as

$$\delta_k = \frac{1}{\sqrt{0.2\pi}} e^{-\frac{x^2}{0.2}} \tag{4.38}$$

for the random variable x.

Thus, the nth dithered chromosome, $\tilde{\mathcal{C}}_{\text{NTF},n}$, belonging to an initial NTF population, $\mathbb{G}_{\text{NTF},0}$, with a system function of order $2M+1$, can be written as

$$\begin{aligned}
&\tilde{\mathcal{C}}_{\text{NTF},n} = \\
&\Big[(a_1 + \delta_{N,1}), (b_1 + \delta_{N,2}), (c_{11} + \delta_{N,3}), (c_{12} + \delta_{N,4}), (d_{11} + \delta_{N,5}), (d_{12} + \delta_{N,6}), \dots \\
&(c_{M1} + \delta_{N,K_N-4}), (c_{M2} + \delta_{N,K_N-3}), (d_{M1} + \delta_{N,K_N-2}), (d_{M2} + \delta_{N,K_N-1}), (\psi_N + \delta_{N,K_N})\Big]^T \\
&= \Big[\tilde{a}_1, \tilde{b}_1, \tilde{c}_{11}, \tilde{c}_{12}, \tilde{d}_{11}, \tilde{d}_{12}, \tilde{c}_{21}, \tilde{c}_{22}, \tilde{d}_{21}, \tilde{d}_{22}, \dots, \tilde{c}_{M1}, \tilde{c}_{M2}, \tilde{d}_{M1}, \tilde{d}_{M2}, \tilde{\psi}_N\Big]^T.
\end{aligned} \tag{4.39}$$

where $\delta_{N,k}$ corresponds to the kth NTF additive dither element. Because a population is defined as the aggregate of n chromosomes, the initial NTF population, $\mathbb{G}_{\text{NTF},0}$,

can be written as

$$\mathbb{G}_{\mathrm{NTF},0} = \left[\tilde{\mathcal{C}}_{\mathrm{NTF},1} \vdots \tilde{\mathcal{C}}_{\mathrm{NTF},2} \vdots \cdots \vdots \tilde{\mathcal{C}}_{\mathrm{NTF},n}\right] = \begin{pmatrix} \tilde{a}_{11,1} & \tilde{a}_{11,2} & \dots & \tilde{a}_{11,n-1} & \tilde{a}_{11,n} \\ \tilde{b}_{12,1} & \tilde{b}_{12,2} & \dots & \tilde{b}_{12,n-1} & \tilde{b}_{12,n} \\ \tilde{c}_{11,1} & \tilde{c}_{11,2} & \dots & \tilde{c}_{11,n-1} & \tilde{c}_{11,n} \\ \tilde{c}_{12,1} & \tilde{c}_{12,2} & \dots & \tilde{c}_{12,n-1} & \tilde{c}_{12,n} \\ \tilde{d}_{11,1} & \tilde{d}_{11,2} & \dots & \tilde{d}_{11,n-1} & \tilde{d}_{11,n} \\ \tilde{d}_{12,1} & \tilde{d}_{12,2} & \dots & \tilde{d}_{12,n-1} & \tilde{d}_{12,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \tilde{c}_{M1,1} & \tilde{c}_{M1,2} & \dots & \tilde{c}_{M1,n-1} & \tilde{c}_{M1,n} \\ \tilde{c}_{M2,1} & \tilde{c}_{M2,2} & \dots & \tilde{c}_{M2,n-1} & \tilde{c}_{M2,n} \\ \tilde{d}_{M1,1} & \tilde{d}_{M1,2} & \dots & \tilde{d}_{M1,n-1} & \tilde{d}_{M1,n} \\ \tilde{d}_{M2,1} & \tilde{d}_{M2,2} & \dots & \tilde{d}_{M2,n-1} & \tilde{d}_{M2,n} \\ \tilde{\psi}_{N,1} & \tilde{\psi}_{N,2} & \dots & \tilde{\psi}_{N,n-1} & \tilde{\psi}_{N,n} \end{pmatrix} \tag{4.40}$$

where $\mathbb{G}_{\mathrm{NTF},0}$ is a $K_N \times n$ matrix and $\tilde{\mathcal{C}}_{\mathrm{NTF},k}$ is the kth dithered chromosome.

Similarly, the nth dithered chromosome, $\tilde{\mathcal{C}}_{\mathrm{STF},n}$, belonging to an initial STF population, $\mathbb{G}_{\mathrm{STF},0}$, with a system function of order $2M+1$ can be written as

$$\begin{aligned} \tilde{\mathcal{C}}_{\mathrm{STF},n} = \Big[&(\rho_1 + \delta_{S,1}), (\nu_{11} + \delta_{S,2}), (\nu_{12} + \delta_{S,3}), \dots \\ \dots, &(\nu_{M1} + \delta_{S,K_S-2}), (\nu_{M2} + \delta_{S,K_S-1}), (\psi_S + \delta_{S,K_S})\Big]^T \\ = \Big[&\tilde{\rho}_1, \tilde{\nu}_{11}, \tilde{\nu}_{12}, \tilde{\nu}_{21}, \tilde{\nu}_{22}, \dots, \tilde{\nu}_{M1}, \tilde{\nu}_{M2}, \tilde{\psi}_S\Big]^T \end{aligned} \tag{4.41}$$

where $\delta_{S,k}$ corresponds to the kth STF additive dither element. Because a population is defined as the aggregate of n chromosomes, the initial STF population, $\mathbb{G}_{\mathrm{STF},0}$, can

be written as

$$\mathbb{G}_{\mathrm{STF},0} = \left[\tilde{\mathcal{C}}_{\mathrm{STF},1} \middle| \tilde{\mathcal{C}}_{\mathrm{STF},2} \middle| \cdots \middle| \tilde{\mathcal{C}}_{\mathrm{STF},n}\right] = \begin{pmatrix} \tilde{\rho}_{11,1} & \tilde{\rho}_{11,2} & \cdots & \tilde{\rho}_{11,n-1} & \tilde{\rho}_{11,n} \\ \tilde{\nu}_{11,1} & \tilde{\nu}_{11,2} & \cdots & \tilde{\nu}_{11,n-1} & \tilde{\nu}_{11,n} \\ \tilde{\nu}_{12,1} & \tilde{\nu}_{12,2} & \cdots & \tilde{\nu}_{12,n-1} & \tilde{\nu}_{12,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \tilde{\nu}_{M1,1} & \tilde{\nu}_{M1,2} & \cdots & \tilde{\nu}_{M1,n-1} & \tilde{\nu}_{M1,n} \\ \tilde{\nu}_{M2,1} & \tilde{\nu}_{M2,2} & \cdots & \tilde{\nu}_{M2,n-1} & \tilde{\nu}_{M2,n} \\ \tilde{\psi}_{S,1} & \tilde{\psi}_{S,2} & \cdots & \tilde{\psi}_{S,n-1} & \tilde{\psi}_{S,n} \end{pmatrix} \quad (4.42)$$

where $\mathbb{G}_{\mathrm{STF},0}$ is a $K_S \times n$ matrix and $\tilde{\mathcal{C}}_{\mathrm{STF},k}$ is the kth dithered chromosome.

