STATISTICAL MODELING OF AN
ELECTRONIC OLFACTORY

By

MEYYAPPAN RAMANATHAN

Bachelor of Science

Bharathiar University

Coimbatore, India

1989

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 1995

STATISTICAL MODELLING OF AN

ELECTRONIC OLFACTORY

Thesis Approved:

Chris Hutchens

Thesis Adviser

George Scheets

R. Ramakumar.

Thomas C. Collins

Dean of the Graduate College

PREFACE

This thesis attempts to provide an understanding of the natural design principles that underlie the observed learning/clustering performance of the modified GLA olfactory neural network which retains the essential clustering properties of the olfactory bulb and paleocortex in pattern recognition. A statistical model is developed to model the proposed hardware implementation of the modified GLA model. This statistical modelling of the modified GLA model will assist in the understanding and optimizing the design and architecture dimensionality and also in intepreting the test results.

I wish to express my gratitude to my advisor, Dr. Chriswell Hutchens, for his incomparable guidance, inspiration, moral and financial support. I am also thankful to Dr. George Scheets and Dr. Ramachandra Ramakumar for serving on my committee. A special thanks is due to Dr. Patrick Shoemaker for his guidance throughout this work. Also, I wish to thank the office of the Naval Ocean Systems Center, San Diego, for the computing resources and the valuable experience I gained throughout this project.

I extend my thanks to my friends in Stillwater and outside for their everlasting support and encouragement.

Finally, my deepest appreciation is extended to my parents, brother and sister's family for their love, support, moral encouragement and understanding.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

NOMENCLATURE

Δ_w	Synaptic increment of w_{ijkl} per training episode
Θ_{ff}	Piriform refractory frequency facilitation threshold
Θ_I	Threshold to periglomerular to eliminate inhibition noise floor
ΘM_j	Threshold of the jth mitral cell
Θ_P	Piriform cell threshold
PHIFF	System clock active in Forward phase
PHIFB	System clock active in Backward phase
PHILRN	System clock active in Learn phase
D_x	Subscripted diode
g	Number of glomeruli in bulb patch
G'_i	Automatic Gain Controlled signal
G'_{imax}	Maximum value of element in G'_i
G^*_i	Glomerulus input (un-normalized)
G^*_{imax}	Maximum value of element in G^*_i
G_i	Normalized glomerulus output; mitral patch input
g_{mx}	Small signal channel transconductance of subscripted MOSFET
$g_s(x)$	Nonlinear mapping function that maps glomeruli activity
h	Number of piriform cells per piriform patch
I^*_i	Aggregate un-thresholded inhibition to glomerulus i

i	Counting index for g
I^*_{ij}	Weighted inhibition on LOT line ij in backward direction
j	Counting index for m
k	Counting index for p
K_{1-4}	Constants of non-linear function
K_G	Constant to set percentage activation of the glomerulus
l	Counting index for h
m	Number of mitral cells per glomerulus
M_{ij}	LOT from jth mitral cell of the ith glomerulus
M_x	Subscripted MOSFET
O_i	Olfactory sensor output; olfactory system input
p	Number of piriform patches in cortex patch
P^*_{kl}	Output of the weight matrix, cortex patch input to lth piriform cell of kth piriform patch
P_{kl}	Output of cortex patch from lth piriform cell in kth piriform patch
PW_{kl}	Winning cortex output; olfactory output
d	Address for selection of output
$V\in$	Error voltage
V_K	Normalization scaling constant
VPGM	Programming voltage
W	Weight matrix
w_{ijkl}	Synaptic weight from LOT M_{ij} to the piriform cell P_{kl}
w_{max}	Maximum value of the synaptic weight w_{ijkl}
W^T	Transpose of the weight matrix

w_n	Minimum value of the synaptic (Naive) weight w_{ijkl}
y	Number of Clustering cycles
V_{osm}	Offset voltage due to mitral cell
V_{osp}	Offset Voltage due to the piriform cell
$\Delta w_{(ij)(kl)}$	Weight Matrix device mismatch error
ΔI_x	Current Transfer Error due to X current mirror
μ_w	Mean of the number of synapses in the winning piriform
σ_w	Standard Deviation of the number of synapses in the winning piriform
a_w	Number of active synapses in a piriform neuron
Σ_{SNR}	Signal to Noise Ratio

CHAPTER I

OLFACTION AND NEURAL NETWORKS

Introduction

Over the past decade, neural networks have drawn constantly increasing research attention and undergone significant developments in essentially three categories. The first category is that of mathematical description and analysis of the learning properties of neural networks, often working from biological and physiological exemplars [1,2]. The second and perhaps the largest, uses computer simulations to verify the validity of the neural network models in addition to demonstrating their applications [3,4]. The third group of research topics is the prospect of compact and dense hardware implementation of neural networks in analog integrated circuit form [5,6,7,8]. This thesis falls into the latter two categories.

