12-2017

Fundamental Limit of Analog Multiplication in Linear Discriminant Classifier

Md Munir Hasan
University of Tennessee

To the Graduate Council:

I am submitting herewith a thesis written by Md Munir Hasan entitled "Fundamental Limit of Analog Multiplication in Linear Discriminant Classifier." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Jeremy Holleman, Major Professor

We have read this thesis and recommend its acceptance:

Benjamin J. Blalock, Syed K. Islam

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Fundamental Limit of Analog Multiplication in Linear Discriminant Classifier

A Thesis Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Md Munir Hasan

December 2017

dedicated to my parents and relatives.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Professor Jeremy Holleman, for the support and guidance he provided me. His knowledge and expertise in research has been an immense help in my graduate studies. His mentoring has helped me grow in the research and professional environment.

I am grateful to Professor Benjamin J. Blalock for sharing his time with me on different problems I had. He has taught me a great deal about analog electronics. I am also grateful to Professor Syed Islam for his guidance in my studies. His knowledge, experience and valuable suggestions have helped me improve my research.

I would like to thank all the ISS group members for their sincere help and friendship. I also thank Dana Bryson for her help throughout the years.

Lastly, I would like to thank Center for International Education and International House for their smooth handling of complicated processes that helped me make my stay here enjoyable.

Abstract

In this thesis analog implementation of a machine learning algorithm, Linear Discriminant Analysis, is analyzed and shown how it performs on a classification problem. Analog machine learning has emerged as a promising field that provides advantages over its digital counterpart in power consumption, circuit area and scalability. Analog computation achieves its efficiency from the physics of device or circuit operation. This allows analog computation to operate on very low signal levels. However, low signal levels make itself vulnerable to noise. Excessive noise levels can render the machine learning system unstable and prone to making wrong decisions. To ensure reasonable accuracy of the system it is essential to understand how noise behaves and propagates along the system.

A key component in analog implementation of the Linear Discriminant Analysis is the analog multiplier. A noise analysis is done for the multiplier to show how noise varies with multiplication factor. This also produces a relationship between signal to noise ratio and energy consumption that gives us a limit of accuracy obtained from the multiplier for a given energy consumption. Numerical analysis is provided to show that Linear Discriminant Analysis is well suited for the classification problem. The performance of a hardware implementation of the analog classifier in commercially available 130nm silicon process is also presented. With four feature input currents and three classes to classify the classifier consumes around 4nW of power. The testing process shows that the classifier is able to perform basic classification task in the presence of noise.

Table of Contents

1 Introduction **1**

2 Modeling Algorithm of the Classification Process **3**

2.1 The DARPA Database 4

2.2 Choice of Features 4

2.3 Behavior of the Feature for Classification 5

2.4 The Linear Discriminant Analysis 7

2.5 MATLAB simulation on DARPA data 11

2.6 Digital filter 11

2.7 Peak Frequency Selection Using Classification Error 13

2.8 Classification on All Class 14

2.9 Effect of Noise on Classification 18

3 Effect of Noise on The Analog Classifier **20**

3.1 Different types of noise in analog circuits 20

3.1.1 White noise 21

3.1.2 Flicker Noise 22

3.2 Noise in Commercially available 130nm CMOS process 24

3.2.1 Measurement of K_w 24

3.2.2 Measurement of K_f 25

3.3 Minimum detectable signal 28
3.4 Noise in analog multiplier circuit 31
3.4.1 Multiplier circuit 31
3.4.2 Noise in the multiplier 32
3.4.3 Signal to noise ratio 37
3.4.4 Fundamental Limit of Multiplication 37
3.4.5 Effective number of bits and Energy per multiplication . . 38
3.4.6 Equivalent relationship in Digital system 39
3.4.7 Multiplication in Classifier 41
3.4.8 Energy per multiply accumulate 43
3.4.9 Effect of noise in classifier decision making 44

4 Testing Process of Linear Classifier **45**
4.1 Current measuring circuit 47
4.2 TIA Offset 48
4.3 Transconductance Board 49
4.4 Testbench PCB 50
4.5 LabView Setup 50
4.6 Floating gate weight estimation 53
4.7 Multiplier current variation with input current 54
4.8 Class separability 55
4.9 Speed of classification 57
4.10 Input referred noise 57
4.11 Input current to the classifier 59
4.12 Setting specific decision boundary 61

5 Conclusion **64**

Bibliography **66**

Appendix **68**

A Floating Gate Programing 69

A.1 Tunneling 69

A.2 Injection 70

Vita **71**

List of Tables

2.1 Description of items in *.mat* file. 4

4.1 Description of digital output signals from LabView 51

4.2 Description of analog data read out 52

4.3 Input current to chip and stability as VDD is varied 60

4.4 Injection and Tunneling signal 62

List of Figures

2.1 Energy spectrum of acoustic signals of different classes plotted on top of each other. Microphone is on the target 5
2.2 Energy spectrum of vibration signal of different classes plotted on top of each other. Accelerometer is on the target. 6
2.3 Histogram of acoustic energy for generator and quiet class at (a) 20Hz (b) 40Hz frequencies. Microphone is on the target. 7
2.4 Histogram of vibration energy for generator and quiet class at (a) 20Hz (b) 40Hz frequencies. Accelerometer is on the target. 8
2.5 Digital Filter response used in extracting energy from time domain signal. 12
2.6 Classification error obtain by sweeping frequencies in two dimension. 13
2.7 Energy spectrum for generator and quiet class audio file for urban environment and microphone place on the target 14
2.8 Scatter plot of audio energies at (a) 60Hz-80Hz pair (b) 20Hz-40Hz pair. 15
2.9 Class boundary for acoustic frequency features found from GA. Other class boundaries are not shown for simplicity. 15
2.10 Class boundary for acoustic and vibration frequency features found from GA. Other class boundaries are not shown for simplicity. . . 16

2.11 Class boundary for acoustic and vibration frequency features found from GA. Other class boundaries are not shown for simplicity. . . 17
2.12 Class boundary for vibration frequency features found from GA. Other class boundaries are not shown for simplicity. 18
2.13 Classification error for center frequency variation. Each variation is done within the given range 10^4 times. 19

3.1 MOSFET input noise model at low frequency 23
3.2 Circuit used to find κ 25
3.3 Calculation of κ. For lpnfet $V_{th0} = 553.3mV$. 26
3.4 Circuit used to calculate flicker noise 27
3.5 Output noise current from simulation of a lpnfet $W = 160n$, $L = 120n$, $I_d = 250pA$. 27
3.6 Total input referred noise for lpnfet ($W = 160n$, $L = 120n$, $I_d = 250pA$) simulation vs model prediction. 28
3.7 Minimum detectable signal level at input at low frequencies as power and area is varied (a) Area is constant at minimum size (W=160nm, L=120nm) (b) Power is set at 1nW. 30
3.8 Differential transconductance amplifier used as multiplier circuit . 31
3.9 Shape of tanh(x) function 33
3.10 Small-signal noise circuit of the multiplier 33
3.11 Simplified small-signal noise circuit of the multiplier 34
3.12 Output noise current from multiplier made of ideal BJT and from hand analysis 36
3.13 Energy consumption with number of bits. 39

3.14 comparison of precision for digital and analog multiplication. m is varied, $f_h = 1GHz$, $I_B = 1nA$, $V_{DD} = 1V$, $C = 1pF$, $W = 1\mu m$, $L = 1\mu m$, $m = 0.5$ (a) area is fixed at $W = 1\mu m$, $L = 1\mu m$ in analog (b) power consumption is fixed at $1\mu W$. in analog 41
3.15 Multiplier current summing in classifier. 42
3.16 Classification result (a) without noise (b) with noise. There are some misclassification at the boundary between two class. 44

4.1 Classifier topology in the chip 46
4.2 The color represents the regions of classes. Red represents [1,0,0], blue is [0,1,0], and green is [0,0,1]. 47
4.3 Circuit for pA current measurement 48
4.4 Offset Calculation for pA current measuring circuit. 49
4.5 Transimpedance board 50
4.6 LabView code for digital signal delivery 51
4.7 LabView code for analog signal delivery 52
4.8 LabView code for analog signal measurement 53
4.9 Analog multiplier circuit 54
4.10 Floating gate weight estimation 55
4.11 LabView code for input current sweep 56
4.12 Multiplier output current variation with input current 56
4.13 Separation of classes 57
4.14 Rise time of a WTA signal 58
4.15 Wrong decisions near decision boundary 59
4.16 CDF and PDF calculation 60
4.17 Floating gate memory circuit 62
4.18 Specific decision boundary set in classifier chip 63

Chapter 1

Introduction

There has been a revival of interest in the discipline of subthreshold mode of transistor operation and analog design as the digital design technology is reaching its physical limit both in terms of energy consumption and speed. The MOS processing technology is progressing towards smaller dimensions. Along with it brings the problem of subthreshold power dissipation also known as leakage power in digital circuits when the transistor is assumed to be turned off. For leakage current typically on the order of 1nA, the leakage power for a system with millions of transistors can be significantly large. With the transistor getting smaller, the supply voltages are also scaling down. But with smaller supply voltages it is hard to maintain reliable performance in digital circuits as other unwanted effects takes over. This is also true for analog circuits. To maintain low power systems and unwanted power dissipation, it is more promising to decrease operating current to the order of pA or nA range. This range of current is easily achieved by transistors operating in the subthreshold regime. Moreover the current voltage relationship in subthreshold regime is shown to be similar to the operation of biological neuron which pioneered the field of the neuromorphic computation [1]. Neuromorphic computation is electronic computation inspired by the analog nature of the neural systems. This takes advantage of the natural physics of analog device operation to carry out computation. This makes it possible to do large amount of computation using only few transistors. Thus a huge savings in area and power can be made over digital implementation of the same computation. For this reason a lot of machine learning techniques which typically uses digital circuits are leaning towards using analog low power circuits. Analog circuits also become advantageous over digital circuits when it comes to implementing deep reinforcement learning and scaling the system according to the computational complexity. However, implementation of such machine learning systems brings the trade of speed and accuracy because analog circuits inherently noisy, subject to mismatch and may need to be slower to maintain a good signal to noise ratio. Despite these challenges there has been a tremendous undertaking in implementing neuromorphic systems specially to make artificial biological systems like electronic cochlea [2] and electronic vision [3]. The way neuromorphic computation and machine learning is moving forward it might not be wrong to think that the future of autonomous artificially intelligent systems will be largely dependent on the progress of intelligent low power analog circuit design.

This thesis explores implementation of linear discriminant classifier algorithm in commercially available 130nm silicon technology using low power analog circuit design techniques. The goal is to implement a sensor that is able to identify presence and absence of objects like car, truck, generator using acoustic and vibration signals. The power budget of the sensor is only 10nW. In this thesis the applicability of linear discriminant analysis in solving the classification problem has been explained. Then circuit design techniques to reliably perform the classification process in the presence of noise is explored. Finally experimental results from fabricated chip is presented.

Chapter 2

Modeling Algorithm of the Classification Process

The task is to identify different objects like car, generator, truck or nothing based on the audio signal and/or accelerometer signal. When presented with any audio signal and/or accelerometer signal, the system should be able to classify it as one of the mentioned objects. Any classification problem involves extracting useful features that helps the computer or any computing device to make intelligent guess on which object those features may be associated with. The general idea is similar objects will show similar feature signature. Consequently different objects will show feature signature that is different to each other. This way any classification algorithm can model the behavior of the features and and make decision when a new feature is presented.

This chapter describes the process of feature selection, choice of classification algorithm and how effective it is on solving this classification problem.

2.1 The DARPA Database

The DARPA database is arranged in *.mat* file. Each *.mat* file contains the sets of data as shown in table 2.1. The notes section describes the instruments (microphone, accelerator, magnetometer, RF etc.). One set of data is located in `/storage/iss/data/darpa_nzero_data/new_sensor_data`. There is total 104 *.mat* files here.

Table 2.1: Description of items in *.mat* file.

Item	Size	Type	Description
notes	17x99	char	notes on data acquisition method
t	1x4500000	double	time sample at which it is taken
ts	4500000x17	double	amplitudes at time t
17 columns in ts represent sensor data from 17 different sensor location			

2.2 Choice of Features

Different objects (generator, car or truck) produce different sounds and vibration due to their differences in weight and structure. This information should be present in the spectrum of the sound and vibration signals. Each class of objects is expected to produce different sound and vibration spectrum. Fig.2.1 shows the average sound and Fig.2.2 vibration spectra of audio and accelerometer recording samples collected from DARPA database.

The shaded regions indicate the span of the energy for all the data. It can be clearly seen that most of the energy is concentrated on the frequencies that shows a peak in the spectrum. These peak frequencies differ from class to class. For this reason the energy at the peak frequencies are chosen as features for classification. Different classes will show differences in energy at any chosen peak frequency. This should provide enough information for the classifier to make decision. Also

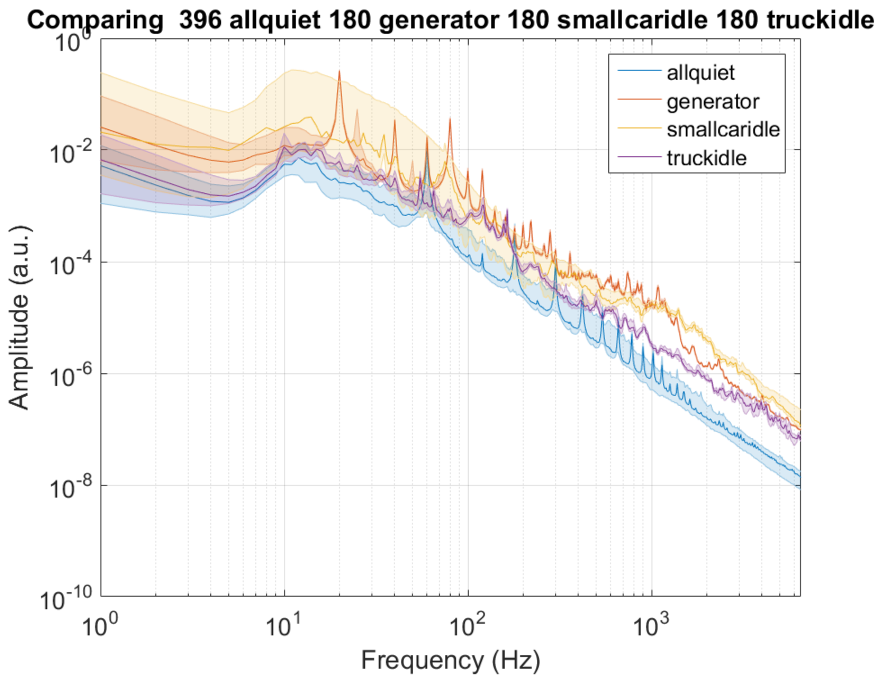


Figure 2.1: Energy spectrum of acoustic signals of different classes plotted on top of each other. Microphone is on the target

peak frequency presents maximum separation of energy content across class which should allow large classification separation.

2.3 Behavior of the Feature for Classification

An analysis of how much the energy at the peak frequencies varies will be useful in making decision on which classification algorithm could be used. A histogram of energy at the peak frequencies of recording at different distances is performed. Each audio and acceleration recording contains 90 seconds of data. Energy content at peak frequencies within $100ms$ of data is recorded. This way each audio or acceleration file of 90 seconds produces 900 samples of peak energy at peak frequencies. These 900 samples of energy of each file is used to plot histogram.

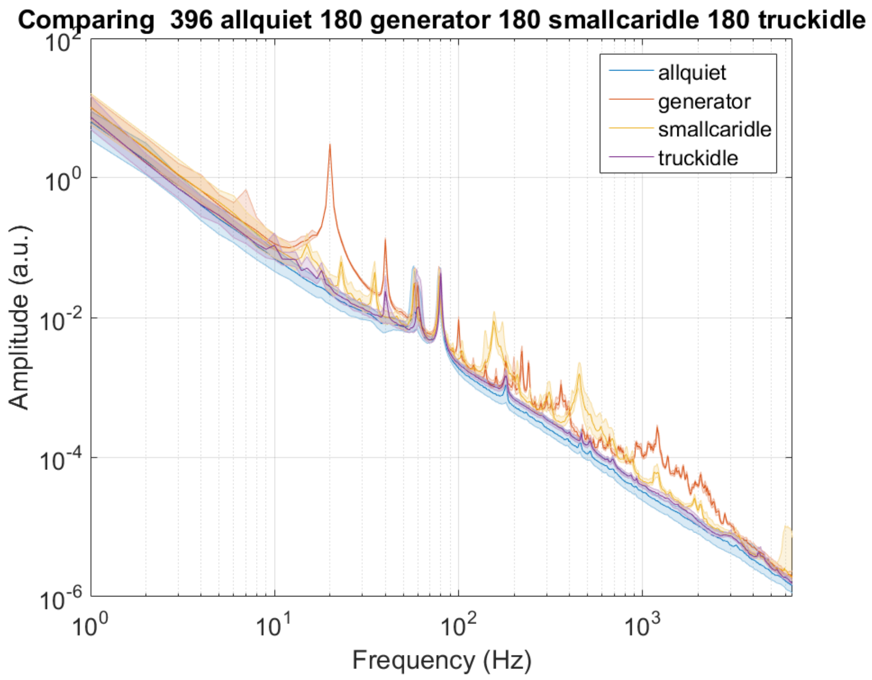
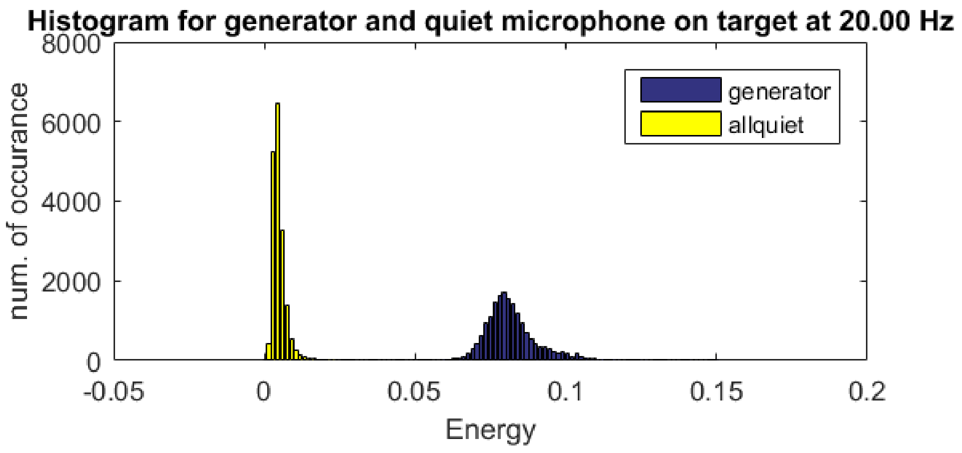


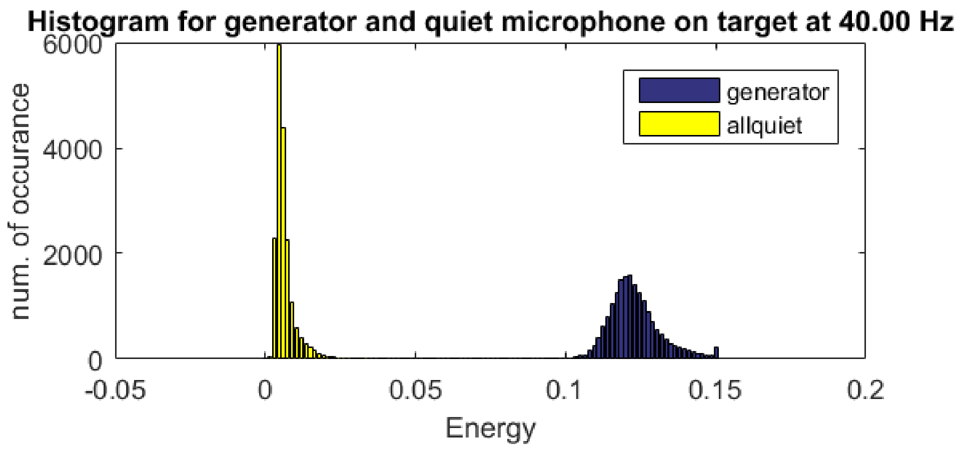
Figure 2.2: Energy spectrum of vibration signal of different classes plotted on top of each other. Accelerometer is on the target.