4.2.4 Convergence

Convergence during the evolution of a particular NTF or STF is defined as the lack of further improvement over 100 successive generations following a minimum of 1000 generations. This criterion improves the probability that the evolved transfer functions will be optimal with respect to their objective functions. However, this optimality does not guarantee overall system stability; that is, even optimized NTFs and STFs may be unstable when implemented in a $\Delta\Sigma$ modulator. Thus, overall design convergence is achieved when the $\Delta\Sigma$ modulator is stable and the observed DR and SNR are better than the observations made for both traditional and contemporary $\Delta\Sigma$ modulator designs of like order and OSR.

4.3 Simulation and Modeling

Because SNR and DR are measures of an ADC's effective resolution and because SNR and DR are measured in the frequency domain, the performance of a $\Delta\Sigma$ modulator is characterized by its observed output spectrum. While the linear quantizer model shown in Figure 2.3 is sufficient for theoretical results and preliminary system

modeling, a simulation of the nonlinear system is necessary to determine the realized effective resolution of a $\Delta\Sigma$ modulator [34]. Thus, the designed systems were verified using a simulator based on the block diagram shown in Figure 4.1.

For this thesis, the output, $y(n)$, of a discrete-time $\Delta\Sigma$ modulator is determined using the linear difference equations which correspond to the blocks shown in Figure 4.1. The $\Delta\Sigma$ modulator's output is then bandlimited to the Nyquist frequency of the input signal, f_{NQ}, where $f_{\mathrm{NQ}} \in [-f_s/2\mathrm{OSR}, f_s/2\mathrm{OSR}]$, and then decimated by a factor of OSR/2. Finally, the SNR and DR are calculated using frequency domain analysis of the spectrum of the decimated output signal.

4.3.1 Linear Difference Equations

Recall that the output, $y(n)$, of a $\Delta\Sigma$ modulator as shown in Figure 4.1 can be expressed as

$$Y(z) = \mathrm{STF}(z)X(z) + \mathrm{NTF}(z)E(z) \tag{4.43}$$

where

$$\mathrm{STF}(z) = \frac{F(z)G(z)}{1 + z^{-1}G(z)H(z)} \tag{4.44}$$

and

$$\mathrm{NTF}(z) = \frac{1}{1 + z^{-1}G(z)H(z)}. \tag{4.45}$$

Because STFs and NTFs are modeled as linear recursive filters, they can be written as

$$\mathrm{STF}(z) = \frac{\sum_{k=0}^{N} \alpha_k z^{-k}}{\sum_{k=0}^{N} \beta_k z^{-k}} \tag{4.46}$$

and

$$\mathrm{NTF}(z) = \frac{\sum_{k=0}^{N} \gamma_k z^{-k}}{\sum_{k=0}^{N} \beta_k z^{-k}} \tag{4.47}$$

where $\{\alpha_k\}$, $\{\beta_k\}$, and $\{\gamma_k\}$ are the sets of real coefficients, $\beta_0 = 1$, and N is the order of the numerator and denominator polynomials.

4.3.1.1 Canonical Solution

Because $F(z)$, $G(z)$, and $H(z)$ are rational functions, the STF and NTF can be expressed as

$$\mathrm{STF}(z) = \frac{\left(\frac{F_n(z)}{F_d(z)}\frac{G_n(z)}{G_d(z)}\right)}{1 + z^{-1}\left(\frac{G_n(z)}{G_d(z)}\frac{H_n(z)}{H_d(z)}\right)} = \frac{H_d(z)F_n(z)G_n(z)}{F_d(z)\Big(G_d(z)H_d(z) + z^{-1}G_n(z)H_n(z)\Big)} \tag{4.48}$$

and

$$\mathrm{NTF}(z) = \frac{1}{1 + z^{-1}\left(\frac{G_n(z)}{G_d(z)}\frac{H_n(z)}{H_d(z)}\right)} = \frac{G_d(z)H_d(z)}{G_d(z)H_d(z) + z^{-1}G_n(z)H_n(z)}. \tag{4.49}$$

Because of practical circuit implementation considerations, $F_d(z)$,$H_d(z)$, and $G_n(z)$ are chosen so that

$$F_d(z) = H_d(z) = G_n(z) = 1.$$

Substituting $F_d(z) = H_d(z) = G_n(z) = 1$ into (4.48) and (4.49),

$$\mathrm{STF}(z) = \frac{F_n(z)}{G_d(z) + z^{-1}H_n(z)} \tag{4.50}$$

and

$$\mathrm{NTF}(z) = \frac{G_d(z)}{G_d(z) + z^{-1}H_n(z)}. \tag{4.51}$$

Comparing (4.50) and (4.51) to (4.47) and (4.46), it can be seen that

$$F_n(z) = \sum_{n=0}^{N} \alpha_n z^{-k}, \tag{4.52}$$

$$G_d(z) = \sum_{k=0}^{N} \gamma_k z^{-k}, \tag{4.53}$$

and

$$G_d(z) + z^{-1} H_n(z) = \sum_{k=0}^{N} \beta_k z^{-k}. \tag{4.54}$$

Substituting (4.53) into (4.54) and solving for $H_n(z)$

$$H_n(z) = \sum_{k=0}^{N} (\beta_k - \gamma_k) z^{-k+1}. \tag{4.55}$$

However, because $H_n(z)$ is causal, $\beta_0 - \gamma_0 = 0$ which implies that $\beta_0 = \gamma_0 = 1$. Thus, $H_n(z)$ can be written as

$$H_n(z) = \sum_{k=1}^{N} (\beta_k - \gamma_k) z^{-k+1}. \tag{4.56}$$

Therefore, the LTI modeled subsystems, $F(z)$, $G(z)$, and $H(z)$, can be expressed as

$$F(z) = F_n(z) = \sum_{k=0}^{N} \alpha_k z^{-k}, \tag{4.57}$$

$$G(z) = \frac{1}{G_d(z)} = \frac{1}{\sum_{k=0}^{N} \gamma_k z^{-k}}, \tag{4.58}$$

and

$$H(z) = H_n(z) = \sum_{k=1}^{N} (\beta_k - \gamma_k) z^{-k+1}. \tag{4.59}$$

4.3.1.2 Simulation Solution

Replacing the linear noise model in Figure 4.1 with a nonlinear quantizer, the discrete-time $\Delta\Sigma$ modulator in Figure 4.1 can be modeled as shown in Figure 4.5 where the quantizer's input is denoted as $Q_i(z)$. From observation of Figure 4.5, the