The essence of a neural network lies in its distributed memory or knowledge processing, using massive interconnections and interactions, and in learning and self organization. The human brain is a signal storage and processing device. Neural networks provide a general framework for signal storage and processing and offer an exciting new approach to simulate human intelligence [9].

Neurobiologists evaluate the functioning of the brain by taking the bottoms up approach, studying the stimulus response characteristics of a single neuron and network of neurons. On the other hand, psychologists study brain function from the cognitive and behavioral level [9].

It is estimated that the human brain contains over 100 billion neurons and there are 10^{14} synapses in the human nervous system. Studies of brain neuroanatomy indicate often more than 1000 synapses are on the input and output of each neuron. Note that, although the neurons switching time (a few milliseconds) is about a millionfold times slower than current computer elements, they have a thousandfold greater connectivity than today's supercomputers [20].

Neurons and the interconnection synapses consistute the key elements for the neural information processing. Most neurons possess tree like structures called dendrites which receive incoming signals from the other neurons across junctions called synapses. There are three parts to a neuron: (1) a neuron body cell, (2) branching extensions called dendrites for receiving input, and (3) an axon that carries the neuron output to the dendrites of other neurons. The synapse represents the junction between an axon and a dendrite [15]. Nerve signal transmission in the brain is of two types: chemical signals across the synapses and electrical signals within the neuron. A neuron collects signals at its synapses by summing all the excitatory and inhibitory influences acting upon it. If the excitatory influences are dominant, then the neuron fires and sends this message to other neurons via the outgoing synapses. In this sense, the neuron function can be modelled as a simple threshold function $f(\bullet)$.

The modulation of synaptic junctions has long been regarded as the likely mechanism for learning and memory [13]. The long term potentiation (LTP) that is observed in the hippocampus, limbic system, and in some cortical structures of the brain, is believed to be similar to the mechanism used for learning[14]. The changes in the synaptic strength due to LTP are rather coarse when compared to the precise and graded weight changes that are offered by artificial neural networks. How a nervous system might respond to the computationally limited neural learning and neural processing that is used by artificial neural networks due to two dimensional connectivity [15] is a topic for much additional research. Extensive research is being carried out using computer simulations on such abstract neural network models to understand the effects of incorporated artificiality and also in an attempt to elucidate the organizational principle at the system level [1,2].

Artificial Neural Networks

The term artificial neural networks means any computing architecture that consists of massively parallel interconnected simple "neural" processors. The current structures of artificial neural networks are often based on the past and present understanding of the biological nervous system. Artificial neural networks are composed of many nonlinear computational elements. These computational elements operate in parallel and are arranged in patterns reminiscent of biological neural networks. Elements are connected via densely connected weights. Weights are typically adapted during use (learning) [3]. The information is held in these weights. New information is captured by changing the strength of the connection, of a group of untrained or partially trained weights. Contrary to Von

Neumann's computer which processes instructions sequentially, neural network models explore many hypothesis simultaneously using their massive parallel structures.

In its simplest form, a neuron sums weighted inputs and passes the result through a non-linearity. The neuron is characterized by an internal threshold or offset and by the type of non-linearity. The various types of mathematical non-linearities applied are hard and soft limiters, sigmoidal logistic non-linearities and hyperbolic tangents [3]. The hyperbolic tangent is similar in shape to the logistic function. It is most often used by the biologists as a mathematical model of nerve-cell activation. The most commonly used mathematical model non-linearity is the sigmoid logistic non-linearity.

One of the principal reasons for the interest in neural network models is the fact that many perform associative functions as a direct consequence of their architecture (and are therefore sometimes termed ' associative memory ' models). These associative functions include the ability to reconstruct original learned patterns from inputs that are fragmented or distorted versions of the patterns, the related ability for novel input patterns to elicit outputs of related patterns that were previously stored in memory, and the ability to link two or more unrelated patterns, especially when they occur at the same time, so that a subsequent input of one elicits the others from memory [11].