Fig.2.3 shows a histogram for generator and quiet class at two frequencies. These histograms shows how the energy at peak frequencies are distributed. It can be seen that the energy follows more or less Gaussian distribution pattern. Histogram of vibration energy shown in Fig.2.4 also shows similar Gaussian distribution.

This kind of pattern helps linear discriminant algorithm to model the data and make decision boundaries. The linear discriminant analysis only uses addition and multiplication to make decisions. These two operations can be easily implemented in subthreshold analog circuits.



(a)



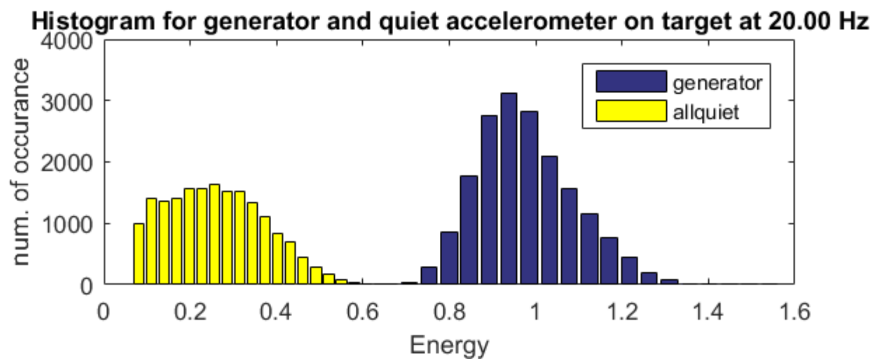
(b)

Figure 2.3: Histogram of acoustic energy for generator and quiet class at (a) 20Hz (b) 40Hz frequencies. Microphone is on the target.

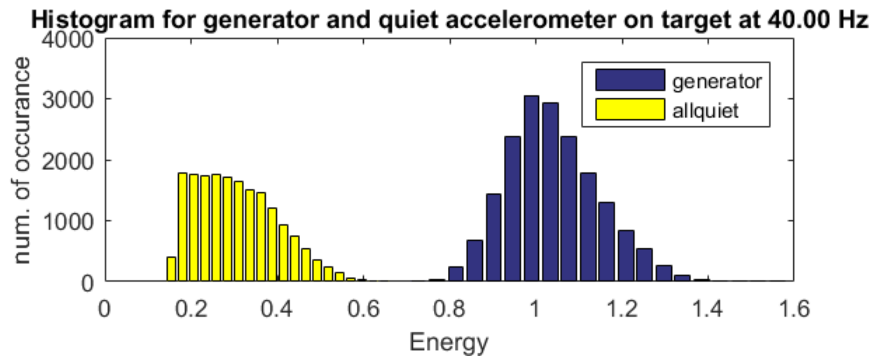
2.4 The Linear Discriminant Analysis

A brief description of the linear discriminant analysis is given here. For class label $C \in \{1, 2, \cdots K\}$ and feature vector $X \in \mathbb{R}^p$ the classification problem can be described as the probability of a class being $C = j$ given a feature set $X = x$. Mathematically

$$F(x) = P(C = j \mid X = x) \tag{2.1}$$



(a)



(b)

Figure 2.4: Histogram of vibration energy for generator and quiet class at (a) 20Hz (b) 40Hz frequencies. Accelerometer is on the target.

If this function is evaluated across all of the class labels then the label with the highest probability gives the class that $X = x$ belongs to. Hence the Eq.2.1 can written as

$$f(x) = \underset{j=1,2,\cdots K}{\arg\max}\, P(C = j \mid X = x) \tag{2.2}$$

Using Bayes' Rule we can write

$$\begin{aligned} P(C = j \mid X = x)P(X = x) &= P(X = x \mid C = j)P(C = j) \\ P(C = j \mid X = x) &= \frac{P(X = x \mid C = j)P(C = j)}{P(X = x)} \end{aligned} \tag{2.3}$$

Using Eq.2.3 in EQ.2.2 can be written as

$$
\begin{aligned}
f(x) &= \underset{j=1,2,\cdots K}{\arg\max} \frac{P(X=x \mid C=j)P(C=j)}{P(X=x)} \\
f(x) &= \underset{j=1,2,\cdots K}{\arg\max} P(X=x \mid C=j)P(C=j) \\
f(x) &= \underset{j=1,2,\cdots K}{\arg\max} P(X=x \mid C=j)\cdot \pi_j
\end{aligned} \tag{2.4}
$$

Because $P(X = x)$ does not depend on j and is constant, removing that term from equation does not influence the function. Here $\pi_j = P(C = j)$ is the prior probability of class j. Rearranging Eq.2.2 into Eq.2.4 allows for us to estimate the conditional class density $P(X = x \mid C = j)$ from the sample data which is the training data. Linear Discriminant Analysis approximates this rule by modeling conditional class densities as multivariate normals.

$$
h_j(x) = P(X=x \mid C=j) = N(\mu_j, \Sigma) \tag{2.5}
$$

i.e. each class j has its own mean $\mu_j \in \mathbb{R}^p$, but shares a common covariance matrix $\Sigma \in \mathbb{R}^{p\times p}$. Hence the multivariate normal density

$$
h_j(x) = \frac{1}{(2\pi^{p/2})det(\Sigma)^{1/2}} e^{-\frac{1}{2}(x_i-\mu_j)^T\Sigma^{-1}(x_i-\mu_j)} \tag{2.6}
$$

We want to find j so that $P(X = x \mid C = j)\cdot\pi_j = h_j(x)\cdot\pi_j$ is largest. Since $\log .$ is a monotonic function, we can consider maximizing $\log h_j(x)\cdot\pi_j$ over $j = 1, 2, \cdots K$. We can define the rule

$$
\begin{aligned}
f^{LDA}(x) &= \underset{j=1,2,\cdots K}{\arg\max} \log\left[\frac{1}{(2\pi^{p/2})det(\Sigma)^{1/2}} e^{-\frac{1}{2}(x_i-\mu_j)^T\Sigma^{-1}(x_i-\mu_j)}\cdot\pi_j\right] \\
&= \underset{j=1,2,\cdots K}{\arg\max}\left[x^T\Sigma^{-1}\mu_j - \frac{1}{2}\mu_j^T\Sigma^{-1}\mu_j + \log\pi_j\right]
\end{aligned}
$$

$$= \underset{j=1,2,\cdots K}{\arg\max}\, \delta_j(x) \tag{2.7}$$

We call $\delta_j(x), j = 1, 2, \cdots K$ the discriminant functions. When we replace π_j, μ_j, Σ with their sample estimates, based on the labeled observations $y_i \in 1, 2, \cdots K$, $x_i \in \mathbb{R}^p$, $i = 1, 2, \cdots n$,

$$\begin{aligned}
\hat{\pi}_j &= \frac{n_j}{n} \\
\hat{\mu_j} &= \frac{1}{n_j} \sum_{y_i=j} x_i \\
\hat{\Sigma} &= \frac{1}{n-K} \sum_{j=1}^{k} \sum_{y_i=j} (x_i - \hat{\mu_j})(x_i - \hat{\mu_j})^T
\end{aligned}$$

The rule can then be written as

$$\hat{f}^{LDA}(x) = \underset{j=1,2,\cdots K}{\arg\max}\, \hat{\delta}_j(x) \tag{2.8}$$

where $\hat{\delta}_j(x)$ is the estimated discriminant function of class j,

$$\begin{aligned}
\hat{\delta}_j(x) &= x^T \hat{\Sigma}^{-1} \hat{\mu_j} - \frac{1}{2} \hat{\mu_j}^T \hat{\Sigma}^{-1} \hat{\mu_j} + \log \hat{\pi}_j \\
&= a_j + b_j^T x
\end{aligned} \tag{2.9}$$

where $a_j = -\frac{1}{2}\hat{\mu_j}^T \hat{\Sigma}^{-1} \hat{\mu_j} + \log \hat{\pi}_j$ and $b_j = \hat{\Sigma}^{-1} \hat{\mu_j}$. For a Given $X = x$ we use Eq.2.8 to find the output class. Eq.2.9 is just a set of equations of lines. It can be written in expanded form as follows.

$$\begin{aligned}
\hat{\delta}_1 &= b_{11} * i_1 + b_{12} * i_2 + ... + w_{1n} * i_n + a_1 \\
\hat{\delta}_2 &= b_{21} * i_1 + b_{22} * i_2 + ... + b_{2n} * i_n + a_2 \\
&\vdots
\end{aligned} \tag{2.10}$$

$$\hat{\delta_m} = b_{m1} * i_1 + b_{m2} * i_2 + ... + b_{mn} * i_n + a_m$$

The decision boundary between two classes can also be found from these equations. The boundary exists where the values of the equations for two classes become equal. The decision boundary between classes j, k is the set of all $X \in \mathbb{R}^p$ such that $\hat{\delta_j(x)} = \hat{\delta_k(x)}$, i.e.

$$\begin{aligned} a_j + b_j^T x &= a_k + b_k^T x \\ (a_j - a_k) + (b_j^T - b_k^T)x &= 0 \end{aligned} \tag{2.11}$$

This is the equation of the line that defines the decision boundary of class j and k.

2.5 MATLAB simulation on DARPA data

For simplicity only generator and quiet classes are chosen first. These two classes are used to show how effective the linear classifier is in differentiating two classes. The energy at any given frequency is extracted by a digital filter. A brief description of the filter used is given below.

2.6 Digital filter

An analog 2nd order resonator filter design is used to make digital filter using MATLAB's *impinvar* function. The analog resonator is given by Eq.2.12

$$F(s) = \frac{1}{s^2 + \frac{2\pi f}{Q}s + (2\pi f)^2} \tag{2.12}$$

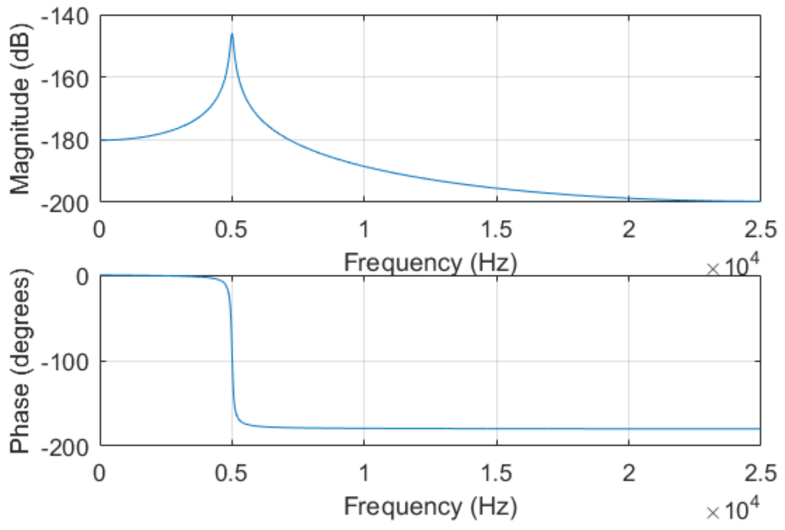


Figure 2.5: Digital Filter response used in extracting energy from time domain signal.

Here f is the resonator frequency and Q is the quality factor. For Q=50, f=5kHz and a sampling frequency of 50kHz the digital response of the filter is given in Fig.2.5. This filter is applied to the time domain signal to filter out information at a given frequency. Then the output time domain signal is squared and summed to get energy at that given frequency.

We have used frequencies from 1Hz to 200Hz at increment of 1Hz. After that a 1% increment of frequency is used. Which means after 200Hz the frequencies are $200(1.01), 200(1.01)^2, \cdots$ etc. This is because energies below 200Hz can be easily captured. After 200Hz an increment of 1% is used because it corresponds to the quality factor. Q=100 provides an increment of $1/Q$ percent.

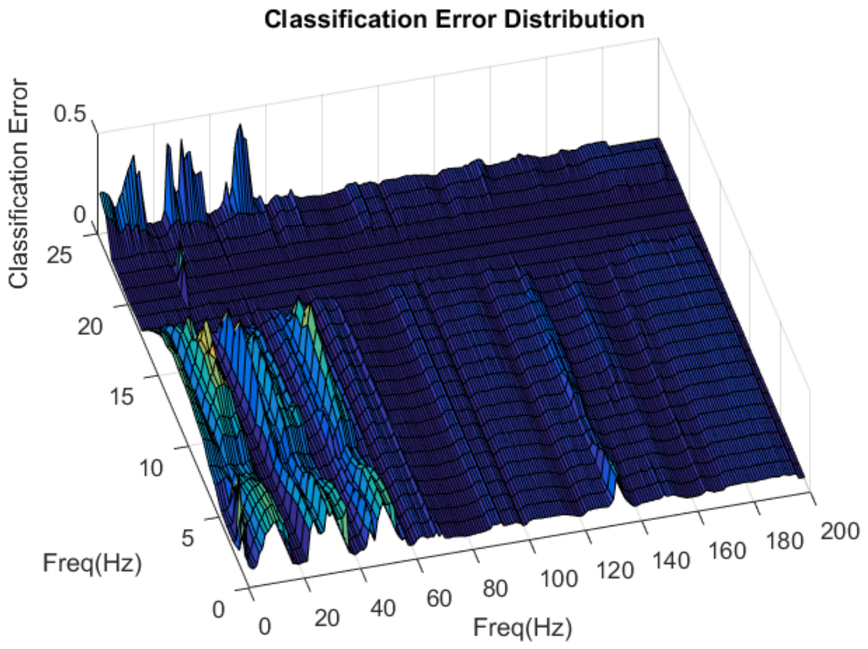


Figure 2.6: Classification error obtain by sweeping frequencies in two dimension.

2.7 Peak Frequency Selection Using Classification Error

We use linear discriminant analysis to do classification over a range of frequencies to see which combinations of frequencies gives us best classification error. To this first the classification is done on features from two frequencies. The frequencies swept over a range and in each iteration classification error is recorded. Fig.2.6 shows a surface plot of such a sweep done on generator and quiet audio file in urban environment and microphone placed on the target.

It can be seen that classification error valleys occur at the harmonics of 20Hz. The reason can be evident from the energy spectrum of the audio as shown in Fig.2.7. At harmonics of 20Hz the energy separation is the greatest. Hence it is expected to see best class separation at these frequencies. Same result is

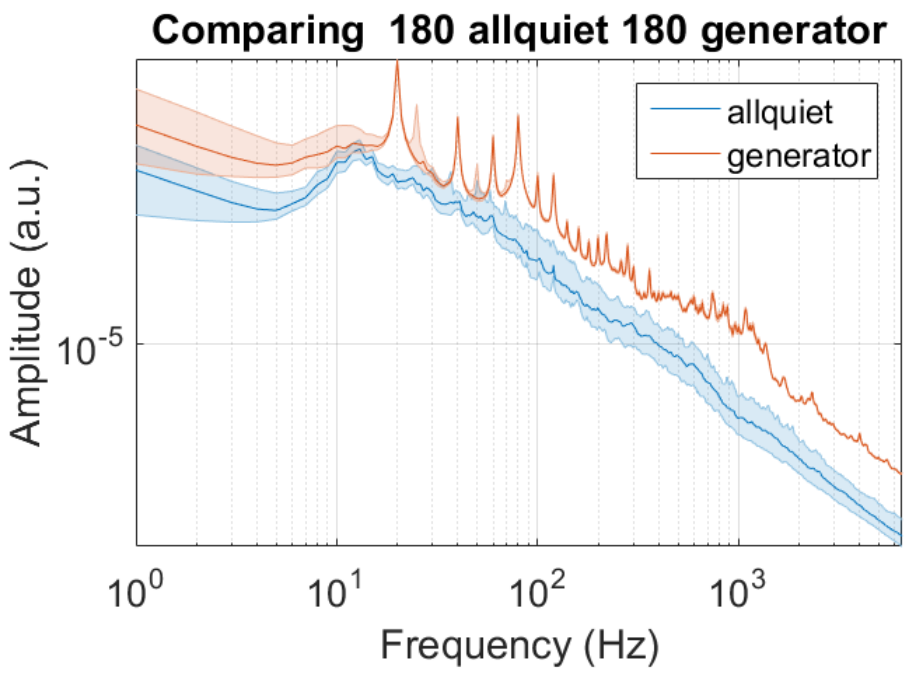


Figure 2.7: Energy spectrum for generator and quiet class audio file for urban environment and microphone place on the target

obtained using Genetic Algorithm (GA) to optimize the classification error for four frequencies. The result is classification error of 0% at harmonics of 20Hz. The Class separation can be better seen in the scatter plot of features at those frequencies in Fig.2.8. The clusters are well separated and a line can be easily drawn between the clusters meaning the linear classifier is able to model this easily.