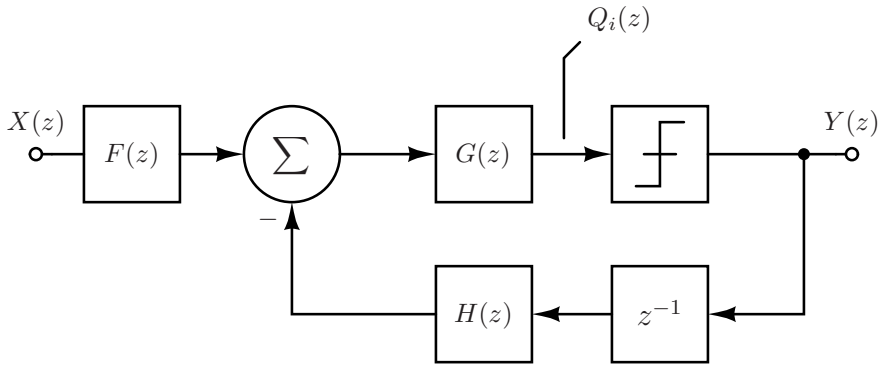

Figure 4.5: Discrete-Time Simulation Model

quantizer's input, $Q_i(z)$, can be given as

$$Q_i(z) = G(z)\left(F(z)X(z) - z^{-1}H(z)Y(z)\right). \tag{4.60}$$

Substituting (4.57), (4.58), and (4.59) into (4.60), the quantizer input, $Q_i(z)$, can be expressed as

$$\begin{aligned} Q_i(z) &= \left(\frac{F_n(z)}{G_d(z)}\right)X(z) - z^{-1}\left(\frac{H_n(z)}{G_d(z)}\right)Y(z) \\ &= \left(\frac{\sum_{k=0}^{N}\alpha_k z^{-k}}{\sum_{k=0}^{N}\gamma_k z^{-k}}\right)X(z) - z^{-1}\left(\frac{\sum_{k=1}^{N}(\beta_k - \gamma_k)\,z^{-k}}{\sum_{k=0}^{N}\gamma_k z^{-k}}\right)Y(z) \end{aligned} \tag{4.61}$$

where $\gamma_0 = 1$. Alternatively, (4.61) can be written such that

$$Q_i(z)\left(\sum_{k=0}^{N}\gamma_k z^{-k}\right) = \left(\sum_{k=0}^{N}\alpha_k z^{-k}\right)X(z) - \left(\sum_{k=1}^{N}\left(\beta_k - \gamma_k\right)z^{-k}\right)Y(z) \tag{4.62}$$

where $\gamma_0 = 1$. Taking the inverse z-transform of (4.62), the quantizer input, $q_i(n)$, can be written as

$$\sum_{k=0}^{N}\gamma_k q_i(n-k) = \sum_{k=0}^{N}\alpha_k x(n-k) - \sum_{k=1}^{N}\left(\beta_k - \gamma_k\right)y(n-k) \tag{4.63}$$

which implies that

$$q_i(n) = \sum_{k=0}^{N}\alpha_k x(n-k) - \sum_{k=1}^{N}\left(\beta_k - \gamma_k\right)y(n-k) - \sum_{k=1}^{N}\gamma_k q_i(n-k). \tag{4.64}$$

Because a single-bit quantizer is implemented, the discrete-time output, $y(n)$, is

$$y(n) = \operatorname{sgn}\left[q_i(n)\right] \tag{4.65}$$

where sgn[$\cdot$] is the signum function.

4.3.2 Decimation Filtering

Consider a $\Delta\Sigma$ modulator that has a sampling frequency, f_s, and an OSR of M, which implies that $f_s = M2f_0$. A $\Delta\Sigma$ modulator's NTF filters the quantization noise power, $P_{e(n)}$, over the quantizer's operational bandwidth, f_{OS}, where $f_{\mathrm{OS}} \in [-f_s/2, f_s/2]$. By bandlimiting the ADC's digital output, $y(n)$, to the input's signal's Nyquist frequency, f_{NY}, where $f_{\mathrm{NY}} \in [-f_0, f_0]$, the power spectral density of the input signal is preserved while the quantization noise power contained in the output signal is significantly decreased.

Ideally, the output signal would be bandlimited by an ideal digital lowpass filter,

referred to as a decimation filter, with a cutoff frequency of $f_s/2M$. However, practical decimation filters require a finite transition bandwidth. As a result, quantization noise power from the transition band will alias into the operational region. To mitigate the aliasing of quantization noise power, a $\Delta\Sigma$ modulator's sampling frequency can be increased and its NTF can be designed such that its stopband corner is increased. For example, consider a 5th order $\Delta\Sigma$ modulator with an equiripple lowpass filter that has a cutoff frequency of $\pi/32$ and a magnitude response as shown in Figure 4.6. After lowpass filtering, the sample rate of the filter's output is much higher than the

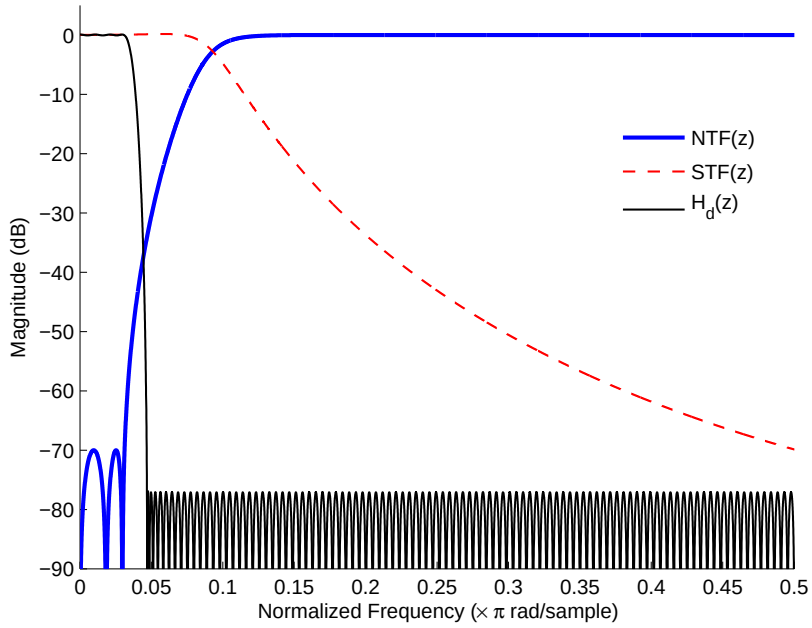

Figure 4.6: $\Delta\Sigma$ Output Decimation Filter Magnitude Response

output signal's Nyquist rate. As such, the output signal's sampling rate is typically reduced, or decimated, to a rate near the input signal's Nyquist rate [26] [14]. In

practice, the decimation operation is combined with the filter and is referred to as decimation filtering.

For this thesis, the output of the simulated $\Delta\Sigma$ modulators was bandlimited using a Parks-McClellan optimal finite impulse response (FIR) lowpass filter. As illustrated in Figure 4.7, the $\Delta\Sigma$ modulator's output, $y(n)$, is lowpass filtered by $H_d(z)$, which was configured with a cutoff frequency, $f_c = f_s/2M$, a transition bandwidth of $f_s/2M$, a maximum passband ripple of 0.1 dB, and a minimum stopband attenuation of 200 dB. Subsequent to filtering, the filter's output signal, $y_f(n)$, was downsampled by a factor of $M/2$.