2.8 Classification on All Class

It was easy to find out peak frequencies for classification for the simple case of generator and quiet class with audio file recorded with microphone placed on the target. But it is challenging to do classification for all class as there are a lot of overlaps of energies as seen in Fig.2.1 and Fig.2.2. For this reason we

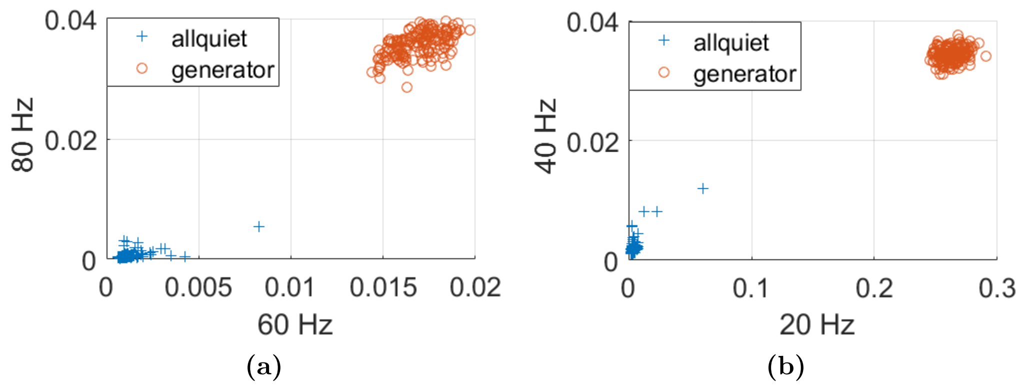


Figure 2.8: Scatter plot of audio energies at (a) 60Hz-80Hz pair (b) 20Hz-40Hz pair.

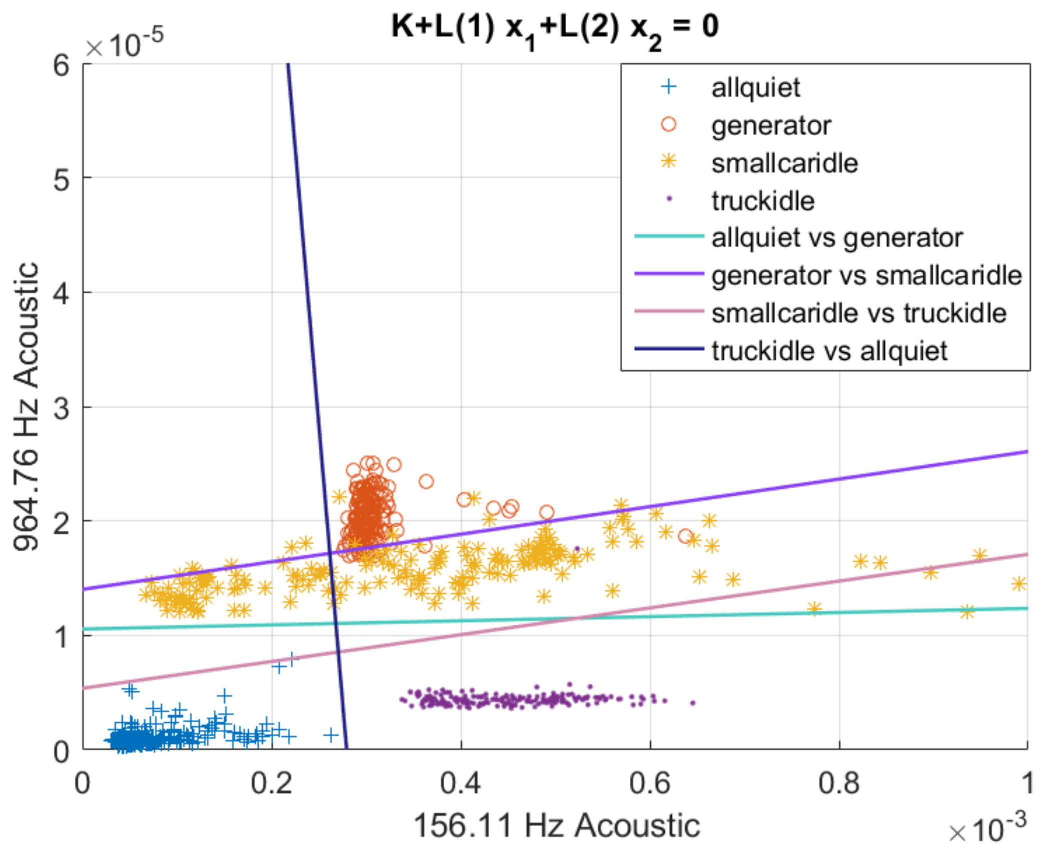


Figure 2.9: Class boundary for acoustic frequency features found from GA. Other class boundaries are not shown for simplicity.

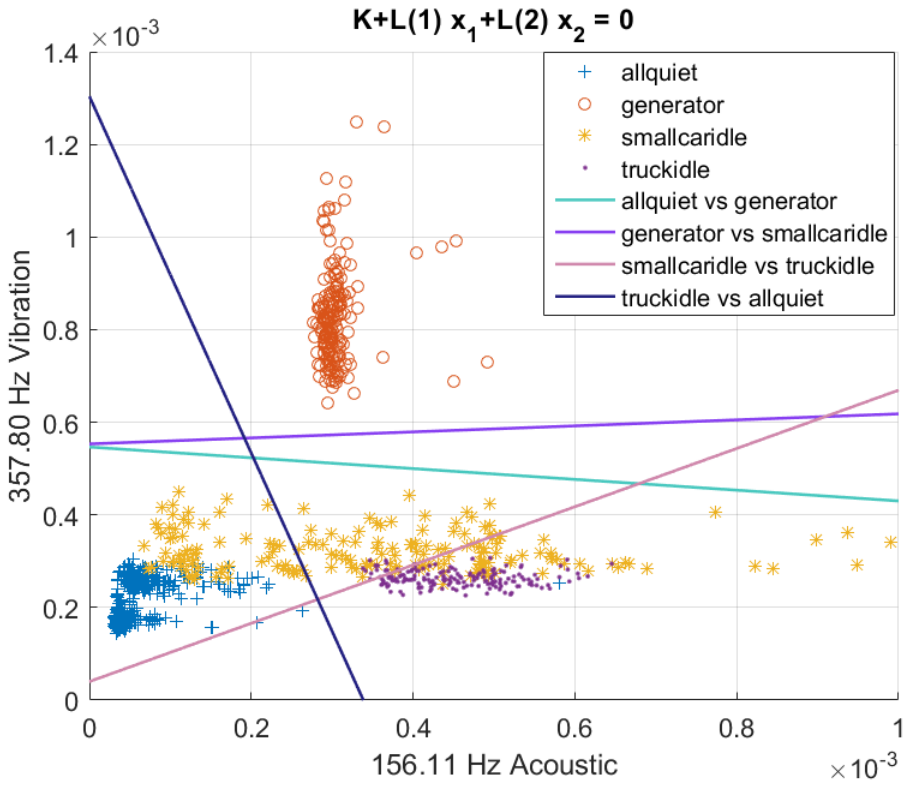


Figure 2.10: Class boundary for acoustic and vibration frequency features found from GA. Other class boundaries are not shown for simplicity.

include the information of vibration data as well in classification process. We limit our analysis to only microphone on the target and accelerometer on the target data. GA is configured to use two acoustic frequencies and two vibration frequencies. After running GA to minimize the classification error the output frequencies were 156.18Hz, 964.76Hz for acoustic frequencies and 357.8Hz and 827.94Hz for vibration frequencies. The minimized error was 0.53%. How well these frequencies separates the classes can be observed approximately with two feature classification and classification boundary.

Fig.2.9 shows the class boundary between different for the acoustic frequency pair. The boundaries are found from linear discriminant classification. The boundaries are reasonably well placed to minimize the misclassification. The

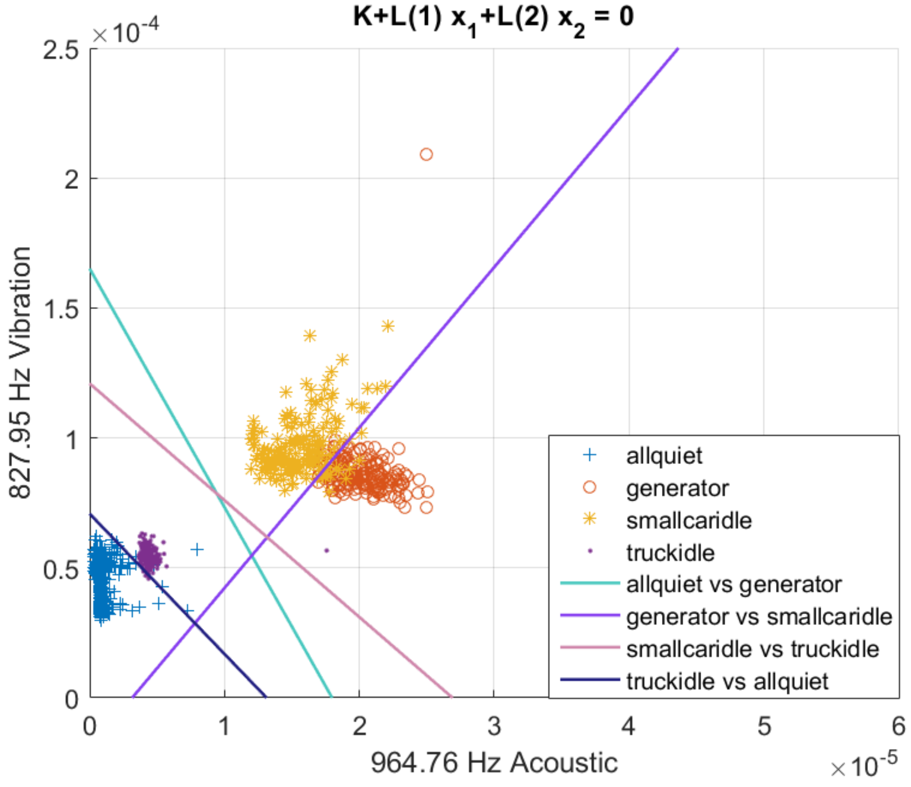


Figure 2.11: Class boundary for acoustic and vibration frequency features found from GA. Other class boundaries are not shown for simplicity.

generator class energy is very well concentrated. But for other classes the energies are spread over a range. This makes it challenging to minimize the classification error to zero. Similar things can be observed from other frequency pairs. Fig.2.10 show decision boundary for one acoustic and vibration frequency pair. Fig.2.11 shows another acoustic and vibration frequency pair. Fig.2.12 shows vibration frequency pair. It might be misunderstood from these figures that a lot of instances are misclassified. But it should be noted that these figures shows decision boundaries for analysis done on two features. With four features the error is only 0.5%. From this analysis it can be said that linear discriminant analysis is well suited to solve this classification problem.

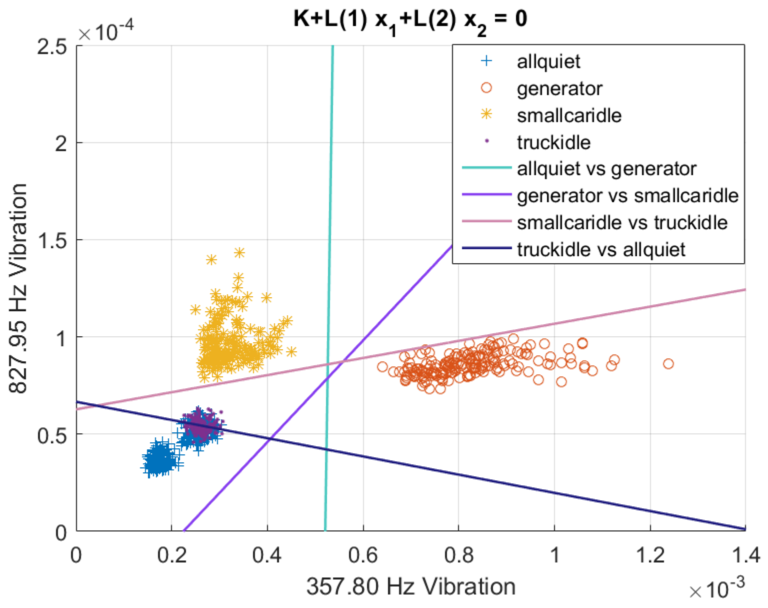


Figure 2.12: Class boundary for vibration frequency features found from GA. Other class boundaries are not shown for simplicity.

2.9 Effect of Noise on Classification

The actual classifier will take feature input from MEMS based frequency resonators. The sensors will have some noise as output. The center frequency might shifted. Moreover there will be unwanted signals captured by the sensors. These noises will affect the classification process. If there is too much noise the classifier might produce incorrect classification results. We need to see how robust the linear classifier is to noise.

First the base classifier model is chosen. The weights of the linear discriminant is calculated from training data of 70% randomly selected data of total database using the frequencies found from GA. The rest 30% is used for testing. For testing we vary the frequencies found from GA randomly within certain amount. Then we use the energies at those frequencies for testing against the trained model. For a certain amount frequency variation, the frequency is varied randomly 10^4 times.

For each of those 10^4 times the classification error against the base trained model is calculated. Then a histogram of the error is plotted. Fig.2.13 shows such a histogram plot. For $\pm 5Hz$ variation the error remains within 1%. As the frequency variation becomes larger the probability of getting higher classification error also increases. Within $\pm 30Hz$ frequency variation the %Error can be over 20% as seen from Fig.2.13(d). This analysis shows that little variation of resonator frequencies does not significantly harms the output of the classifier. This also shows that the frequency choice returned by the GA can be taken as relatively accurate measure of peak energy frequencies.

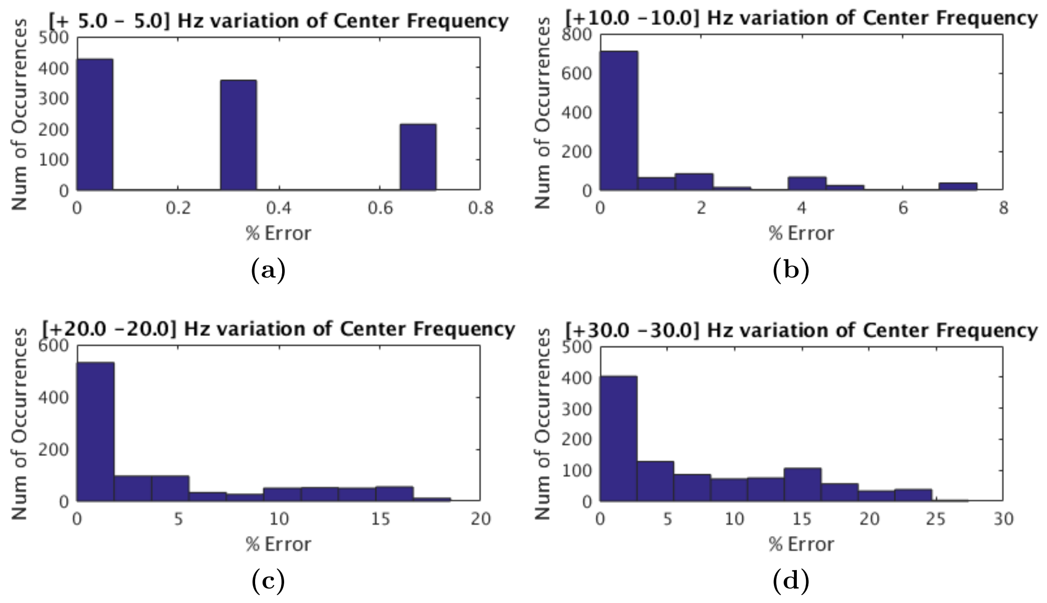


Figure 2.13: Classification error for center frequency variation. Each variation is done within the given range 10^4 times.

Chapter 3

Effect of Noise on The Analog Classifier

Analog circuits are affected by noise much more than digital circuits. Typically analog circuits are cascaded so that noise form all the stages accumulates at the output. This makes it challenging to achieve a good signal to noise ratio (SNR) at the output while maintaining reasonable power and area constraints. The signal levels are often small in lower power circuits hence it is important to consider noise in low power systems. In this chapter the effect of noise in the analog classifier will be discussed and how it affects the output will be analyzed.

3.1 Different types of noise in analog circuits

There are many types of noise that affect the analog circuits. But the most common ones that we can readily observe will be discussed and taken into account. There is an important theorem that relates input and output noise power which will be useful in noise analysis. The theorem states that for any Linear Time Invariant (LTI) system characterized by transfer function $H(f)$ will have output noise power $S_Y(f)$ for input noise power $S_X(f)$ following way

$$S_Y(f) = S_X(f)|H(f)|^2 \tag{3.1}$$

where f is frequency. Using this we can find the input referred noise if we know the noise at the output. Another important theorem is that if the noise sources are uncorrelated then we can find the total noise power by superposition. If there are noise sources with power $\bar{v}_{n1}^2, \bar{v}_{n2}^2, \cdots \bar{v}_{nm}^2$ then the total power is the sum of the powers.

$$\bar{v}_{nT}^2 = \bar{v}_{n1}^2 + \bar{v}_{n2}^2 + \cdots + \bar{v}_{nm}^2 \tag{3.2}$$

Most noise sources found in analog circuits are uncorrelated. Hence we can use the superposition to find the total noise power.

3.1.1 White noise

White noise is composed of two components, thermal and shot noise. Thermal is a noise that is universal to all electronic equipment. It is believed to be caused by random thermal motions of charge carriers. And shot noise is believed to be caused by discrete random arrivals of the charge carriers traversing an energy barrier. It is accepted that shot noise require DC current flow whereas thermal noise requires no current flow. But It was shown that the inherent process of the two noise is same [4, chapter 11] and we do not need to include both noise in analysis. Shot noise is just thermal noise but is due to one directional current flow where as thermal noise is due to both directional current flow. In weak inversion, current is one direction diffusion current and noise current in weak inversion is well modeled as shot noise. The shot noise power spectral density (PSD) in a

MOSFET in subthreshold region operating in saturation is given by

$$\bar{I}_w^2 = 2q\bar{I}\Delta f \tag{3.3}$$

where q is the charge on the charge carrier, $\bar{I}$ is the mean current flowing through the device and Δf is the system bandwidth. The small signal transconductance of a MOSFET in saturation operating in subthreshold region is given by

$$g_m = \frac{\kappa I_D}{U_T} \tag{3.4}$$

where κ is gate channeling coupling coefficient also know as subthreshold exponential factor and $U_T = kT/q$ is the thermal voltage. Here k is Boltzmann constant and T is absolute temperature. Using Eq.3.3 and Eq.3.4 together with Eq.3.1 the input referred noise voltage at the gate can be found as

$$\begin{aligned}
\bar{v}_w^2 &= \frac{\bar{I}_s^2}{g_m^2} \\
\bar{v}_w^2 &= \frac{2kTU_T}{\kappa^2}\frac{1}{I_D}\Delta f \\
\bar{v}_w^2 &= \frac{K_w}{I_D}\Delta f \qquad (V^2) \\
\bar{v}_w^2 &= \frac{K_w}{I_D} \qquad (V^2/Hz)
\end{aligned} \tag{3.5}$$

where $K_w = 2kTU_T/\kappa^2$. From the above equation it can be seen that the noise is inversely proportional to the current flowing through the device. Hence by increasing the device current white noise can be reduced.