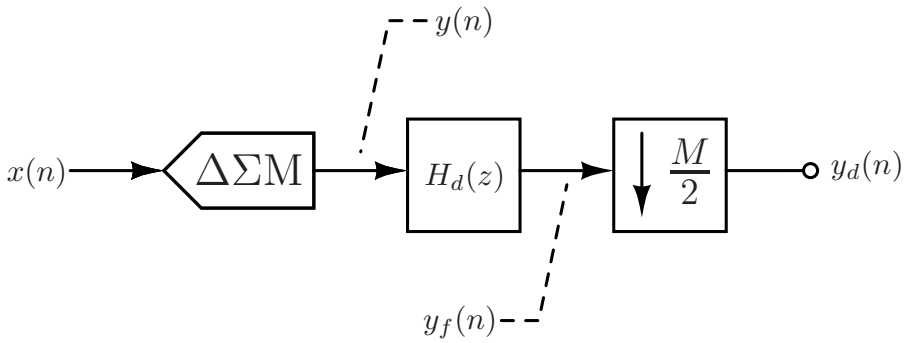

Figure 4.7: $\Delta\Sigma$ Modulator Output Decimation and Filtering Block Diagram

4.3.3 Numerical Analysis

The SNR or DR of a $\Delta\Sigma$ modulator are typically calculated from the characteristics of the $\Delta\Sigma$ modulator's output spectrum; that is, the effective resolution of a $\Delta\Sigma$ modulator can be calculated by using the Discrete Fourier Transform (DFT) or equivalently a Fast Fourier Transform (FFT) of its output.

In the time domain, the power of the finite length decimated output signal, $y_d(n)$,

is calculated as

$$P_{y_d(n)} = \frac{1}{N} \sum_{k=0}^{N-1} y_d^2(k) \tag{4.66}$$

Using Parseval's theorem, the average output power, P_{avg}, of the decimated output signal, $y_d(n)$, can be written as

$$P_{\text{avg}} = \frac{1}{N} \sum_{k=0}^{N-1} |y_d(n)|^2 = \frac{1}{N^2} \sum_{k=0}^{N-1} |Y_d(k)|^2 \tag{4.67}$$

where $Y_d(k)$ corresponds to the kth element, or bin, of the DFT of $y_d(n)$. Thus, a $\Delta\Sigma$ modulator's SNR and DR can be determined by calculating signal and noise power from the $\Delta\Sigma$ modulator's output spectrum. Because the output signal has been decimated by a factor of $M/2$, the post-decimation operational bandwidth is f_{BW}, where $f_{\text{BW}} \in [-f_s/2M, f_s/2M]$. Thus, for a N-point FFT, the FFT bins, $\{k\}$, that correspond to the operational bandwidth of the $\Delta\Sigma$ modulator are $\{k : 0 \leq k \leq N/4 - 1\}$ and $\{k : 3N/4 \leq k \leq N - 1\}$ assuming N is even. Because the output signal is real,

$$Y_d(k) = Y_d^*(N - k),$$

and the power can be calculated using the bins, $\{k : 0 \leq k \leq N/4 - 1\}$.

Once the output signal has been decimated and filtered, windowing is performed to reduce the impact of spectral leakage due to non-coherent sampling [4]. For this thesis, a normalized Chebyshev window with a sidelobe suppression of 200 dB was selected. The Chebyshev window was normalized so that it would not affect the calculated output power spectral density. To illustrate, the windowed output signal, $y_w(n)$, can be written as

$$y_w(n) = y_d(n) w_n(n) \tag{4.68}$$

where $y_d(n)$ corresponds to the filtered and decimated output signal and $w_n(n)$ cor-

responds to the normalized window function. If $y_d(n)$ and $w_n(n)$ are uncorrelated, the power of the windowed output signal, $P_{y_w(n)}$, can be written as

$$P_{y_w(n)} = E\left[y_w^2(n)\right] = E\left[y_d^2(n)\right] E\left[w_n^2(n)\right] = P_{y_d(n)} P_{w_n(n)} \tag{4.69}$$

which implies that $P_{y_w(n)} = P_{y_d(n)}$ when

$$P_{w_n(n)} = E\left[w_n^2(n)\right] = 1.$$

Thus, if a window, $w(n)$, has an average power, $P_{w(n)}$, the normalized window, $w_n(n)$, is

$$w_n(n) = \frac{w(n)}{\sqrt{P_{w(n)}}}$$

so that

$$E\left[w_n^2(n)\right] = E\left[\frac{w^2(n)}{P_{w(n)}}\right] = \frac{1}{P_{w(n)}} E\left[w^2(n)\right] = 1. \tag{4.70}$$

Because windows can smear the signal energy into adjacent FFT bins, the signal power, P_s, can be calculated by summing the DFT coefficients within the bounds of the input signal lobe; that is,

$$P_s = \frac{2}{N^2} \sum_{k=K_1}^{K_2} |Y_w(k)|^2, \tag{4.71}$$

where K_1 and K_2 correspond to the leading and trailing FFT bins of the input signal lobe and N is the length of the DFT. Similarly, the noise power, P_n, can be calculated from the DFT coefficients. In particular, the average noise power, $P_{n,a}$, can be expressed as

$$P_{n,a} = \frac{2}{N^2} \left(\sum_{k=0}^{K_1-1} |Y_w(k)|^2 + \sum_{k=K_2+1}^{N/4-1} |Y_w(k)|^2 + (K_2 - K_1)|\hat{Y}_w^2(k)| \right) \tag{4.72}$$

where K_1 and K_2 correspond to the leading and trailing FFT bins of the fundamental signal lobe, $\hat{Y}_w$ is the estimated average noise magnitude in the fundamental signal lobe's FFT bins, and N is the length of the DFT. Observe that for narrow band input signals, that is, for

$$K_2 - K_1 \ll N/4 - 1,$$

the average noise power can be approximated as

$$P_{n,a} \approx \frac{2}{N^2}\left(\sum_{k=0}^{K_1-1}|Y_w(k)|^2 + \sum_{k=K_2+1}^{N/4-1}|Y_w(k)|^2\right) \tag{4.73}$$

Thus, the SNR expressed in decibels can be calculated by substituting (4.71) and (4.73) into (2.21) such that

$$\mathrm{SNR}_{\mathrm{dB}} = 10\log\left(\frac{P_s}{P_{n,a}}\right) = 10\log\left(\frac{\sum_{k=K_1}^{K_2}|Y_w(k)|^2}{\sum_{k=0}^{K_1-1}|Y_w(k)|^2 + \sum_{k=K_2+1}^{N/4-1}|Y_w(k)|^2}\right) \tag{4.74}$$

Recall that DR is defined as the ratio of the maximum to the minimum detectable signal levels. Because the minimum detectable signal level is determined by the peak noise floor magnitude, the DR can be defined numerically as the ratio of the maximum signal power to a uniform noise floor which is equal in magnitude to the observed peak noise floor magnitude. As such, if $y_{d,e}(n)$ is the noise component of the decimated output signal, $y_d(n)$, the observed peak noise floor, N_p, is defined as

$$N_p = \max\langle|Y_{d,e}(k)|\rangle \tag{4.75}$$

where $Y_{d,e}(k)$ is the DFT of $y_{d,e}(n)$. Thus, the average power, $P_{n,p}$, for a uniform noise

floor with a magnitude equal to the observed peak noise floor value, N_p, is given as

$$P_{n,p} = \frac{2}{N^2} \sum_{k=0}^{N/4-1} N_p = \frac{2}{4N} N_p = \frac{N_p}{2N}. \tag{4.76}$$

Thus, the DR expressed in dB can be calculated by substituting (4.71) and (4.76) into (2.21) such that

$$\begin{aligned} \mathrm{DR}_{\mathrm{dB}} &= 10 \log \left(\frac{P_s}{P_{n,p}} \right) \\ &= 10 \log \left(\frac{\frac{2}{N^2} \sum_{k=K_1}^{K_2} |Y_w(k)|^2}{\frac{N_p}{2N}} \right) \\ &= 10 \log \left(\frac{4}{N_p N} \sum_{k=K_1}^{K_2} |Y_w(k)|^2 \right) \end{aligned} \tag{4.77}$$

where N_p corresponds to the largest noise magnitude contained in the output spectrum.

4.4 Results and Observations

In this section, three $\Delta\Sigma$ design methods are compared. Specifically, the methods are the Delta Sigma (DelSig) Toolbox for MATLAB®, the Chebyshev filter, and the HOG algorithm based design method. The DelSig method claims to maximize the SNR by uniformly distributing the system function zeros over the passband and then optimizing the pole locations. Chebyshev filters generate equiripple stopbands and maximally flat passbands and are typically used to maximize DR. The HOG algorithm based design method maximizes a weighted combination of SNR, DR, and a 1-norm approximation of the passband as described in Section 4.1. Each method is used to design $\Delta\Sigma$ modulators over a range of OSRs and filter orders. A detailed comparison

of 5th and 6th order systems is presented followed by an inclusive summary of all design cases.

Figure 4.8 shows a comparison of the NTF magnitude responses for the various design techniques for a 5th order $\Delta\Sigma$ modulator with an OSR of 32. As illustrated, the peak and average stopband spectrum for NTFs derived by the HOG algorithm is lower than both the classical Chebyshev based NTF and the DelSig toolbox's NTF. Additionally, note that the shape of the stopband frequency response reflects the optimization for a weighted combination of SNR and DR.

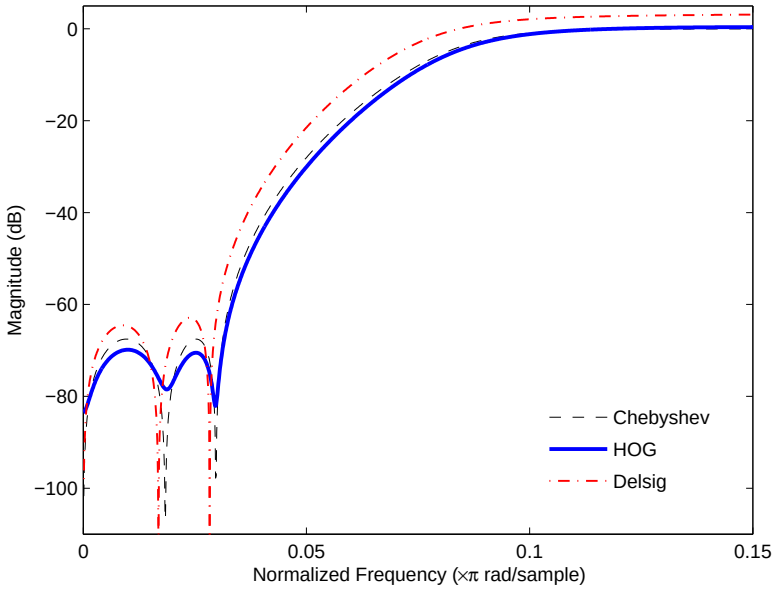

Figure 4.8: 5th Order NTF Comparison
OSR: 32 - BW: π/OSR (rad/sample)

Figure 4.9 shows a comparison of the STF magnitude responses for the corresponding NTFs illustrated in Figure 4.8. The STF designed with the HOG algorithm,

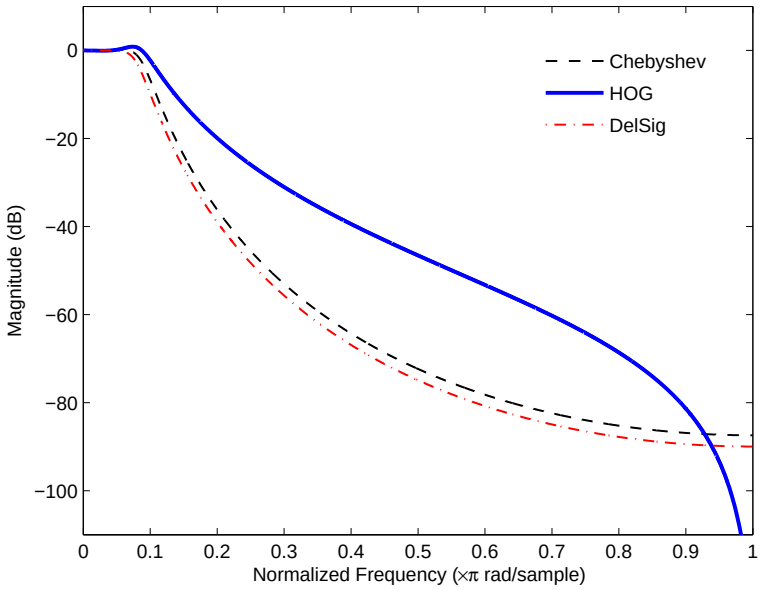

Figure 4.9: 5th Order STF Comparison
OSR: 32 - BW: π/OSR (rad/sample)

shown in Figure 4.9, was optimized as described in Section 4.1.2.

Figure 4.10 illustrates the output spectra for the 5th order $\Delta\Sigma$ modulator that uses the STFs and NTFs shown in Figures 4.8 and 4.9. From the spectra shown in Figure 4.10, the calculated SNRs and DRs for the DelSig Toolbox, Chebyshev filter, and HOG algorithm based design method are summarized in Table 4.2. Based on the SNR and DR values, it can be seen that the HOG algorithm based design produces NTFs and STFs which achieve a higher effective resolution than both the DelSig method and the Chebyshev filter.

Figure 4.11 shows a comparison of the NTF magnitude responses for the various design techniques for a 6th order $\Delta\Sigma$ modulator with an OSR of 32. Again, the peak and average stopband spectrum for NTFs determined by the HOG algorithm is lower than both the classical Chebyshev based NTF and the Delta Sigma toolbox's NTF

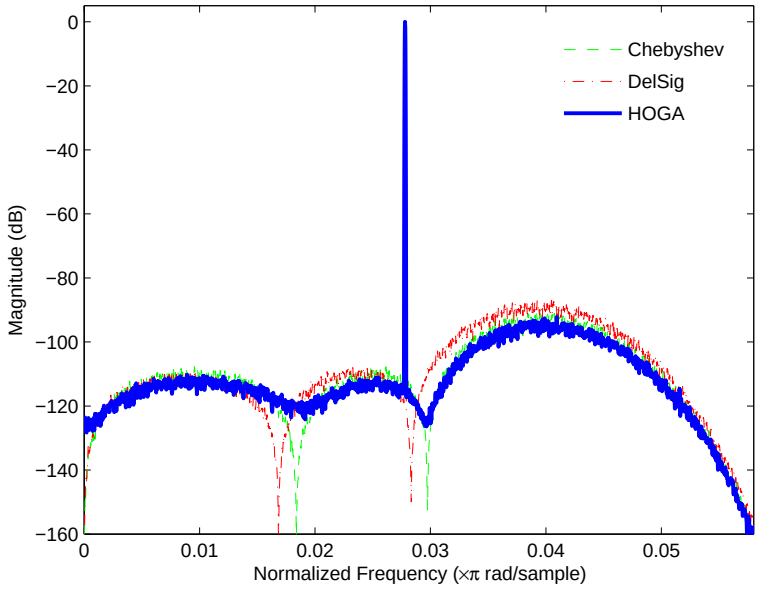

Figure 4.10: 5th Order Output PSD Comparison
OSR: 32 - BW: π/OSR (rad/sample)- FFT Length: 8192

Table 4.2: 5th Order $\Delta\Sigma$ Modulators: Calculated SNRs and DRs

Design Method	$\mathbf{SNR_{dB}}$	$\mathbf{DR_{dB}}$
DelSig Toolbox	83	66
Chebyshev Filter	81	72
HOG Algorithm	87	76

and has a stopband shape which reflects the optimization for a weighted combination of SNR and DR.