3.1.2 Flicker Noise

Another dominant noise in the MOSFET is flicker noise. It is also called 1/f noise because the PSD follows a 1/f characteristic. This noise has significant power at low frequencies. So we need to account for that in our analysis. The input referred flicker noise is given by [5]

$$\bar{v}_f^2 = \frac{K_f}{A}\frac{1}{f} \qquad (V^2/Hz)$$
$$\bar{v}_f^2 = \frac{K_f}{A}\int_{f_l}^{f_h}\frac{df}{f} = \frac{K_f}{A}\ln\frac{f_h}{f_l} \qquad (V^2) \tag{3.6}$$

where K_f is process dependent fit constant, A is the area under the gate and f_h and f_l is, respectively, highest and lowest frequencies of operation. The constant K_f can also be defined as flicker noise density at 1Hz for a unit size transistor. From the equation it can be seen that flicker noise depends both on the area and the current flowing through the device.

The total noise power at the input of the transistor is the sum of the noise powers. Using Eq.3.2 the total noise at the input and output is

$$\bar{v}_n^2 = \frac{K_w}{I_D} + \frac{K_f}{A}\frac{1}{f} \tag{3.7}$$
$$\bar{I}_o^2 = (\frac{K_w}{I_D} + \frac{K_f}{A}\frac{1}{f})(\frac{\kappa I_D}{U_T})^2 \tag{3.8}$$

Fig.3.1 shows how the two noise sources are replaced by input referred noise.

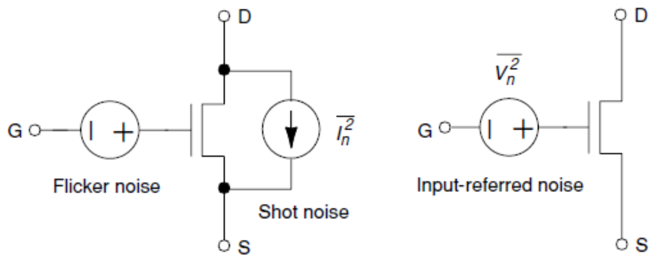


Figure 3.1: MOSFET input noise model at low frequency

3.2 Noise in Commercially available 130nm CMOS process

The classifier is implemented in 130nm IBM process. Hence it makes sense to do noise analysis specific to this process. In particular it is essential to find theoretical value of the minimum input signal that can be measured using the minimum size transistor in this process. Also how the noise affects the output and what SNR levels provides acceptable results should be explored. Before that we need to find K_w and K_f from Eq.3.7.

3.2.1 Measurement of K_w

To measure $K_w = 2ktU_T/\kappa^2$ we need the value of κ which is the subthreshold exponential factor. The drain current in a subthreshold MOSFET is given by [4, chapter 3]

$$I_D = I_0 e^{(\kappa V_g - V_s)/U_T}(1 - e^{-V_{ds}/U_T}) \tag{3.9}$$

V_{ds} is the drain to source voltage. For small values of V_{ds} drain current is a linear function of V_{ds}. At $V_{ds} > 4U_T$ current goes in saturation. κ is given as $\kappa = C_{ox}/(C_{ox} + C_d)$. Here C_{ox} is oxide capacitance and C_d is incremental capacitance in depletion layer. From the simulation of a MOSFET with fixed V_{ds} and varying V_g in cadence we can find out the effective value of κ. Eq.3.9 can be modified as

$$\begin{aligned}
I_D &= I_0 e^{(\kappa V_g - V_s)/U_T}(1 - e^{-V_{ds}/U_T}) \\
I_D &= I_0 e^{(\kappa V_g - V_s)/U_T} \times Const. \\
I_D &= C \times e^{(\kappa V_g - V_s)/U_T} \\
\frac{dI_D}{dV_g} &= C \times e^{(\kappa V_g - V_s)/U_T} \frac{\kappa}{U_T}
\end{aligned}$$

$$\frac{dI_D}{dV_g} = I_D \frac{\kappa}{U_T}$$
$$\kappa = \frac{U_T \frac{dI_D}{dV_g}}{I_D} \tag{3.10}$$

In CADENCE a minimum size (W = 160nm, L = 120nm) lpnfet is set up as shown in Fig.3.2 The gate voltage is varied V_g and drain current is measured. The data is then exported to MATLAB to carryout calculation of Eq.3.10. Using $U_T = 25mV$, $V_{DD} = 1V$, $V_{ds} = 1V$ the results are shown in Fig.3.3. For $V_g > 500mV$ the MOSFET goes into inversion. The average value of κ can be approximated from the subthreshold part of Fig.3.2(c) as $\kappa = 0.68$. Then K_w is evaluated as

$$K_w = \frac{2kTU_T}{\kappa^2}$$
$$K_w = \frac{2 \times 1.38 \times 10^{-23} \times 300 \times 25 \times 10^{-3}}{0.68^2}$$
$$K_w = 4.48 \times 10^{-22} \qquad (V^2A/Hz) \tag{3.11}$$

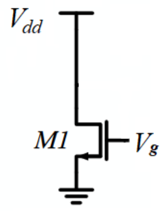
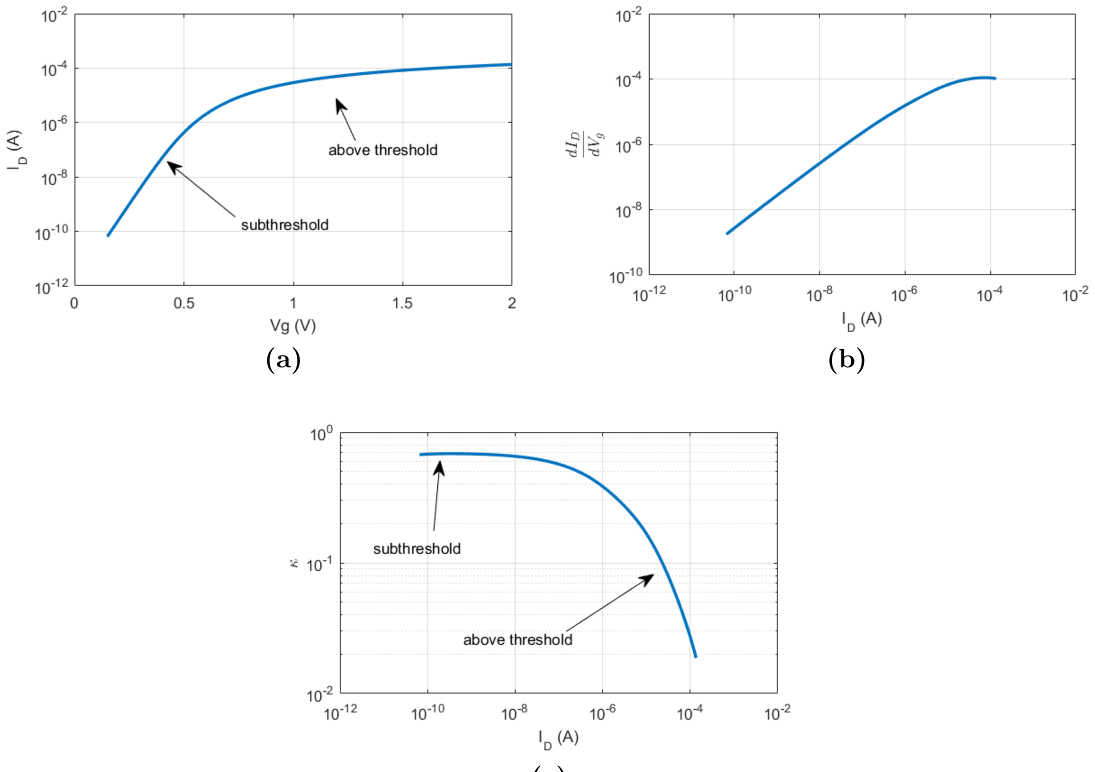


Figure 3.2: Circuit used to find κ

3.2.2 Measurement of K_f

K_f is a process dependent fit constant [6]. Using the noise analysis output from CADENCE we can find the fit constant. The circuit configuration in Fig.3.4 is



(a) **(b)** **(c)**

Figure 3.3: Calculation of κ. For lpnfet $V_{th0} = 553.3mV$.

used and V_g is set such that the device is in subthreshold region. The process simulation model uses a flicker noise exponent factor $n = 0.95$. Using this exponent factor the fit constant for flicker noise is calculated. From the noise simulation output noise current and the calculated K_f is found as shown in Fig.3.5. Although the output noise consists of both flicker and white noise, the lower frequency noise is mostly dominated by flicker noise. So treating the output noise as only consisting of flicker noise is sufficient to calculate K_f.

$$\bar{I}_f^2 = \frac{K_f}{A}\frac{1}{f^n}(\kappa\frac{I_d}{U_T})^2$$
$$K_f = \bar{I}_f^2 A f^n/(\kappa\frac{I_d}{U_T})^2 \tag{3.12}$$

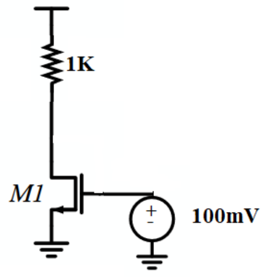
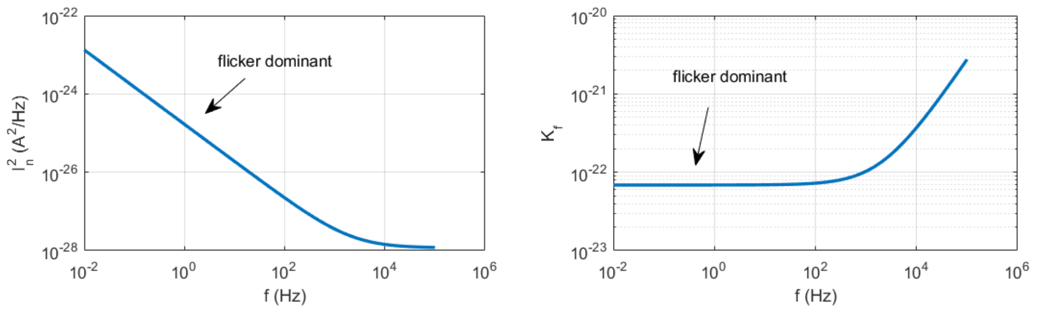


Figure 3.4: Circuit used to calculate flicker noise

From Fig.3.5(b) the value of K_f is constant in flicker noise dominant part. The



(a) (b)

Figure 3.5: Output noise current from simulation of a lpnfet $W = 160n$, $L = 120n$, $I_d = 250pA$.

fit constant is calculated as $K_f = 6.9 \times 10^{-23}$. To see how well the simulation noise current fits the prediction by our model the total noise current calculated. The total noise current is the sum of white noise and flicker noise as given by Eq.3.13. The total noise calculated from model equations seems to agree well with simulation output which is shown in Fig.3.6

$$\bar{I}_o^2 = \frac{K_f}{A}\frac{1}{f^n}(\kappa\frac{I_d}{U_T})^2 + 2qI_d \tag{3.13}$$

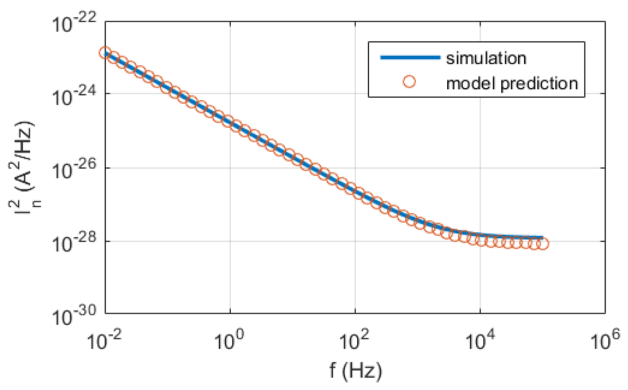


Figure 3.6: Total input referred noise for lpnfet ($W = 160n$, $L = 120n$, $I_d = 250pA$) simulation vs model prediction.

3.3 Minimum detectable signal

Low power circuits operate on very low signal levels. This makes the input signals to the circuit susceptible to be corrupted by noise. This sets a minimum limit to the input signal that can be reliably processed by the circuit. The noise is superimposed on the input signal. Hence if the signal level in comparable to the noise level then the signal will be completely buried inside the noise. It is necessary for the signal to have reasonable amplitude to be distinguishable from the noise. For a single MOSFET operating in subthreshold regime from Eq.3.7 setting $I_D = 1nA, f_l = 0.01Hz, f_h = 1GHzA = WL = 160nm \times 120nm$, the input referred noise is

$$\bar{v}_n^2 = \frac{4.48 \times 10^{-22} \times (f_h - f_l)}{I_D} + \frac{6.3928 \times 10^{-24}}{A} \ln(\frac{f_h}{f_l})$$

$$\bar{v}_n^2 = 448 \qquad \mu(V^2)$$

$$\bar{v}_n = 21.2 \qquad mV$$

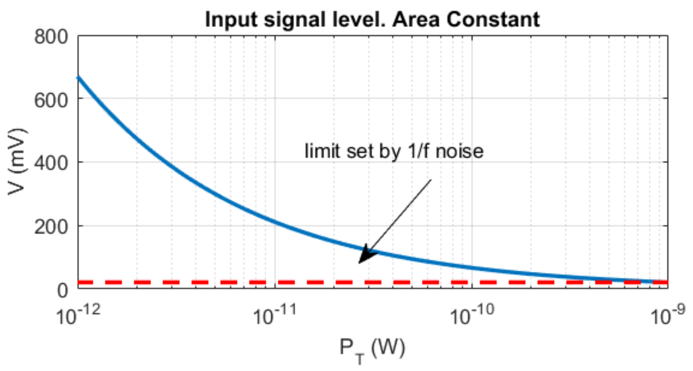
If we take the square root of the input noise power we get the noise amplitude level. This means we cannot measure anything below 21.2mV at the input at low frequencies. By same argument we can say at the output the current level is

$$\bar{i}_n^2 = \bar{v}_n^2 g_m^2$$
$$\bar{i}_n^2 = \bar{v}_n^2 (\frac{\kappa I_D}{U_T})^2$$
$$\bar{i}_n^2 = 3.31 \times 10^{-19} \qquad (A^2)$$
$$\bar{i}_n = 0.5 \qquad nA$$

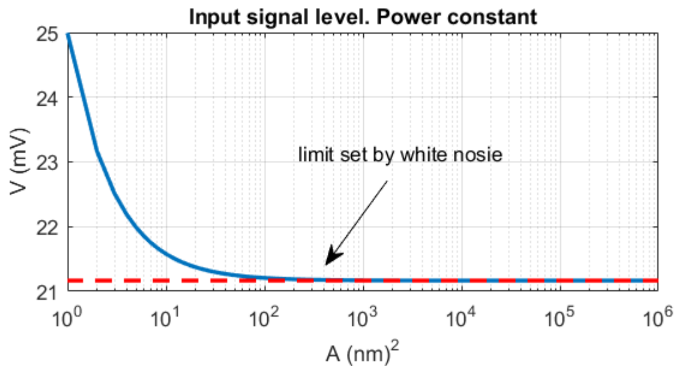
So for a minimum size lpnfet, at bias current level of 1nA, we need more than 0.5nA of current to treat it as signal. Eq.3.7 can also be written as

$$\bar{v}_n^2 = \frac{K_w}{I_D}\Delta f + \frac{K_f}{A}\ln(\frac{f_h}{f_l})$$
$$\bar{v}_n^2 = \frac{K_w V_{DD}}{V_{DD} I_D}\Delta f + \frac{K_f}{A}\ln(\frac{f_h}{f_l})$$
$$\bar{v}_n^2 = \frac{K_w V_{DD}}{P_T}\Delta f + \frac{K_f}{A}\ln(\frac{f_h}{f_l}) \tag{3.14}$$

V_{DD} is the supply voltage. Eq.3.14 relates the input referred noise with the total power and area of of MOSFET. Increasing power will reduce the white noise but the flicker noise will be unaffected. Increasing the area will reduce the flicker noise but the white noise will be unaffected. From this equation limit of minimum detectable signal at input can be found as function of power and area. Such a limit is shown in Fig.3.7. First power is varied from 1pW to 1nW with $V_{DD} = 1V$ keeping the area constant at $WL = 160nm \times 120nm$. As the area is fixed the flicker noise is constant. Hence even if the white noise part decreases, total noise cannot go below the fixed flicker noise. This is shown in Fig.3.7(a). Also lower portion under the curve is labeled as unreachable. This is because any signal level on those region will be buried under the noise. So the minimum measurable signal level should be above the curve. A similar analysis is shown in Fig.3.7(b) when power is kept constant at 1nW with $V_{DD} = 1V$ and area is varied from $1nm^2$ tot $1\mu m^2$. This time the minimum noise is set by the fixed power. To minimize the noise from a MOSFET both power and area needs to be optimized to obtain minimum noise.



(a)



(b)

Figure 3.7: Minimum detectable signal level at input at low frequencies as power and area is varied (a) Area is constant at minimum size (W=160nm, L=120nm) (b) Power is set at 1nW.

3.4 Noise in analog multiplier circuit

The multiplier is the basic building block of the analog classifier. It takes current I_B as input and produces a current I_o as follows.

$$I_o = m \times I_B \tag{3.15}$$

where m is the coefficient. The multiplier is implemented by a differential transconductance amplifier.