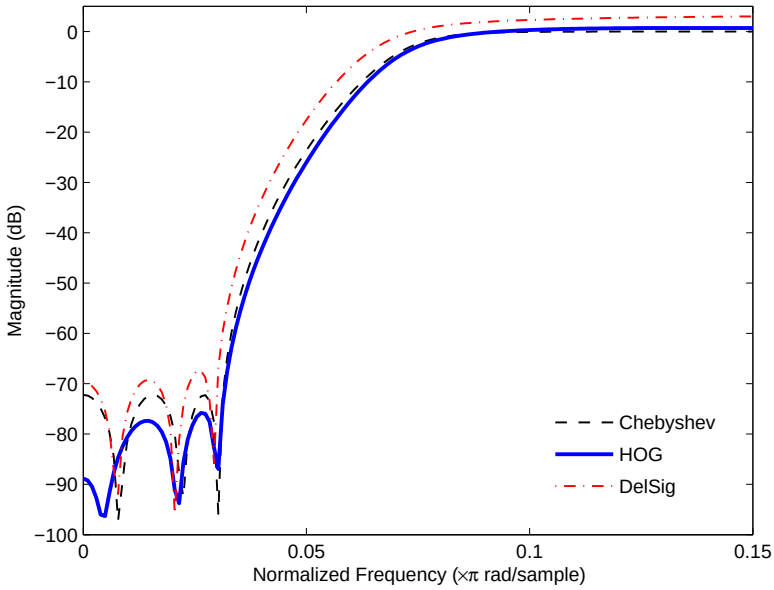

Figure 4.11: 6th Order NTF Comparison
OSR: 32 - BW: π/OSR (rad/sample)

Figure 4.12 shows a comparison of the STF magnitude responses for the corresponding NTFs illustrated in Figure 4.11. Again, the passband of the STF designed with the HOG algorithm, shown in Figure 4.12, was optimized as described in Section 4.1.2.

Figure 4.13 illustrates the output spectra for the 6th order $\Delta\Sigma$ modulator that uses the STFs and NTFs shown in Figures 4.11 and 4.12. From the spectra shown in Figure 4.13, the calculated SNRs and DRs for the DelSig Toolbox, Chebyshev filter, and HOG algorithm based design method are summarized in Table 4.3. Based

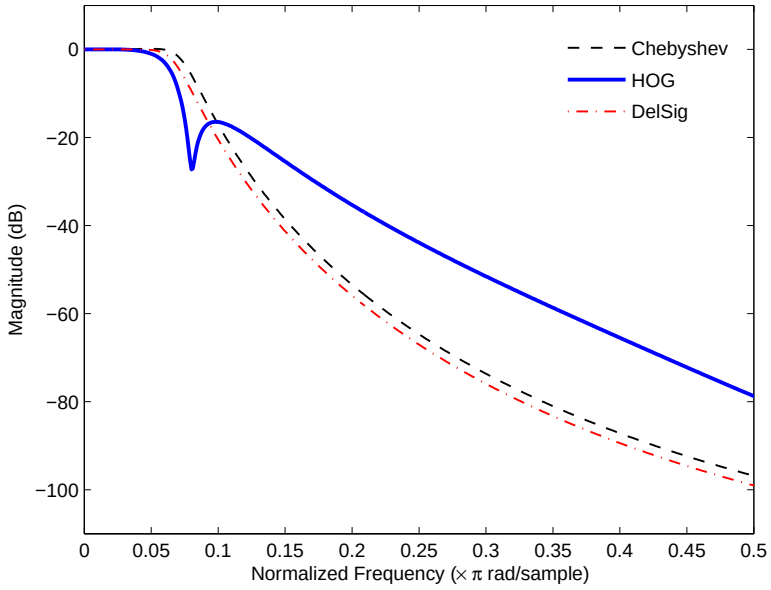

Figure 4.12: 6th Order STF Comparison
OSR: 32 - BW: π/OSR (rad/sample)

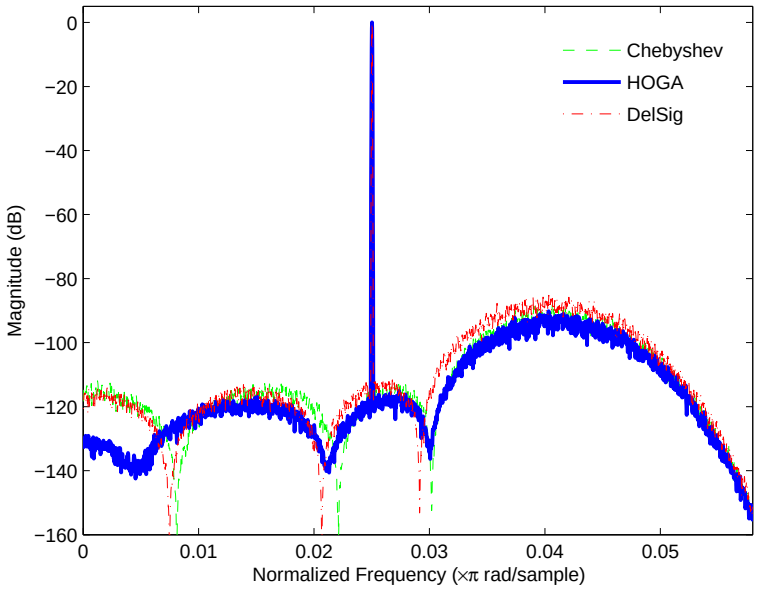

Figure 4.13: 6th Order Output PSD Comparison
OSR: 32 - BW: π/OSR (rad/sample) - FFT Length: 8192

on the SNR and DR values, it can be seen that the HOG algorithm based design produces NTFs and STFs which achieve a higher effective resolution than both the DelSig method and the Chebyshev filter.

Table 4.3: 6th Order $\Delta\Sigma$ Modulators: Calculated SNRs and DRs

Design Method	$\mathbf{SNR_{dB}}$	$\mathbf{DR_{dB}}$
DelSig Toolbox	88	74
Chebyshev Filter	87	77
HOG Algorithm	91	80

Several $\Delta\Sigma$ modulators were designed using the methods previously described. Figures 4.14, 4.15, and 4.16 summarize the resulting SNRs and DRs as a function of filter order for OSRs of 32, 64, and 128, respectively. Note that the HOG algorithm based method provides improved SNR and DR over both the classical and contemporary design methods.