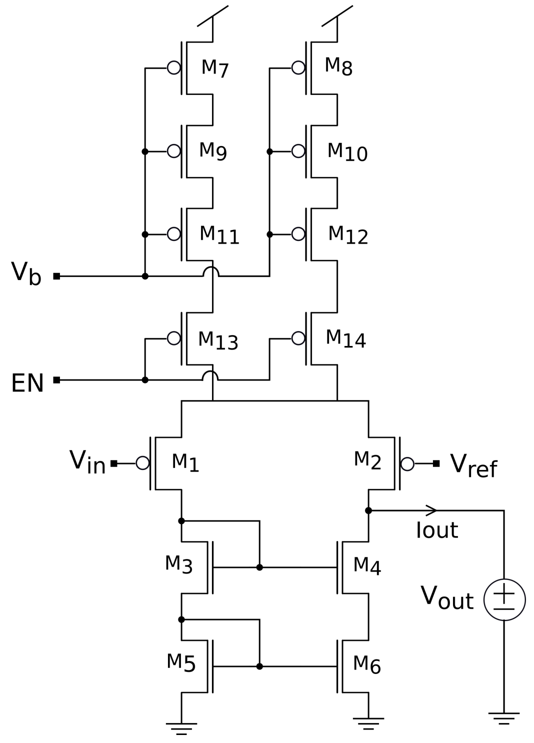


Figure 3.8: Differential transconductance amplifier used as multiplier circuit

3.4.1 Multiplier circuit

The circuit is shown in Fig.3.8. The tail current or the bias current is produced by pMOS current source using voltage V_b. EN acts as an enabler that connects or disconnects the bias current sources from the rest of the circuit. There are two current sources in parallel pushing tail current for the differential pair. If one source produces I_T current then two parallel sources produce bias current $I_B = 2I_T$. The current mirror is made of nMOS cascode current mirror. The current mirror performs subtraction of two current from the two legs of the differential pair. The current mirror draws equal current in both legs. So any imbalances in currents in M_1 and M_2 goes to output. This is how the current mirror performs subtraction. The output current is given by

$$I_o = I_B \tanh[\frac{\kappa}{2U_T}(V_{in} - V_{ref})] \tag{3.16}$$

If we compare Eq.3.16 with Eq.3.15 using I_T as multiplicand then the coefficient m from the circuit is

$$m = \tanh[\frac{\kappa}{2U_T}(V_{in} - V_{ref})] \tag{3.17}$$

The voltage difference $(V_{in} - V_{ref})$ is used to set the magnitude of the multiplier m. This voltage difference has a $tanh(.)$ relationship with the multiplier. Fig.3.9 shows how the function varies with the argument. Its value is limited to [-1, +1]. Hence the multiplier value m is also limited to [-1, +1] for bias current I_B. If we assume I current goes through M_5 then M_6 and M_1 also carries I. M_2 then carries $(I + mI_B)$. But the sum of currents through M_1 and M_2 has to be I_B. So $I + I + mI_B = I_B$ which means $I = (1 - m)I_B/2$. Hence M_1 carries $(1 - m)I_B/2$ and M_2 carries $(1 + m)I_B/2$.

3.4.2 Noise in the multiplier

To find the total output noise current we have to use the small signal circuit including the noise generator for each transistor. All the voltage/current inputs are ac short/open respectively. The resulting circuit is shown in Fig.3.10. The

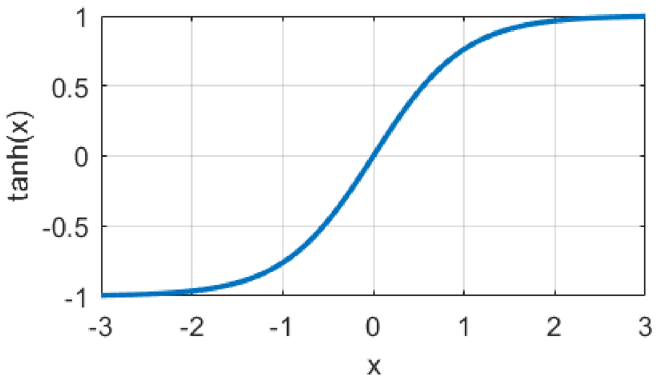


Figure 3.9: Shape of tanh(x) function

transconduntance $g_s = g_m + g_{mb}$ includes the body effect. The next step is to

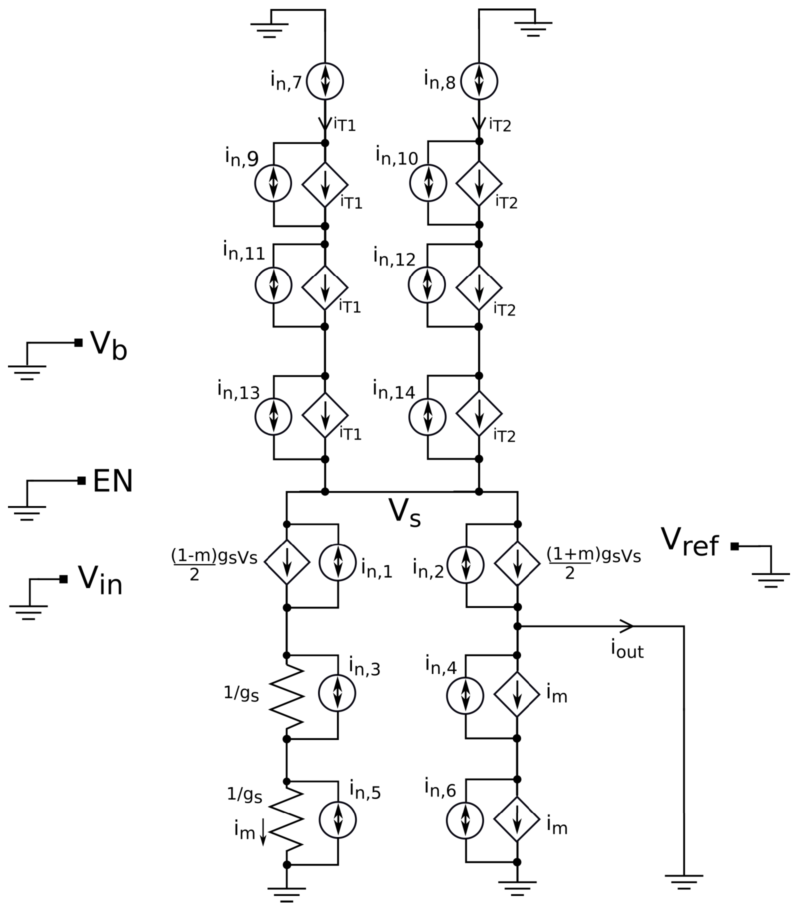


Figure 3.10: Small-signal noise circuit of the multiplier

find the gain of each noise generator, i_k, to the output of the multiplier, $\alpha_k(f)$, while setting all other noise sources to zero. Since they are uncorrelated we can use superposition to add up all the individual noise. The total output noise will be given by

$$\overline{i_{out}^2(f)} = \sum_{\forall k} |\alpha_k(f)|^2 \overline{i_k^2(f)} \tag{3.18}$$

We can simplify the small signal circuit further as shown in Fig.3.11. The g_s generators of M_1 and M_2 are replaced with $1/g_s$ resistors and dependent current sources. For simplicity the subscript n in noise currents is dropped in calculation.

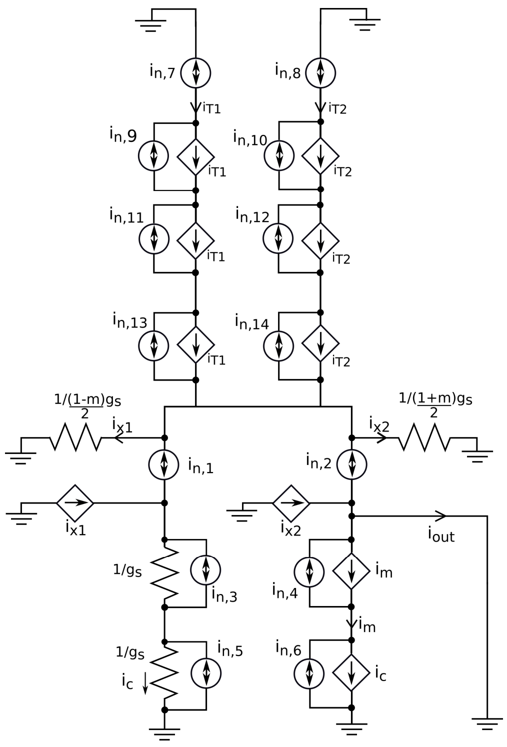


Figure 3.11: Simplified small-signal noise circuit of the multiplier

Gain for i_7 and i_8: The noise current for i_7 goes through $2/(1-m)g_s$ and $2/(1+m)g_s$ resistors and divides into $(1-m)i_7/2$ and $(1+m)i_7/2$ respectively. $i_{x1} = (1-m)i_7/2$ which generates $i_m = i_c = (1-m)i_7/2$. $i_{x2} = (1+m)i_7/2$.

Total output current is $i_{out} = i_{x2} - i_m = (1+m)i_7/2 - (1-m)i_7/2 = mi_7$. The gain for i_7 is $\alpha_7 = m$. Similarly the gain for i_8 is $\alpha_8 = m$.

Gain for i_9 to i_{14}: The noise currents for i_9 to i_{14} cannot flow to the output because i_7 and i_8 will be open circuited. Hence the gain for these currents is zero.

Gain for i_1: i_1 draws current through $2/(1-m)g_s$ and $2/(1+m)g_s$ and divides into $i_{x1} = -(1-m)i_1/2$ and $i_{x2} = -(1+m)i_1/2$ respectively. $i_m = i_c = i_1 + i_{x1} = (1+m)i_1/2$. Total output current is $i_{out} = i_{x2} - i_m = -(1+m)i_1$. The gain for i_1 is $\alpha_1 = -(1+m)$.

Gain for i_2: For i_2, $i_{x1} = -(1-m)i_2/2$ and $i_{x2} = -(1+m)i_2/2$. $i_m = i_c = i_{x1} = -(1-m)i_2/2$. Total output noise current $i_{out} = i_{x2} + i_2 - i_m = (1+m)i_2$. The gain for i_2 is $\alpha_2 = (1-m)$.

Gain for i_3 and i_4: i_3 only circulates in M_3. So i_{x1}, i_{x2}, i_m are all zero. The gain for i_3 is zero. Similarly for i_4 the currents i_{x1}, i_{x2}, i_m are all zero. The gain for i_4 is also zero.

Gain for i_5: i_5 circulates through M_5. $i_m = i_c = -i_5$. Total output noise current $i_{out} = -i_m = i_5$. The gain for i_5 is $\alpha_5 = 1$.

Gain for i_6: i_6 flows through M_4 to the output. $i_m = i_6$. Total output noise current $i_{out} = -i_m = -i_6$. Gain for i_6 is $\alpha_6 = -1$.

Total noise output noise current: If we assume that the output noise is dominated by thermal noise in the subthreshold region then the total noise current is calculated using Eq.3.18.

$$\begin{aligned}
\overline{i_{out}^2(f)} &= \alpha_7^2\overline{i_7^2(f)} + \alpha_8^2\overline{i_8^2(f)} + \alpha_1^2\overline{i_1^2(f)} + \alpha_2^2\overline{i_2^2(f)} + \alpha_5^2\overline{i_5^2(f)} + \alpha_6^2\overline{i_6^2(f)} \\
\overline{i_{out}^2(f)} &= \alpha_7^2 \times 2q\frac{I_B}{2} + \alpha_8^2 \times 2q\frac{I_B}{2} \\
&\quad + \alpha_1^2 \times 2q\frac{(1-m)}{2}I_B + \alpha_2^2 \times 2q\frac{(1+m)}{2}I_B \\
&\quad + \alpha_5^2 \times 2q\frac{(1-m)}{2}I_B + \alpha_6^2 \times 2q\frac{(1-m)}{2}I_B
\end{aligned}$$

$$
\begin{aligned}
\overline{i^2_{out}(f)} &= [m^2\frac{1}{2} + m^2\frac{1}{2} \\
&\quad + (1+m)^2(1-m)\frac{1}{2} + (1-m)^2(1+m)\frac{1}{2} \\
&\quad + (1-m)\frac{1}{2} + (1-m)\frac{1}{2}]2qI_B \\
\overline{i^2_{out}(f)} &= [m^2 + 2(1-m^2)\frac{1}{2} + (1-m)]2qI_B \\
\overline{i^2_{out}(f)} &= (2-m)2qI_B \qquad (3.19)
\end{aligned}
$$

This equation shows when $m = -1$ noise is maximum and minimum when $m = +1$. This conclusion can also be reached by intuition. When $m = -1$, all of the current is steered through M_1 to M_3, M_5 which is also copied to M_4, M_6. Hence noise current is maximum. But when $m = +1$, all of the current is steered to output and there is no current in M_3 to M_6. so the noise current is minimum. If the noise bandwidth is Δf then output noise current is

$$
\begin{aligned}
\overline{i^2_{out}} &= \int_0^\infty (2-m)2qI_B df \\
\overline{i^2_{out}} &= (2-m)2qI_B\Delta f \qquad (3.20)
\end{aligned}
$$

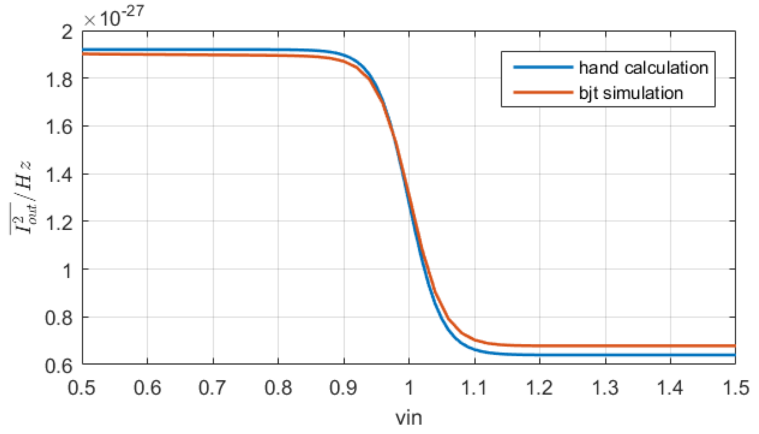


Figure 3.12: Output noise current from multiplier made of ideal BJT and from hand analysis

The cadence simulation of the multiplier shows presence of 1/f noise. To verify how well this analysis is able to predict output noise current, we simulate for output noise current of a multiplier made of ideal BJTs. This only exhibits thermal noise and behaves like MOSFET in subthreshold with $\kappa = 1$. The result is shown in Fig.3.12. The hand analysis result and the simulation output matches quite accurately.

3.4.3 Signal to noise ratio

The output current power is $I_o^2 = m^2 I_B^2$. Hence the signal to noise ratio (SNR) is

$$SNR = \frac{I_o^2}{\overline{i_{out}^2}}$$

$$SNR = \frac{m^2 I_B^2}{(2-m)2qI_B\Delta f}$$

$$SNR = \frac{m^2 I_B}{(2-m)2q\Delta f} \tag{3.21}$$

$$SNR = \frac{m^2 P_T}{(2-m)V_{DD}2q\Delta f} \tag{3.22}$$

Here $P_T = V_{DD} I_B$ is the power consumption in the multiplier. Few observation can be made from the SNR equation. The SNR depends on the multiplier operating point m. Improved SNR requires higher power consumption P_T. Also higher SNR means slower operation.

3.4.4 Fundamental Limit of Multiplication

From the equation of SNR a fundamental limit of the multiplier operation can be obtained that displays how much one quantity is limited when other quantity is chosen. Eq.3.21 can be rearranged as follows.

$$\frac{SNR \times \Delta f}{I_B} = \frac{m^2}{(2-m)2q} \tag{3.23}$$

m assumes values in the range $-1 \leq m \leq +1$. Hence the left side of Eq.3.23 is expressed in terms of physical constant. This lets us calculate a limit of how much can be achieved from a multiplier. If we translate noise bandwidth into time period of operation $\Delta f = 1/\tau$ then $P_T/\Delta f = P_T\tau = J_m$ represents energy consumed in joules for one operation of multiplication. Eq.3.22 can be written as.

$$J_m = \frac{(2-m)2q}{m^2} SNR \times V_{DD} \tag{3.24}$$

This equation has one interesting outcome at $m = 0$. The energy required to achieve any value of SNR is unbounded. However, this makes sense because at $m = 0$ there is no output from the multiplier. Hence, there is no value of energy that can produce any output signal. For a fixed value of m Eq.3.24 shows a linear relationship with SNR and energy per multiplication.

3.4.5 Effective number of bits and Energy per multiplication

If we were to convert the analog current into digital domain then the number of bits required for a given SNR is

$$b = \frac{SNR - 1.76}{6.02} \tag{3.25}$$

where all the values are given in dB. Replacing the SNR with Eq.3.22 we obtain energy required for a given number of bits.

$$J_m = \frac{(2-m)2qV_{DD}}{m^2} 10^{(6.02b+1.76)/10} \tag{3.26}$$

We can see that the relationship of number of bits with the energy per multiplication is exponential. If $b = 0$ there will still be some energy that will be spent. Fig.3.13 shows the energy variation with number of bits for $m = -1$.

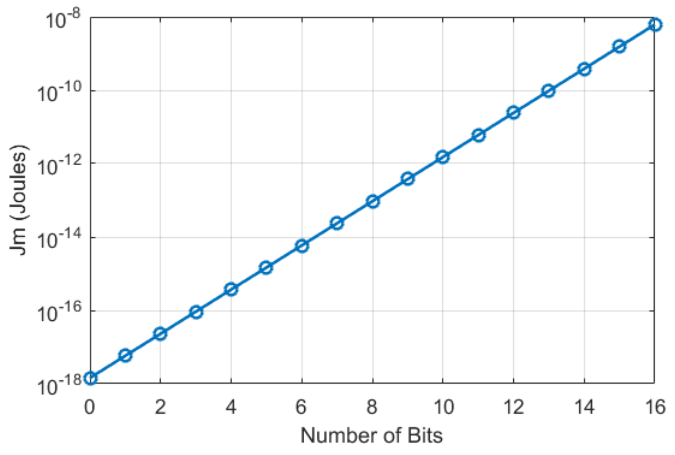


Figure 3.13: Energy consumption with number of bits.

3.4.6 Equivalent relationship in Digital system

For most digital systems the power and area are proportional to the number of bits. But for some digital systems the power and area scales as a polynomial function of the number bits. Multiplication have power and area cost that scales as square of the number of bits. For a given SNR the equivalent number of bits b required for a digital computation, a conversion from SNR to equivalent bit is necessary. One is given by Shanon-Hartley equation [7] for data communication.

$$b = \frac{1}{2}\log_2(1 + SNR) \qquad (bits/sample) \tag{3.27}$$

Another is given from Analog to Digital Conversion (ADC) theory given by Eq.3.25. This is more appropriate for use. A single transistor with width W, length L, operating with supply voltage V_{DD}, clock frequency f, load capacitance C consumes dynamic power fCV_{DD}^2 [8]. Then the resource precision equation for digital multiplication is given by

$$P_D = L_p[\frac{SNR - 1.76}{6.02}]^2 \tag{3.28}$$

$$A_D = L_a[\frac{SNR - 1.76}{6.02}]^2 \tag{3.29}$$

where $L_p = fCV_{DD}^2$ and $L_a = WL$. A comparison of power consumption and area consumption with respect to SNR for digital and analog scenario can be drawn. If we include 1/f noise in the total output noise current then

$$\overline{i_{out}^2(f)} = (2-m)2qI_B + (2-m)\frac{K_f}{A}\frac{1}{f}(\frac{\kappa I_B}{U_T})^2 \qquad (A^2/Hz) \tag{3.30}$$

Using this to find analog SNR and Eq.3.28 ,3.29 for digital SNR, power vs. SNR and area vs SNR can be compared similar to what was presented in [9]. Such a comparison is shown in Fig.3.14. For a fixed area the 1/f noise is fixed. With increase of power consumption white noise decreases but the total noise cannot decrease below the fixed 1/f noise. This is why for fixed area consumption analog SNR cannot improve beyond what is set by the 1/f noise limit. For low values of SNR analog multiplier has better power consumption than digital counterpart. Similar situation is observed when the power is fixed. This time the fixed power consumption makes white noise constant. As the area is increased for analog case 1/f noise decreases but the total noise cannot decrease below the fixed white noise. Hence there is an upper limit on the SNR. In the case of area consumption here also the analog circuit performs better. The digital circuit take more area than the analog counterpart for the same SNR.

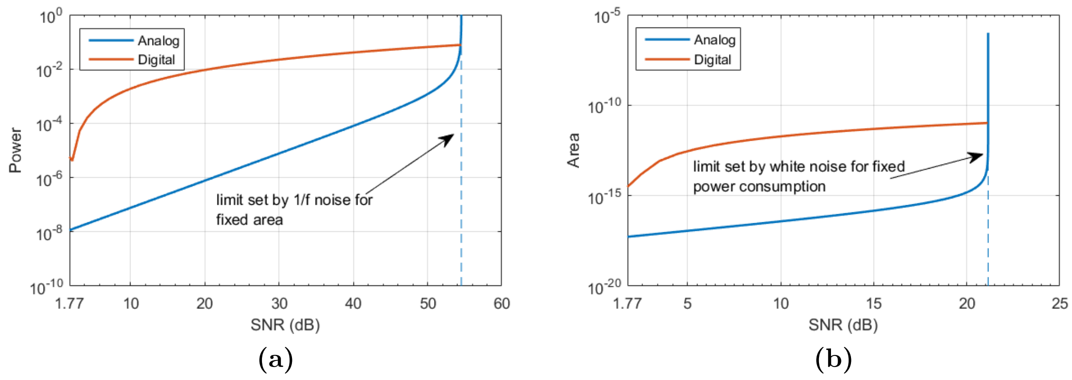


Figure 3.14: comparison of precision for digital and analog multiplication. m is varied, $f_h = 1GHz$, $I_B = 1nA$, $V_{DD} = 1V$, $C = 1pF$, $W = 1\mu m$, $L = 1\mu m$, $m = 0.5$ (a) area is fixed at $W = 1\mu m$, $L = 1\mu m$ in analog (b) power consumption is fixed at $1\mu W$. in analog

3.4.7 Multiplication in Classifier

In the classifier the actual output current is the sum of several multiplier output current as follows.

$$I_{out} = m_1 I_1 + m_2 I_2 + \cdots + m_n I_n$$
$$I_{out} = \sum_{i=1}^{n} m_i I_i \tag{3.31}$$

To implement this, it may be intuitive to connect the output of n multiplier circuit cells to get the sum of the outputs. However it is sufficient to use only

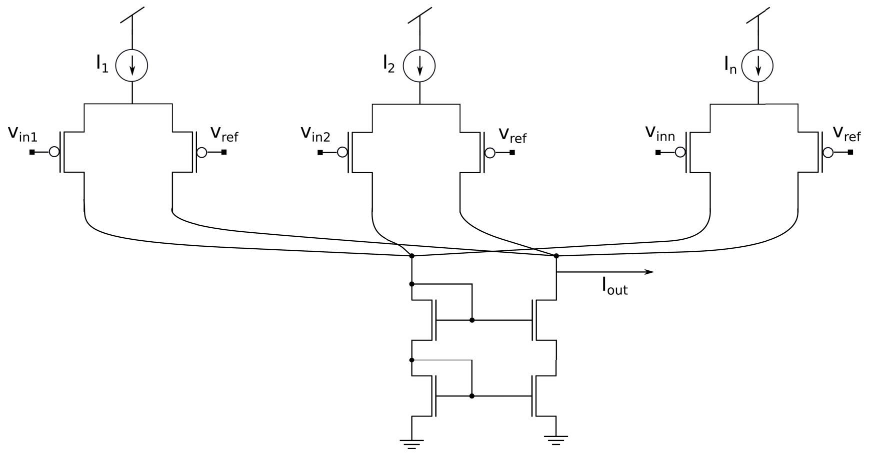


Figure 3.15: Multiplier current summing in classifier.

one current mirror to get the sum of multiplier currents as shown in Fig.3.15 The output noise current in this case is calculated the same way as single multiplier cell. The output noise current for this case is as follows.

$$\begin{aligned} \overline{i_{out}^2} &= \sum_{i=1}^{n} \overline{i_{m_i}^2} \\ \overline{i_{out}^2} &= \sum_{i=1}^{n} (2 - m_i) 2qI_i\Delta f \end{aligned} \tag{3.32}$$

The SNR for the classifier is

$$\begin{aligned} SNR &= \frac{(\sum_{i=1}^{n} m_i I_i)^2}{\sum_{i=1}^{n}(2 - m_i)2qI_i\Delta f} \\ SNR &= \frac{(\sum_{i=1}^{n} m_i P_i)^2}{\sum_{i=1}^{n}(2 - m_i)2qV_{DD}P_i\Delta f} \end{aligned} \tag{3.33}$$

Here P_i is power consumption for each multiplier cell. To make this equation manageable it is useful to define average multiplier $\overline{m}$ as follows.

$$\overline{m} = \frac{\sum_{i=1}^{n} m_i I_i}{\sum_{i=1}^{n} I_i}$$

$$
\begin{aligned}
\overline{m} &= \frac{\sum_{i=1}^{n} m_i V_{DD} I_i}{\sum_{i=1}^{n} V_{DD} I_i} \\
\overline{m} &= \frac{\sum_{i=1}^{n} m_i P_i}{\sum_{i=1}^{n} P_i} \\
\overline{m} &= \frac{\sum_{i=1}^{n} m_i P_i}{P_T} \qquad (3.34)
\end{aligned}
$$

Here P_T is the total power consumed in the classifier. Using this we can simplify the SNR for the classifier from Eq.3.33

$$
\begin{aligned}
SNR &= \frac{(\sum_{i=1}^{n} m_i P_i)^2}{\sum_{i=1}^{n}(2P_i - m_i P_i) 2qV_{DD}\Delta f} \\
SNR &= \frac{(\overline{m} P_T)^2}{(2P_T - \overline{m} P_T) 2qV_{DD}\Delta f} \\
SNR &= \frac{\overline{m}^2 P_T}{(2 - \overline{m}) 2qV_{DD}\Delta f} \qquad (3.35)
\end{aligned}
$$

This equation is much more tractable and has similar form as Eq.3.22.

3.4.8 Energy per multiply accumulate

The multiply-accumulate (MAC) is the operation defined by multiplication of two quantities and adding the product to an accumulator.

$$
\begin{aligned}
a &\leftarrow a + b \times c \\
a &\leftarrow a \times 1 + b \times c \qquad (3.36)
\end{aligned}
$$

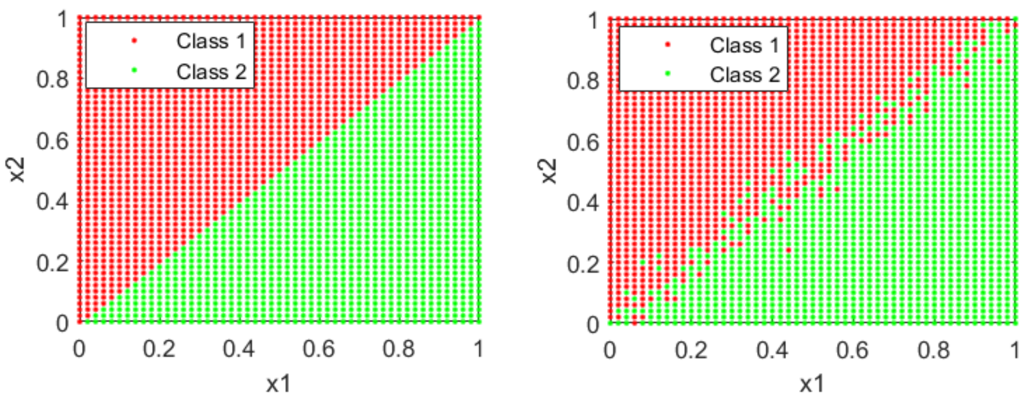
The addition process does not take any extra energy because it can be done by combining two wires carrying current a and $b \times c$ and Kirchoff's law takes care of the addition. In this case the MAC operation can be thought of as two multiplication operation. One with multiplier 1, bias current a and other with multiplier c, bias current b. As $P_T/\Delta f = P_T \tau = J_m$ is total energy, using Eq.3.35

we can compute energy per MAC as

$$J_{MAC} = \frac{(2-\overline{m})2q}{\overline{m}^2} SNR \times V_{DD} \tag{3.37}$$

3.4.9 Effect of noise in classifier decision making

The classifier operates on comparing two currents. If current from one class is greater than or equal to the other then the first class wins. However when two currents are equal, noise in the current randomly makes the currents deviate from the actual values. This makes some wrong decisions at the decision boundary. The effect can be simulated in MATLAB for two variable simple binary classification. When there is no noise the class boundary is distinct and clean as shown in Fig.3.16(a). When some Gaussian noise is introduced at the multiplier output there are some wrong decisions and class boundary is not clearly identified as shown in Fig.3.16. By increasing the SNR the number of wrong decisions can be reduced.



(a) (b)

Figure 3.16: Classification result (a) without noise (b) with noise. There are some misclassification at the boundary between two class.

Chapter 4

Testing Process of Linear Classifier

The classifier works by taking current as inputs and based on the weights stored in the memory the classifier classifies the input into separable classes. Let the input currents be $i_1,\ i_2,\ ...\ ,\ i_n$. And the weights for m classes be $[w_{11}, w_{12}, ..., w_{1n}]$, $[w_{21}, w_{22}, ..., w_{2n}]$, ... , $[w_{m1}, w_{m2}, ..., w_{mn}]$. The output currents for each of the classes from the classifier is given by.

$$\begin{aligned} I_1 &= w_{11} * i_1 + w_{12} * i_2 + ... + w_{1n} * i_n \\ I_2 &= w_{21} * i_1 + w_{22} * i_2 + ... + w_{2n} * i_n \\ &\vdots \\ I_m &= w_{m1} * i_1 + w_{m2} * i_2 + ... + w_{mn} * i_n \end{aligned} \tag{4.1}$$

The output class is identified by looking at the highest output current from the array of output currents $[I_1, I_2, ..., I_m]$. The winner take all logic takes care of finding out which output class has highest current. Fig.4.1 shows the topology of the classifier. It has four rows for input current and three columns for three

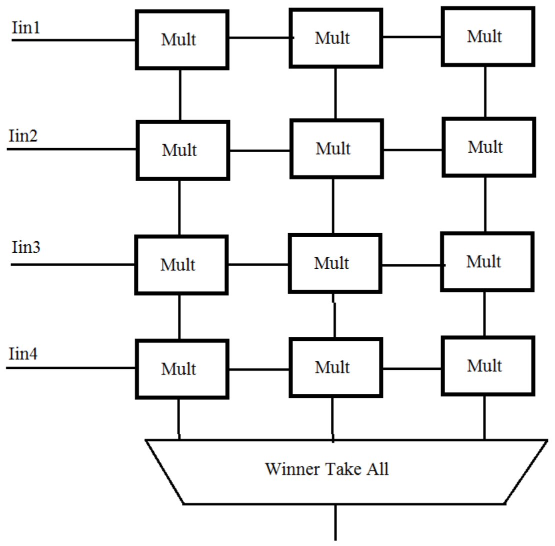


Figure 4.1: Classifier topology in the chip

classes.

Now for a simple case of three output classes and two input currents the classifier can be described by two dimensional equation of lines. For example for the linear classifier equation below, the classes are separated by three lines in cartesian coordinate system as shown in Fig.4.2.

$$\begin{aligned} I_1 &= 0.25 * i_1 + 0.6 * i_2 + 0.3 \\ I_2 &= -0.1 * i_1 + 0.3 * i_2 + 0.6 \\ I_3 &= 0.3 * i_1 + 0.3 * i_2 + 0.4 \end{aligned} \tag{4.2}$$

Hence if we take two input of the classifier to vary and set appropriate weights then we should see an output response similar to Fig.4.2 which have its classes separated by lines. This chapter will describe how the classifier chip is set up with external circuits to apply input signals, measure outputs using LabView and how the results extracted.

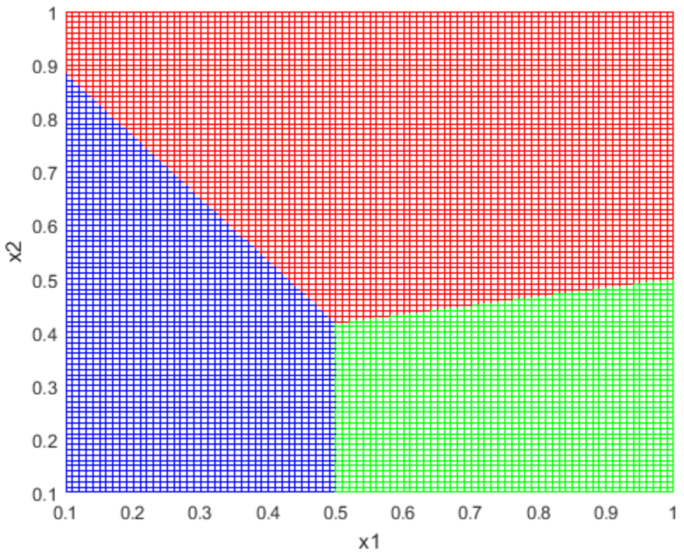


Figure 4.2: The color represents the regions of classes. Red represents [1,0,0], blue is [0,1,0], and green is [0,0,1].

4.1 Current measuring circuit

The currents out of the multipliers I_{in} are on the order of pA range. So to reliably measure the current the circuit in Fig.4.3 is used. A negative feedback configuration is used to ensure V_{ref} voltage at the input current node. The current provides a voltage drop across $10G\Omega$ resistor. The second stage provides a gain of 10 on that voltage.

$$V_{o1} = 10G\Omega * I_{in} + V_{ref} \tag{4.3}$$

$$V_{out} - V_{ref} * \frac{2}{20} = V_{o1} - V_{ref} \tag{4.4}$$

$$V_{out} = V_{ref} + 10 * 10G\Omega * I_{in} \tag{4.5}$$

Hence for $V_{ref} = 1V$, $V_{out} = 1.1V$ for $1pA$ current, $V_{out} = 1.2V$ for $2pA$ current and so on. LMC6482IN CMOS dual rail to rail input and output operational

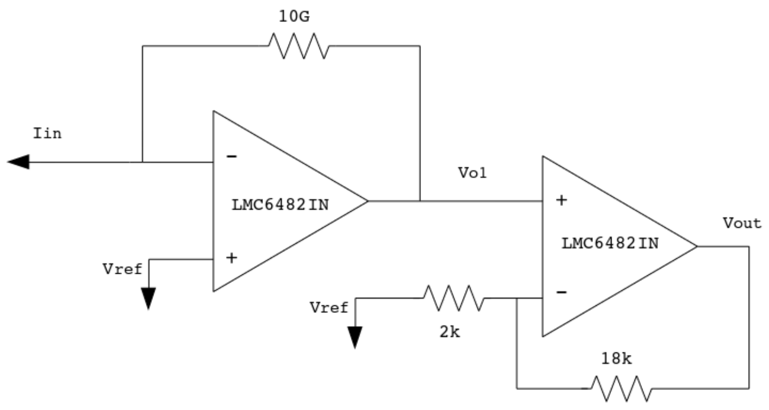


Figure 4.3: Circuit for pA current measurement

amplifiers has been used in this circuit. This circuit will be referred to as transimpedance amplifier (TIA).

4.2 TIA Offset

Here it is shown how the offsets of the amplifier used for pA current measurement affects the output results. It is easy to calculate the effects of offsets using superposition. From Fig.4.4 for the first stage output voltage without offset is $V_{o1} = R_{10G\Omega} I_{in} + V_{ref}$. Output with only offset is $V_{o1} = V_{OS} + I_B R_{10G\Omega}$. Hence the output with the effect of offset is

$$V_{o1} = R_{10G\Omega} I_{in} + V_{ref} + V_{OS} + I_B R_{10G\Omega} \tag{4.6}$$

For the second stage the output without offset is $V_{out} = (1 + \frac{R_{18k\Omega}}{R_{2k\Omega}})V_{o1} - \frac{R_{18k\Omega}}{R_{2k\Omega}} V_{ref}$. Output with only offset is $V_{out} = (1 + \frac{R_{18k\Omega}}{R_{2k\Omega}})V_{OS} + I_B R_{18k\Omega}$. Output with the effect of offset is

$$V_{out} = (1 + \frac{R_{18k\Omega}}{R_{2k\Omega}})V_{o1} - \frac{R_{18k\Omega}}{R_{2k\Omega}} V_{ref} + (1 + \frac{R_{18k\Omega}}{R_{2k\Omega}})V_{OS} + I_B R_{18k\Omega} \tag{4.7}$$

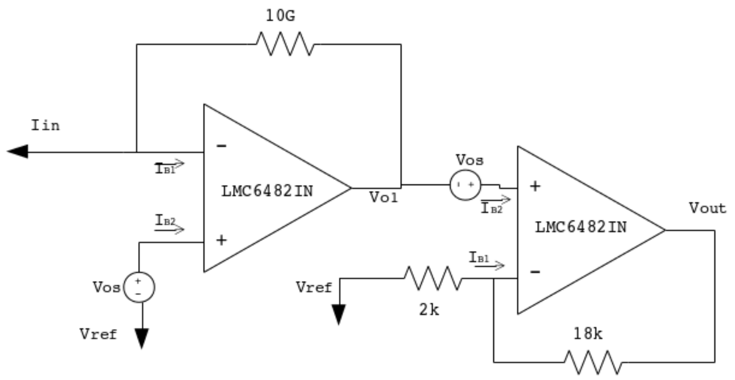


Figure 4.4: Offset Calculation for pA current measuring circuit.

Combining Eq.4.6 and Eq.4.7 we have the final output from the circuit as

$$\begin{aligned} V_{out} &= (1+\frac{R_{18k\Omega}}{R_{2k\Omega}})I_B R_{10G\Omega} + V_{ref} \\ &+ (1+\frac{R_{18k\Omega}}{R_{2k\Omega}})V_{OS} + (1+\frac{R_{18k\Omega}}{R_{2k\Omega}})V_{OS} \\ &+ (1+\frac{R_{18k\Omega}}{R_{2k\Omega}})I_B R_{10G\Omega} + I_B R_{18k\Omega} \end{aligned} \tag{4.8}$$

From the data sheet of LMC6482 the typical offsets are $V_{OS} = 0.11mV$ and $I_B = 0.02pA$ at 25^oC. This amounts to a offset output of around $4.2mV$. So if $I_{in} = 0$ and $V_{ref} = 1V$ the output voltage will be $V_{out} = V_{ref} + 4.2mV = 1.0042V$. This amounts to equivalent input current of around $0.42fA$ which is three orders of magnitude lower than the current we want to measure. (Using equation $V_{out} = V_{ref} + 10 * 10G\Omega * I_{in}$)

4.3 Transconductance Board

A transconductance board is used to convert analog signal into currents which is used to supply the input currents to the classifier chip. Fig. shows the board used in the measurement. (I don't have the circuit diagram for this one.)