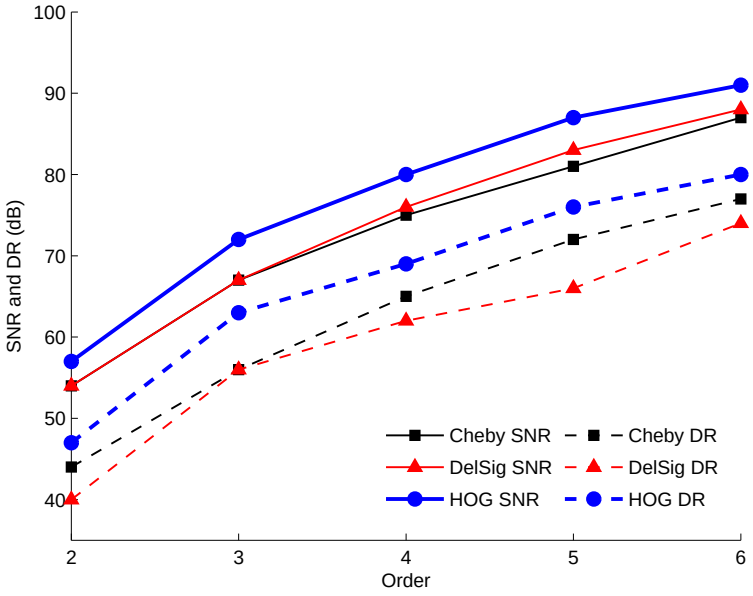

Figure 4.14: SNR and DR Results with OSR = 32

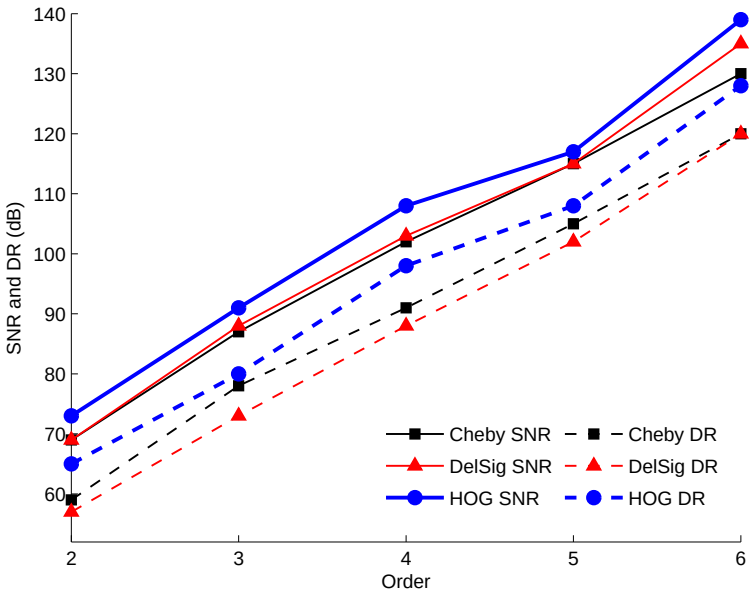

Figure 4.15: SNR and DR Results with OSR = 64

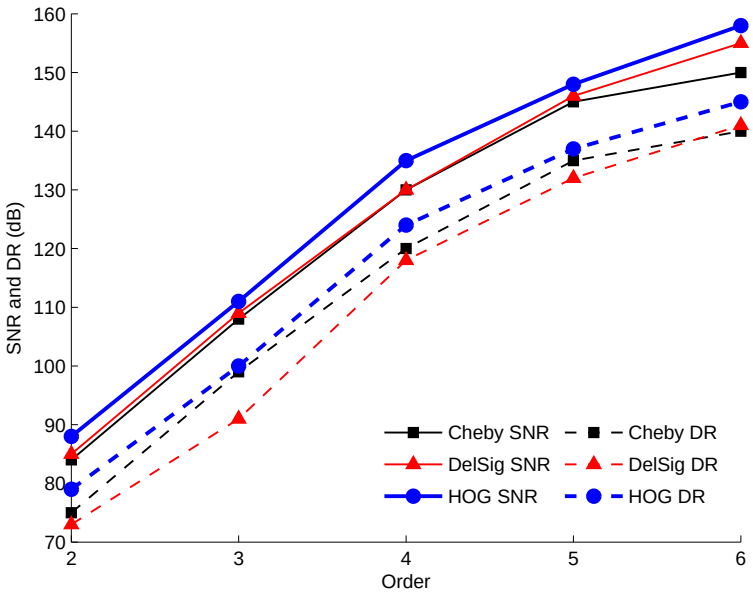

Figure 4.16: SNR and DR Results with OSR = 128

CHAPTER 5

CONCLUSION

The performance of digital algorithms in mixed-signal systems can usually be improved by simply increasing the word length or applying more processing capabilities. However, the performance of a mixed-signal system's ADCs determines the effective resolution of the system's input signals, and as such, the ADC performance is often the limiting factor of the performance for mixed signal systems. Thus, the performance of mixed-signal systems can typically be improved by increasing the effective resolution of their ADCs.

$\Delta\Sigma$ modulators are an ADC architecture that uses relatively simple analog circuitry including a low order quantizer and a feedback loop to sample analog signals with high signal-to-noise ratios (SNRs) and large dynamic ranges (DRs). $\Delta\Sigma$ modulators achieve high SNRs and large DRs by using a feedback loop filter, or noise transfer function (NTF), to attenuate the quantizer's noise in the frequency bands of interest while passing the input signal to the output via the signal transfer function (STF). Thus, the effective resolution of a $\Delta\Sigma$ modulator can be improved by optimizing the NTF and STF for SNR and DR.

For a given oversampling rate (OSR), $\Delta\Sigma$ modulator STFs and NTFs were optimized with respect to a weighted combination of SNR and DR. Thus, when compared to both classical polynomial based design techniques and contemporary EDA based design techniques, the SNR and DR optimization based design was shown to provide increased DR and SNR thereby increasing the $\Delta\Sigma$ modulator's effective resolution.

This optimized design problem was solved using a novel genetic algorithm (GA) which was developed in Chapter 3. This GA, which is referred to as the Hybrid Orthogonal Genetic (HOG) algorithm uses modified genetic operators to improve algorithm performance and accuracy when compared to traditional GAs. In Chapter 3, it was shown that the HOG algorithm can effectively solve widely accepted benchmark multimodal problems across a broad range of dimensions. It was also shown that the robust methods inherent to the HOG algorithm design provide repeatable highly-optimized solutions to known optimal solutions. As such, the HOG algorithm based method of designing $\Delta\Sigma$ modulators is shown to determine optimal NTFs and STFs with respect to SNR and DR.

REFERENCES

[1] J.T. Alander. On Optimal Population Size of Genetic Algorithms. In *CompEuro '92. Computer Systems and Software Engineering, Proceedings.*, pages 65–70, 1992.

[2] Thomas Back. Optimization by Means of Genetic Algorithms. pages 1–8, Germany: Technische Universita Ilmenau, 1991.

[3] Tobias Blickle and Lothar Thiele. A Comparison of Selection Schemes Used in Evolutionary Algorithms. *Evolutionary Computation*, 4(4):361, 1996.

[4] M. Cerna and A. F. Harvey. National Instruments Corportation: Application Note 041: The Fundamentals of FFT-Based Signal Analysis and Measurement, 2000.