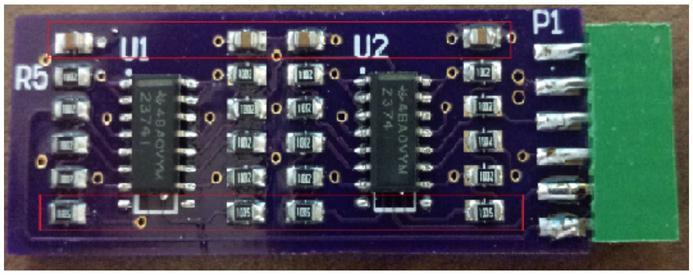


Figure 4.5: Transimpedance board

4.4 Testbench PCB

The classifier chip sits on a PCB which sets up power supply and other necessary signals ready for delivery to the chip. The PCB is powered by +8.5V and -3.0V power supply. They are used to make chip main power supply of 1V and other necessary supply voltages.

4.5 LabView Setup

The LabView setup consists of applying appropriate input signals to the chip with proper timing and reading back the output signals from the chip and log them in a file. The capability of LabView to send and read signals reduces the need for many power supply, signal generators and oscilloscopes. Also LabView is a good choice of tool for measuring chip performance because it allows control of timing of the signal delivery, signal acquisition and log data. Several digital signals are delivered to the chip using an infinite for loop as shown in Fig.4.6. The controls at *data*2 in the front panel can be used to supply digital signals to the chip using mouse or program. Table4.1 below gives a description of each of the digital signals. Similarly the analog signals are applied using an infinite loop as shown in Fig.4.7. This code uses the control box *Current* in the front panel to supply analog voltage to the transconductance board which supplies input

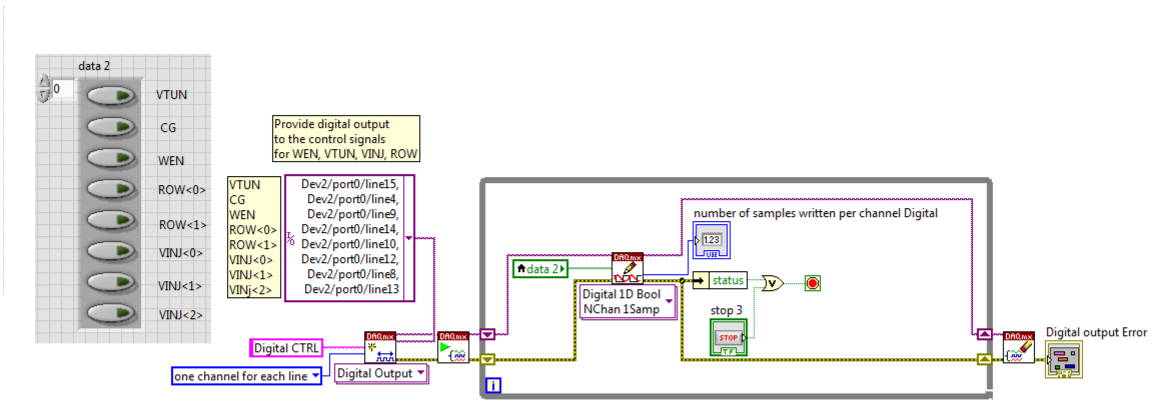

data 2
VTUN
CG
WEN
ROW<0>
ROW<1>
VINJ<0>
VINJ<1>
VINJ<2>
Provide digital output to the control signals for WEN, VTUN, VINJ, ROW
VTUN CG WEN ROW<0> ROW<1> VINJ<0> VINJ<1> VINj<2>
Dev2/port0/line15, Dev2/port0/line4, Dev2/port0/line9, Dev2/port0/line14, Dev2/port0/line10, Dev2/port0/line12, Dev2/port0/line8, Dev2/port0/line13
Digital CTRL
one channel for each line
Digital Output
data 2
Digital 1D Bool NChan 1Samp
number of samples written per channel Digital
status
stop 3
Digital output Error

Figure 4.6: LabView code for digital signal delivery

Table 4.1: Description of digital output signals from LabView

signal	Description
VTUN	Applies 7V at the memory transistor for tunneling
CG	Control gate signal that controls injection
WEN	Enable signal for the injection transistor
ROW⟨0⟩	row select bit for decoder logic
ROW⟨1⟩	row select bit for decoder logic
VINJ⟨0⟩	injection select bit for 1st column select
VINJ⟨1⟩	injection select bit for 2nd column select
VINJ⟨2⟩	injection select bit for 3rd column select

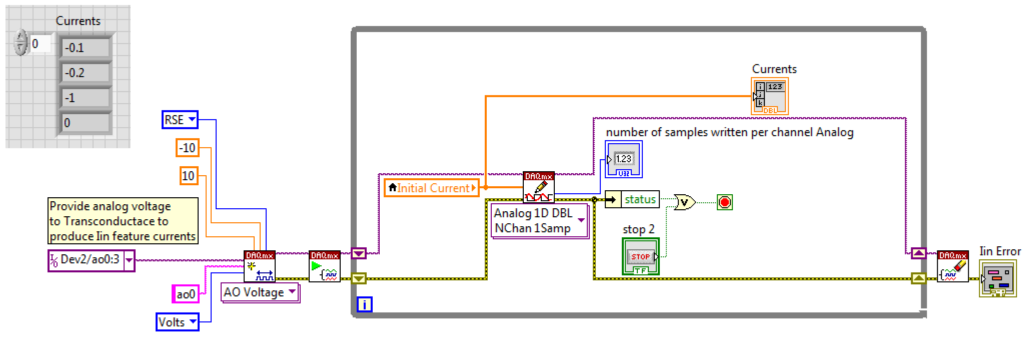

Currents
0
-0.1
-0.2
-1
0
RSE
-10
10
Provide analog voltage to Transconductace to produce Iin feature currents
Dev2/ao0:3
ao0
Volts
AO Voltage
DAQmx
Initial Current
Currents
number of samples written per channel Analog
Analog 1D DBL NChan 1Samp
status
stop 2
STOP
Iin Error
i

Figure 4.7: LabView code for analog signal delivery

Table 4.2: Description of analog data read out

Signal	Description
I_{op1}	Output voltage of TIA for the 1st column of the classifier
I_{op2}	Output voltage of TIA for the 2nd column of the classifier
I_{op3}	Output voltage of TIA for the 3rd column of the classifier
WTA_1	1st column output from winner take all
WTA_2	2nd column output from winner take all
WTA_3	3rd column output from winner take all

currents to the classifier chip. By varying the values in control box *Current* in the front panel the input currents can be varied.

There are several analog signals to be measured. They are sampled at a rate of 1KHz and displayed to the oscilloscope in the front panel continuously 100 samples at a time. The LabView code is shown in Fig.4.8. This code also includes additional codes for averaging out the value of 100 samples of data that is being displayed on the oscilloscope. This helps see the value of the signals when the data has become steady and it is easy to read out this value rather than looking at the oscilloscope. The table4.2 below describes the type of data being read out.

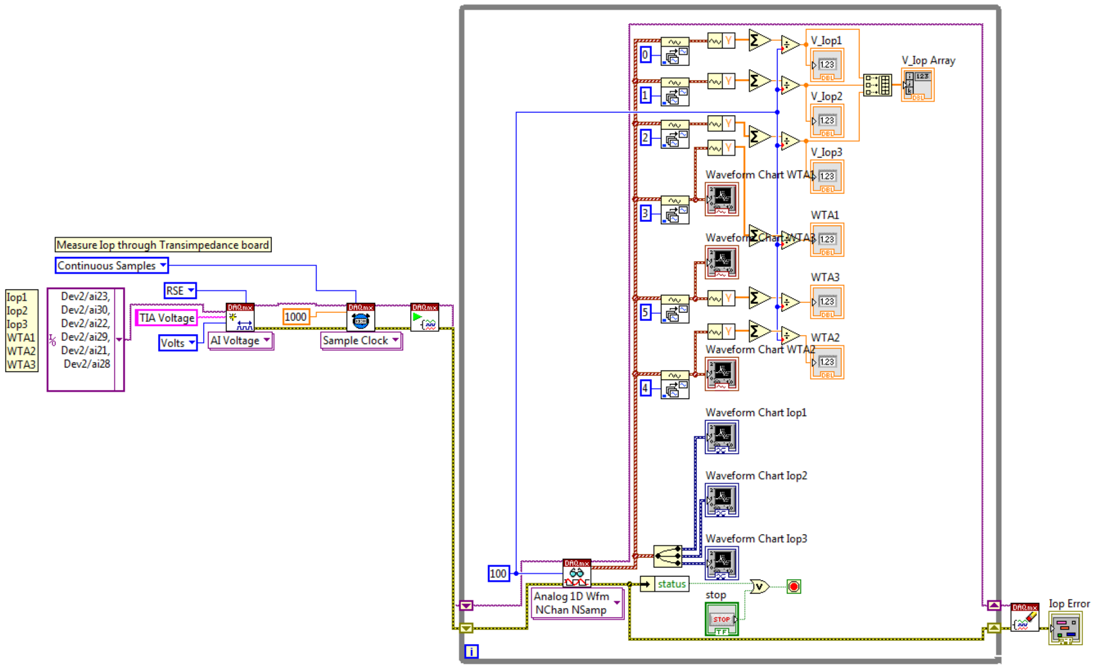


Figure 4.8: LabView code for analog signal measurement

4.6 Floating gate weight estimation

When the chip is fabricated there are some static charges trapped on the gates because of the fabrication process. Hence there are some weight already present in the floating gates even before the process of setting the weights begins. The weights in the floating gates can be estimated using the multiplier output current. Fig.4.9 shows the multiplier used in the chip. The input current is sourced from node VB. ENz acts as enabler for the circuit. $VREF$ is used to set multiplication operating point. The floating gate weight is applied to node VIN. The multiplication of the input current and the weight is proportional to the current difference of the two legs. If $VREF$ becomes equal to VIN then equal current flows on both legs and difference current becomes zero. Hence by varying $VREF$ and checking the multiplier output current value of VIN or the weight

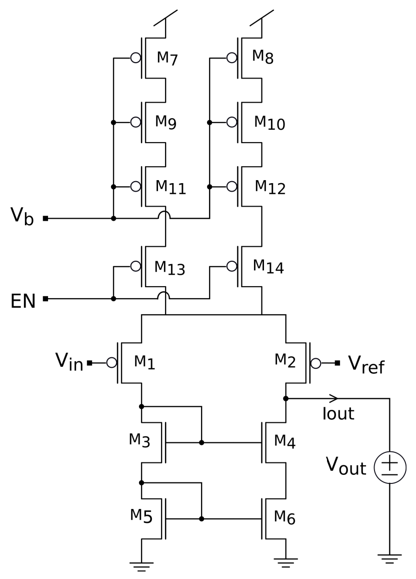


Figure 4.9: Analog multiplier circuit

can be estimated. The output multiplier current is measured using the TIA voltage. From Eq.4.5 if current is zero then output voltage is V_{ref} which is 1V. Fig.4.10 shows the output voltage of TIA for three columns of the classifier as $VREF$ is varied. Input current is supplied to one row and other input current is kept zero. From the figure it can be estimated that on that row, column 1 has weight of 0.8 and the other two has weight of 0.9.

4.7 Multiplier current variation with input current

Since this is a linear classifier the output current from a multiplier for a class should vary linearly with input current. Hence if we increase the input current from zero the output current should also increase or decrease. To measure this the first two input currents are kept variable and the other two are kept constant at zero. Here the value of the weights stored in the memory transistors are not

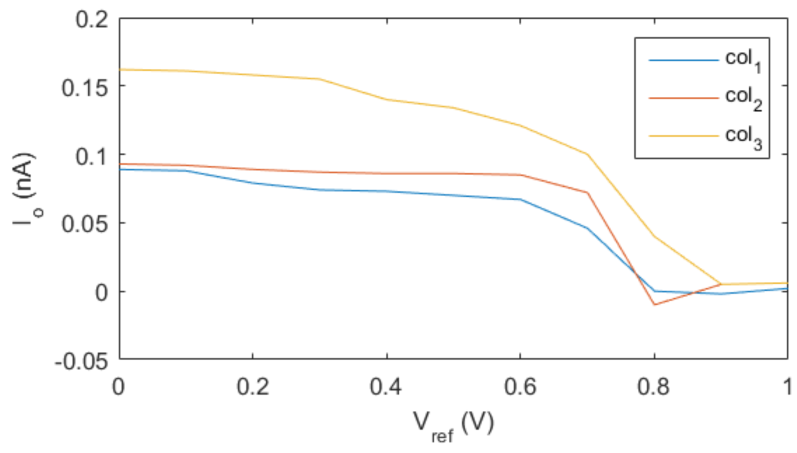


Figure 4.10: Floating gate weight estimation

important. As long as the weights are not zero any value will do. The currents in the first two row the classifier is varied in a for loop and the output voltage is logged in a .csv file. Then the .csv file is imported in MATLAB and the output current is plotted. The code for current variation and data logging is shown in Fig.4.11. The output current is measured in units of voltage from TIA. In the MATLAB script the actual current is calculated using back calculation. The result is shown in Fig.4.12. It can be seen that I_{op1} and I_{op3} decreased from a value and I_{op2} increased steadily as the input currents were changed.

4.8 Class separability

With the same code used in the previous section the output from the winner take all can also be logged. By varying the input currents the winner take all data is logged and a MATLAB script is used to plot the results. The output class is identified by the WTA value having the highest value. The result is shown in Fig4.13. It can be seen that the classes are separated by lines which indicates that the classifier is able to differentiate the input space.

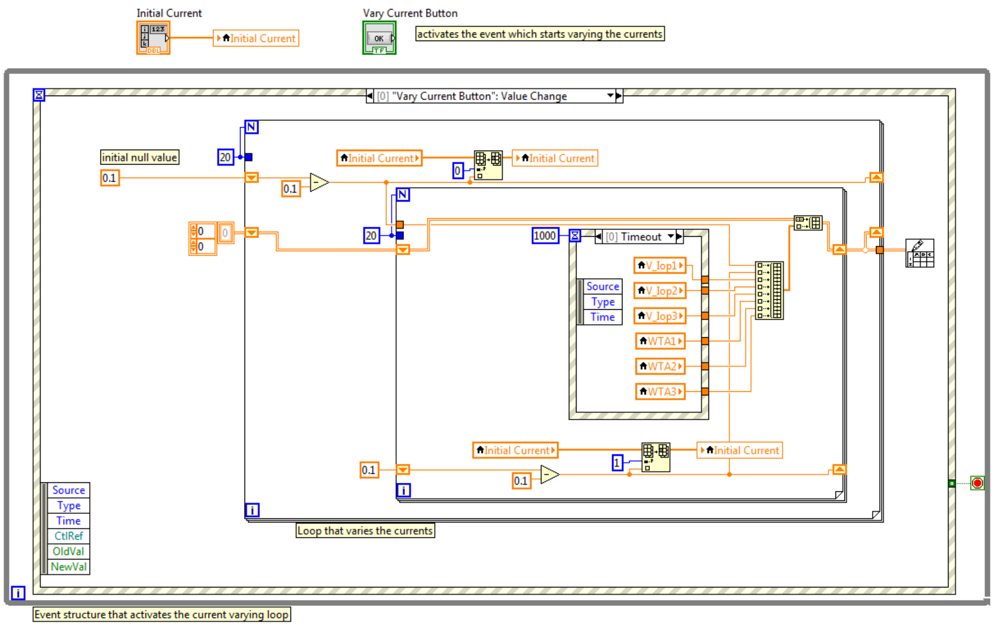


Figure 4.11: LabView code for input current sweep

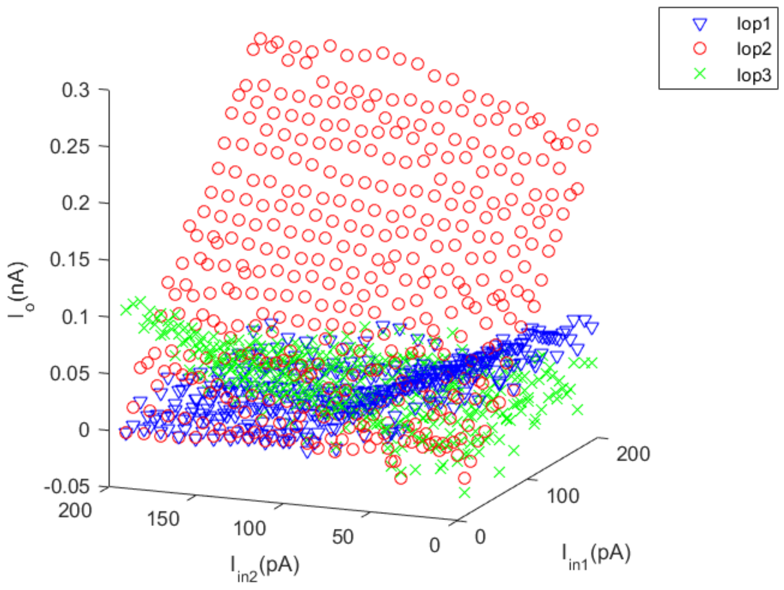


Figure 4.12: Multiplier output current variation with input current

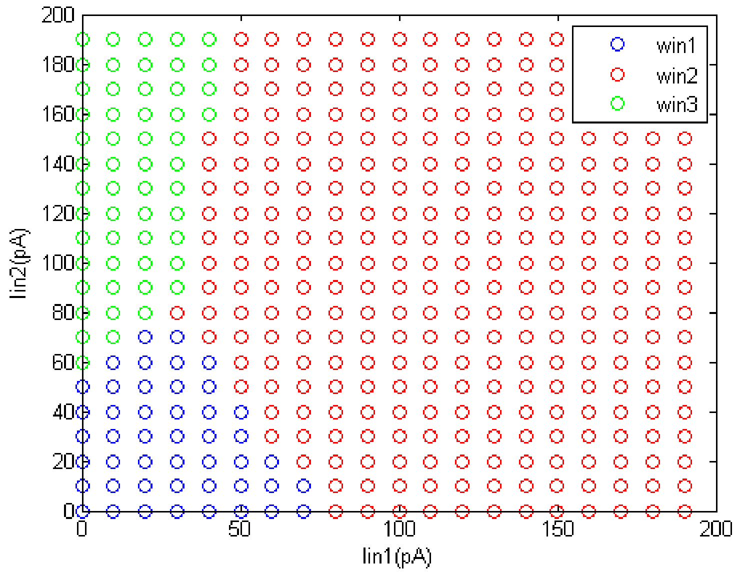


Figure 4.13: Separation of classes

4.9 Speed of classification

The speed of classification is defined as how fast the winner take all signal changes its value. The idea is that the classifier will run on a clocked system. A pattern is presented to the input and the result is read out in each clock cycle. Hence if it takes 10us to change the decision from the input to the output path, then it is possible to classify 100k vectors/second. To measure the time the winner take all signal changes its decision, the change of the WTA signal is logged in .csv. The result is shown in Fig.4.14. From the figure the rise time is estimated to be less than 10ms.