[5] X. Chen and T.W. Parks. Design of IIR Filters in the Complex Domain. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, 38(6):910–920, 1990.

[6] James A. Cherry and W. Martin Snelgrove. *Continuous-Time Delta-Sigma Modulators for High-Speed A/D Conversion: Theory, Practice and Fundamental Performance Limits*. Springer, 1st edition, September 1999.

[7] Wikipedia contributors. Analog-to-Digital Converter, December 2007.

[8] G. Cortelazzo and M. Lightner. Simultaneous Design in Both Magnitude and Group-delay of IIR and FIR Filters Based on Multiple Criterion Optimization. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, 32(5):949–967, 1984.

[9] George B. Dantzig and Mukund N. Thapa. *Linear Programming: 1: Introduction*. Springer, 1 edition, 1997.

[10] Angela M. Dean and Daniel Voss. *Design and Analysis of Experiments*. Springer, corrected edition, December 2000.

[11] D.B. Fogel. What is evolutionary computation? *Spectrum, IEEE*, 37(2):26, 28–32, 2000.

[12] Fred W. Glover and Gary A. Kochenberger. *Handbook of Metaheuristics*. Springer, 1 edition, 2003.

[13] R.M. Gray. Quantization Noise Spectra. *Information Theory, IEEE Transactions on*, 36(6):1220–1244, 1990.

[14] Monson H. Hayes. *Schaum's Outline of Digital Signal Processing*. McGraw-Hill, 1 edition, August 1998.

[15] Shinn-Ying Ho, Li-Sun Shu, and Jian-Hung Chen. Intelligent Evolutionary Algorithms for Large Parameter Optimization Problems. *Evolutionary Computation, IEEE Transactions on*, 8(6):522–541, 2004.

[16] Hwei Hsu. *Schaum's Outline of Probability, Random Variables, and Random Processes*. McGraw-Hill, 1 edition, October 1996.

[17] David Johns and Ken Martin. *Analog Integrated Circuit Design*. Wiley, 1 edition, November 1996.

[18] N. Karaboga and B. Cetinkaya. Design of minimum phase digital IIR filters by using genetic algorithm. In *Signal Processing Symposium, 2004. NORSIG 2004. Proceedings of the 6th Nordic*, pages 29–32, 2004.

[19] Walt Kester. Which ADC architecture is right for your application? *Analog Dialogue*, 39(6), June 2005.

[20] Walt Kester. Analog Devices: Analog to Digital Converters: Tutorial MT-022: ADC Architectures III: Sigma-Delta ADC Basics, February 2006.

[21] Mücahit Kozak and Izzet Kale. *Oversampled Delta-Sigma Modulators: Analysis, Applications and Novel Topologies*. Springer, 1 edition, July 2003.

[22] B. P. Lathi. *Modern Digital and Analog Communication Systems*. Oxford University Press, USA, 3 edition, March 1998.

[23] Yiu-Wing Leung and Yuping Wang. An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization. *Evolutionary Computation, IEEE Transactions on*, 5(1):41–53, 2001.

[24] Hong Li, Yong-Chang Jiao, Li Zhang, and Ze-Wei Gu. *Genetic Algorithm Based on the Orthogonal Design for Multidimensional Knapsack Problems*, pages 696–705. 2006.

[25] Linear Technology Corporation, 1630 McCarthy Blvd., Milpitas, CA 95035-7417. *LTC2209 16-Bit, 160Msps ADC*, 2007.

[26] Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck. *Discrete-Time Signal Processing*. Prentice Hall, 2nd edition, February 1999.

[27] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill Companies, 2Rev ed edition, December 1984.

[28] P. W. Poon and J. N. Carter. Genetic Algorithm Crossover Operators for Ordering Applications. *Computers & Operations Research*, 22(1):135–147, 1995.

[29] Kenneth V. Price, Rainer M. Storn, and Jouni A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Springer, 1 edition, December 2005.

[30] L. Rabiner, N. Graham, and H. Helms. Linear Programming Design of IIR Digital Filters with Arbitrary Magnitude Function. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, 22(2):117– 123, 1974.

[31] Colin R. Reeves and Jonathan E. Rowe. *Genetic Algorithms - Principles and Perspectives: A Guide to GA Theory*. Springer, 1 edition, December 2002.

[32] Ruhul Sarker, Masoud Mohammadian, and Xin Yao. *Evolutionary Optimization*. Springer, 1 edition, 2002.

[33] J. David Schaffer. -. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 4–7, George Mason University, June 1989. Morgan Kaufmann Publishers, Inc.

[34] R. Schreier. An Empirical Study of High-Order Single-Bit Delta-Sigma Modulators. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions on]*, 40(8):461–466, 1993.

[35] Richard Schreier and Gabor C. Temes. *Understanding Delta-Sigma Data Converters*. Wiley-IEEE Press, November 2004.

[36] Peter Stubberud and Greg Lull. Design of Bandpass Delta Sigma Modulators using Differential Evolution. In *IASTED: Signal and Image Processing, Proceedings of*, 2006.

[37] Peter A. Stubberud and Matthew E. Jackson. A Hybrid Orthogonal Genetic Algorithm for Global Numerical Optimization. In *Proceedings of the 2008 19th International Conference on Systems Engineering - Volume 00*, pages 282–287. IEEE Computer Society, 2008.

[38] Kit-Sang Tang, Kim-Fung Man, Sam Kwong, and Zhi-Feng Liu. Design and Optimization of IIR Filter Structure Using Hierarchical Genetic Algorithms. *Industrial Electronics, IEEE Transactions on*, 45(3):481–487, 1998.

[39] George B. Thomas, Maurice D. Weir, Joel Hass, and Frank R. Giordano. *Thomas' Calculus, 11th Edition*. Addison Wesley, 11th edition, October 2004.

[40] Jinn-Tsong Tsai, Jyh-Horng Chou, and Tung-Kuan Liu. Optimal Design of Digital IIR Filters by Using Hybrid Taguchi Genetic Algorithm. *Industrial Electronics, IEEE Transactions on*, 53(3):867– 879, 2006.

[41] Jinn-Tsong Tsai, Tung-Kuan Liu, and Jyh-Horng Chou. Hybrid Taguchi Genetic Algorithm for Global Numerical Optimization. *Evolutionary Computation, IEEE Transactions on*, 8(4):365–377, 2004.

[42] M. E. Van Valkenburg. *Analog Filter Design*. Oxford University Press, USA, new ed edition, June 1995.

[43] D. Whitley. The GENITOR Algorithm and Selective Pressure: Why Rank-Based Allocation of Reproductive Trials is Best. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 116–123. Morgan Kaufmann, 1989.

VITA

Graduate College
University of Nevada, Las Vegas

Matthew Edward Jackson

Home Address:
540 Truffles Street
Henderson, Nevada 89015

Degrees:
Bachelor of Science, Electrical Engineering, 2003
University of Wyoming

Thesis Title: Optimal Design of Discrete-Time $\Delta\Sigma$ Modulators

Thesis Examination Committee:
Chairperson, Professor, Dr. Peter Stubberud, Ph.D.
Committee Member, Professor, Dr. Yahia Bagzous, Ph.D.
Committee Member, Professor, Dr. Sahjendra Singh, Ph.D.
Committee Member, Professor, Dr. Brendan O'Toole, Ph.D.