4.10 Input referred noise

When the classifier works near the decision boundary it will produce some wrong results. These wrong result are because of the noise present in the system

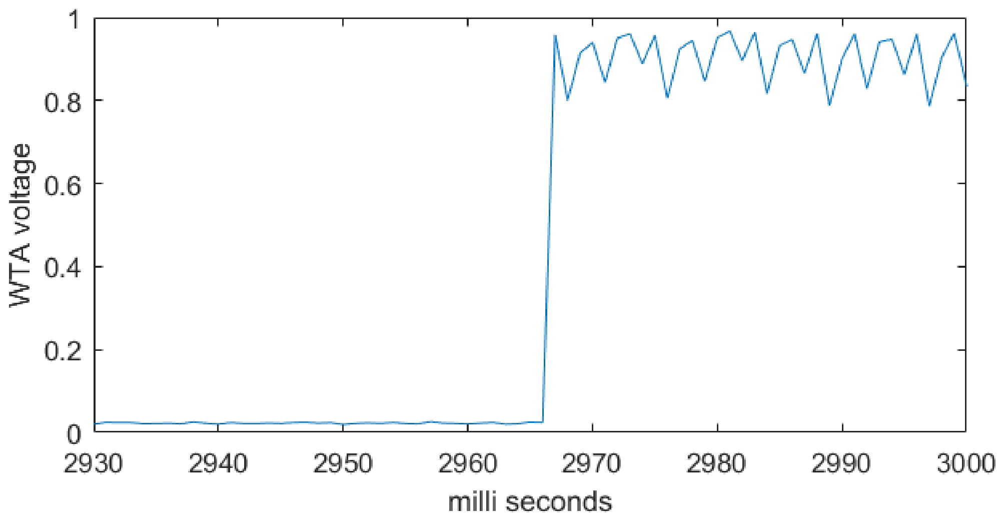


Figure 4.14: Rise time of a WTA signal

which is assumed to be gaussian. This noise will cause uncertainty in the outcome the classification result. To measure the noise a statistical approach is used. If the classification result is sampled near the decision boundary many times, probability of the class being classified correctly is found. This represents the cumulative probability density function (CDF). Differentiating this CDF, a probability density function (PDF) is found. LabView code is used to sweep one input variable near the decision boundary 100 times and the other variable is kept constant a value. The winner take all data is logged. The probability is calculated as 1 if the classification result is correct and 0 if the result is wrong. Fig.4.15 shows the winner take all decisions near the decision boundary. It can be seen that from steady decisions the winner take all is affected by noise at and near the decision boundary and then gets steady again. From this data the CDF and PDF is calculated as shown in Fig.4.16. Curve fitting is used to fit to the experimental CDF. Then CDF is differentiated with respect to input current to get the PDF. From the figure the standard deviation σ_N is estimated to be

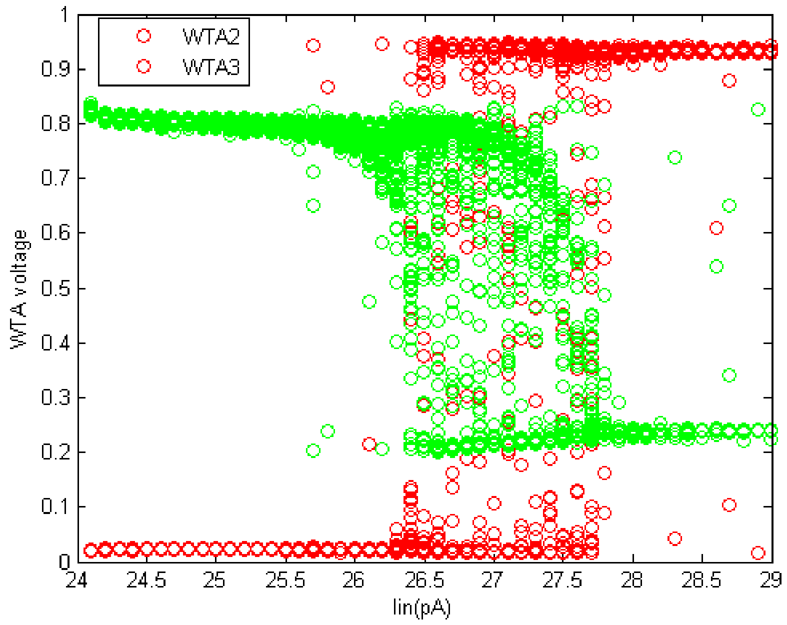


Figure 4.15: Wrong decisions near decision boundary

around 0.3pA. This can be interpreted as the input-referred noise of the classifier.

4.11 Input current to the classifier

A Source Meter Unit (SMU) is used to measure the input current into the chip. The SMU is set up to measure current at a voltage of 1V. The SMU is connected to the main VDD rail of the chip and the chip is turned on. A current of around 4nA is found to be sourcing from the SMU. A summary of the input current as the supply rail voltage to the chip is varied, is given in table4.3.

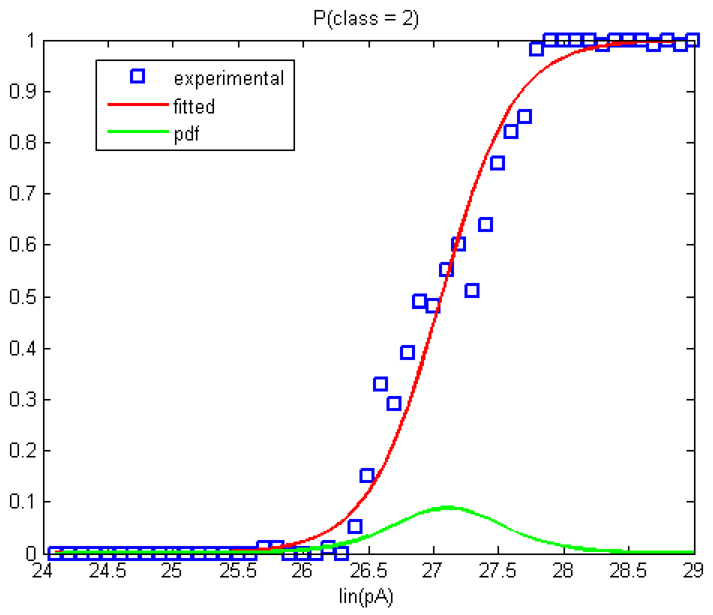


Figure 4.16: CDF and PDF calculation

Table 4.3: Input current to chip and stability as VDD is varied

VDD	I_{supply}	WTA stability
1.0V	4nA	stable
0.9V	4nA	stable
0.8V	4nA	stable
0.7V	3-5nA	unstable
0.6V	3-5nA	stops working

4.12 Setting specific decision boundary

The classifier needs to have specific decision boundary to classify real input patterns. One way to set specific decision boundary is to set the weights in the memory transistors the same as the weights of the decision boundary. But this has a problem. The transistors fabricated in the chip will have mismatch. Hence even if the weights are accurately set up the same for every single memory, the currents output from them will be different. This is why instead of setting up the weights that way, an iterative approach is adopted. The tunneling is global for every memory transistor. So when tunneling happens every transistor's weights go to high value. The injection process brings the value of the weights down. One by one for every floating gate the injection process continues until the weight is increased a little. The output classification pattern is examined. This process continues until the output classification pattern is obtained as required. This way the weights will be set up properly the way it needs to be to produce the required output classification. In Eq.4.2 there is a constant term on the right hand side which is the bias term. This is set in the chip by setting a constant current in the third row. This current multiplied by the weight in the floating gate memory produces a constant bias term. The weight of the floating gate memory for the bias term is also subject to iterative weight setting process.

Fig.4.17 shows the circuit diagram of the floating gate (FG) memory transistor. Tunneling happens when CG node is pulled down to 0V and $VTUN$ node is pulled up to 7V. For injection to happen transistor $T2$ needs to turn on. This is done by sending digital high to W_EN and SEL which pulls vdd_int node up to 3V. $VINJ$ is pulled down. Using CG a voltage of around 3V is coupled to node FG. This turns on $T2$. When current flows in $T2$ some of the electrons are injected into the FG node. The table4.4 summarizes the process. A pulse

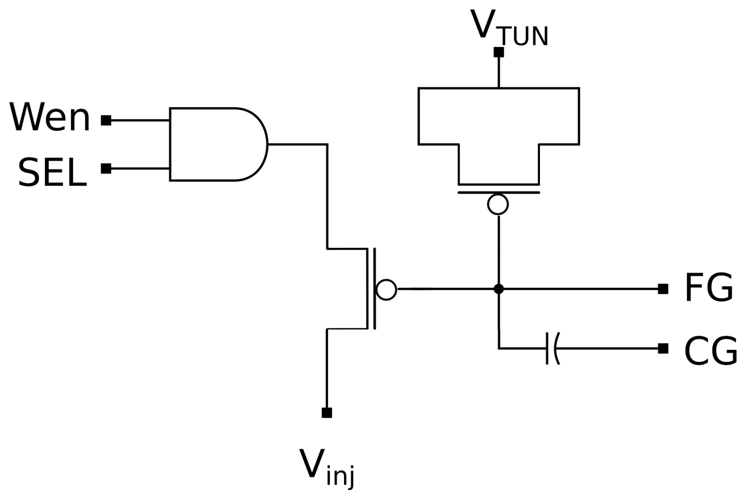


Figure 4.17: Floating gate memory circuit

Table 4.4: Injection and Tunneling signal

process	VTUN	CG	WEN	SEL	VINJ
Tunneling	7V	0V	0	0	1.5V
Injection	0	3V	3V	3V	0

train of CG with period 100ms is applied for 50 cycles before the output currents are checked. A LabView code automates the process of injection in an iterative manner until the desired output classification pattern is obtained. An attempt to obtain an output pattern like in Fig.4.2 provides the result shown in Fig.4.18 after many iterations. It is not exactly the same as Fig.4.2 but is on its way towards that direction.

It was very difficult to implement specific decision boundary in the chip because every time one floating gate is adjusted other set values on other floating gates changes from the previous value a little. A look at the circuit diagram reveals that there are some injection going on in $T2$ transistor when it is not supposed to happen. When injection is not happening vdd_int and $VINJ$ is 0V. So there should not be any current in $T2$. But the $VINJ$ node does not go all

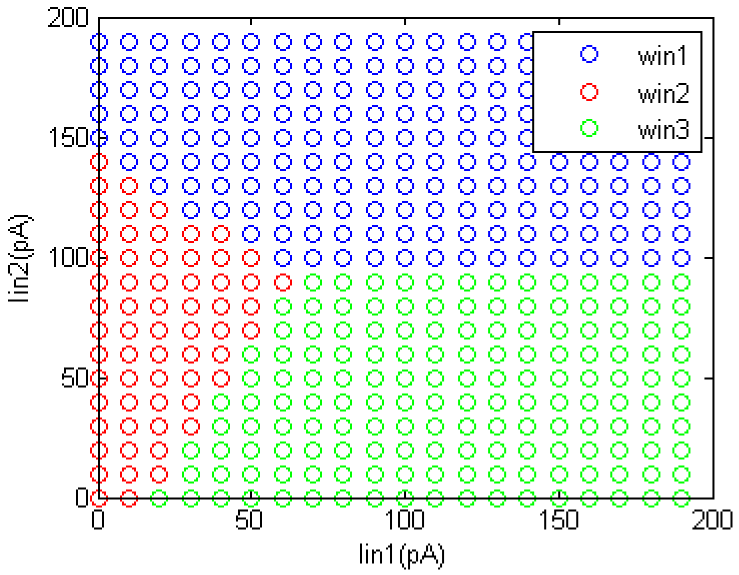


Figure 4.18: Specific decision boundary set in classifier chip

the way to 0V for the PCB circuit that supplies the $VINJ$ to the chip. There is a small voltage around 0.3V on that node when it should be 0V. Since CG is global which supplies 3V on the floating that needs injection, it also connects to other floating gates. Hence there is a current in $T2$ from $VINJ$ to vdd_int. This injection causes the floating gates to change its value when there should not be any injection. Although this current should not have enough energy to cross over the oxide barrier but it is a possible explanation. Further on the programing routine of the floating gate is given in the appendix.

Chapter 5

Conclusion

This thesis explored the applicability linear discriminant classification algorithm in solving the problem at hand. Linear discriminant analysis uses multiplication and addition to make decision. These two operations have been implemented by using analog multiplier and summing the output current together. The analog nature of the computation helped it use only few transistors to implement multiplication, far less than its digital counterpart. The physics of Kirchhoffs law let us directly compute the sum by simply connecting the output currents from the multiplier together.

It is shown that by using proper frequencies the classifier can achieve good accuracy. Genetic Algorithm has been used to find the optimum set of frequency combinations. However, any numerical method could be used to do the same job. A mix of acoustic and vibration frequencies show better performance in classification.

The thesis also explored the impact of noise in the decision making process. Since the classifier operates on very low current level, the output decisions are affected by the noise. While noise limits the ability of the circuits to operat at arbitrarily low power values and some wrong decisions are unavoidable at decision boundary, reasonable SNR can still provide reliable classification results. The noise of multiplier and the classifier showed there is a minimum level of energy that needs to be spent to achieve a given level of accuracy. This fundamental limit is useful in designing the classifier in a larger scale because it predicts how much accurate the classifier will be for a given power supply.

Experimental results show operation of the classifier in a commercially available 130nm silicon process. The results clearly show that the analog implementation of the linear discriminant classifier reliably identifies three different classes while the input current is varied. Although the analog devices suffer from offset, proper biasing techniques corrects for that error. For example, the offsets associated with floating gate outputs are generally not problematic since the floating gate values are set just enough such that the circuit produces a desired output current, irrespective of the actual floating gate value.

In conclusion, the analog design of machine learning shows itself as a powerful and resource saving alternative to digital computation techniques. There is a growing interest in the subject of approximate computation because not all computation techniques need 32-bit/64-bit accuracy. So any computation techniques like analog computation that is resource saving, can take us closer to achieving low cost, mobile artificial intelligence.

Bibliography

[1] C. A. Mead. *Analog VLSI and Neural Systems.* Reading, MA: Addison Wesley, 1989. 1

[2] Lyon R. F. Sarpeshkar. R and Mead C. A. A low-power wide-dynamic range analog vlsi cochlea. *Analog Integrated Circuits and Signal Processing*, 1998. 2

[3] AG Andreou KA Boahen. A contrast sensitive silicon retina with reciprocal synapses. *Advances in Neural Information Processing Systems*, 1991. 2

[4] G. Indiveri S. Liu, J. Kramer and T. Delbruck. *Analog VLSI: Circuits and principles.* The MIT Press, Massachusetts, 2002. 21, 24

[5] R. Sarpeshkar. Analog versus digital: Extrapolating from electronics to neurobiology. *Neural Comput.*, 10(7):1601–1638, 1998. 23

[6] J. Holleman S. Young, J. Lu and I. Arel. On the impact of approximate computation in an analog destin architecture. *IEEE Trans. On Neural Networks and Learning Systems*, 25(5), 2014. 25

[7] C.E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27, 1948. 40

[8] j Rabeay. *Digital integrated circuits.* Englewood Cliffs, N.J, Prentice Hall, 1996. 40

[9] B. J Hosticka. Performace comparison of analog and digital circuits. *Proceedings of the IEEE*, 73:25–29. 40

Appendix

A Floating Gate Programing

Floating gate used in the classifier store a voltage by storing electrons on the gate of a pMOS transistor. The transistor size is $360nm \times 360nm$. The source and drain is tied together so that the transistor acts like a capacitor. The gate acts as the floating gate FG. The voltages at the floating gate is changed by tunneling and hot electron injection.

A.1 Tunneling

When $VTUN$ is pulled up to 7V and CG is pulled down to 0V, electron from the FG node tunnels into the substrate through the gate oxide. Lack of electron at FG causes the voltage at FG to increase. This increase in voltage would cause the multiplier output current to increase. Hence by watching the multiplier output current we would know that the voltage at FG has increased. We would induce tunneling for one second by pulling $VTUN$ to 7V and CG to 0V. Then pull $VTUN$ to 0V and wait about a second to let the system settle down. Then measure the multiplier output current. If we need to increase the current even more the procedure is repeated. The following shows the algorithm.

Algorithm 1 Tunnleing Routine

```
$I_o \leftarrow$ desired multiplier output current
$i_o \leftarrow$ measure multiplier output current
$CG \leftarrow 0V$
while $i_o < I_o$ do
    $VTUN \leftarrow 7V$
    wait 1000ms
    $VTUN \leftarrow 0V$
    wait 1000ms
    $i_o \leftarrow$ measure multiplier output current
end while
$VTUN \leftarrow 0V$
```

A.2 Injection

The injection current is applied by pulling the output of the AND gate to high 3V and pulling V_{inj} to 0V. Then a pulse of 3V is applied the gate of injection transistor using CG node. The current flow through the injection transistor creates some electron hole pair. Some of the electrons will have enough energy to cross over through the oxide into the FG node. Accumulation of electrons decreases the voltage at FG. This is repeated until we get the desired output multiplier current. The following shows the algorithm.

Algorithm 2 Injection Routine

```
$I_o \leftarrow$ desired multiplier output current
$i_o \leftarrow$ measure multiplier output current
$WEN \leftarrow 3V$
$SEL \leftarrow 3V$
$V_{inj} \leftarrow 0V$
while $i_o > I_o$ do
    100 pulse train of CG=3V with 100ms period
    wait 1000ms
    $i_o \leftarrow$ measure multiplier output current
end while
$WEN \leftarrow 0V$
$SEL \leftarrow 0V$
```

Vita

Md Munir Hasan was born in Bangladesh on 16 November 1990. The ”Md” is an abbreviation of the name ”Muhammad” and this how it is usually used in Bangladesh. He received his B.S. degree from Bangladesh University of Engineering and Technology in 2013. He worked as a Software Engineer in Samsung R&D institute Bangladesh from 2013 to 2015. There he worked in Windows Phone project. He started graduate school in electrical engineering at University of Tennessee Knoxville in Fall 2015. His has research interests at the boundary of machine learning and analog computation.