Neural network models offer their greatest potential in areas such as speech processing, image recognition, and pattern classification. In such applications, many hypothesis are pursued in parallel, high fault tolerant computation rates are required, and the existing computer systems are far from equalling human performance. When compared to traditional computing methods, the benefits of neural networks extend beyond the high computation rates provided by massive parallelism. The degree of robustness or fault tolerance provided by neural networks is greater than the fault tolerance provided by sequential digital computers. Because of the many processing elements and the robust interconnection, damage to a few neurons and synapses does not significantly impair overall performance. Like humans, true neural networks must recognize partial input information [10].

Olfactory Model

The modelling and fabrication of an olfactory is a difficult task since olfaction theories are still in the developmental stages. A functional model is required to allow purely functional designs to be pursued. On the one hand, a computer simulation of a too detailed anatomical olfactory model may result in a model which is beyond the feasibility of silicon implementation and may result in large volumes of difficult to analyze data, while on the other hand, too much abstraction and simplification of the anatomical olfactory results in the model loosing its relevance to biology altogether with the potential loss of computational power associated with the anatomical model. Thus, the efforts towards a moderate level of abstraction for the olfactory model is necessary. The correct choice of a model helps to understand the model as well as preserve the essential features of the model. A moderate level of abstraction for the GLA olfactory model [17] has been proposed by Granger, Lynch, and Ambros-Ingerson [10]. The interested reader is referred to the work of Granger et.al for details [16, 17, 18].

Richard Granger, Gary Lynch, and Amberose-Ingerson have reported a potentially useful model for the investigation of the aggregate network learning and memory properties of olfaction in behaving animals. This model referred to as the GLA model henceforth, deals with the interacting structures of the olfactory bulb and piriform cortex that have been observed in rats [16, 17, 18]. Computer simulations of this model have attractive computational properties, such as (a) the ability to identify clusters in the input cue environments at various levels of detail, hence achieving a form of hierarchial clustering, (b) the extensibility to unsupervised learning, (c) the ability to detect a weak odor obscured by a stronger one or identifying the significant component of a complex odor. A central feature of this model is the periodic sampling of input odors at the theta rhythm to which network response is locked. The theta rhythm matches the rhythm for both the hippocampal firing and the rate at which rats sample odor during learning. With successive sniffs of the input, hierarchical clustering and unmasking operations proceed sequentially.

Proposed Hardware Implementation of the Olfactory Model

We propose an direct implementation of the GLA model which retains the essential clustering properties of the olfactory bulb (OB) and paleocortex. Our proposed hardware model possesses several favorable features including: mixed mode, in lieu of a pure analog approach; current and voltage mode processing; discrete, coarse and unidirectional weight updates, leading to a simplified learning algorithm, and single quadrant multipliers.

The hierarchial clustering at the theta rhythm in the original network facilitates the use of a synchronous or clocked approach rather than full analog concurrent parallel processing. The input cues, analog current input vectors O_i , are assumed to be generated by sensors which are sampled periodically at an artificial theta rhythm (clock) θ_t. However, the generalized model input is not restricted to any frequency, spatial sampling or time series i.e speech or frequency spectrum spatial samples or image will suffice. For each cycle in this rhythm, there are two major nonoverlapping phases: activation of the OB and feedforward excitation of the piriform cortex (PC) indicated by PHIFF, followed by feedback inhibition of the OB by the PC indicated by PHIFB . The clocking sequence of the olfactory system is shown in Figure 1. Prior to the actual clustering, the network is trained over a set of the input cues by updating the forward (excitatory) nonvolatile weights in parallel according to the adult plasticity rule, utilizing hebbian learning coincident with the simultaneous activity at a winning piriform cell and an active mitral patch. Even though system control is derived through the clocks, the actual computation between the clocking is truly analog, concurrent, and carried out in parallel.

The essential blocks in our architecture consist of: the glomeruli normalizer within the OB, to normalize the glomerulus activity; a mitral patch within each glomeruli, to thermometer encode the networks's normalized inputs G_i^* ; the sparse weight matrix, to sparsely project lateral olfactory tract (LOT) activity onto the PC via the modifiable synapses winner take all (WTA) piriform patches within the PC, to exhibit the winner take all competition; tie resolver, to resolve potential ties that occur among two or more winning piriform cells within a piriform patch[17].

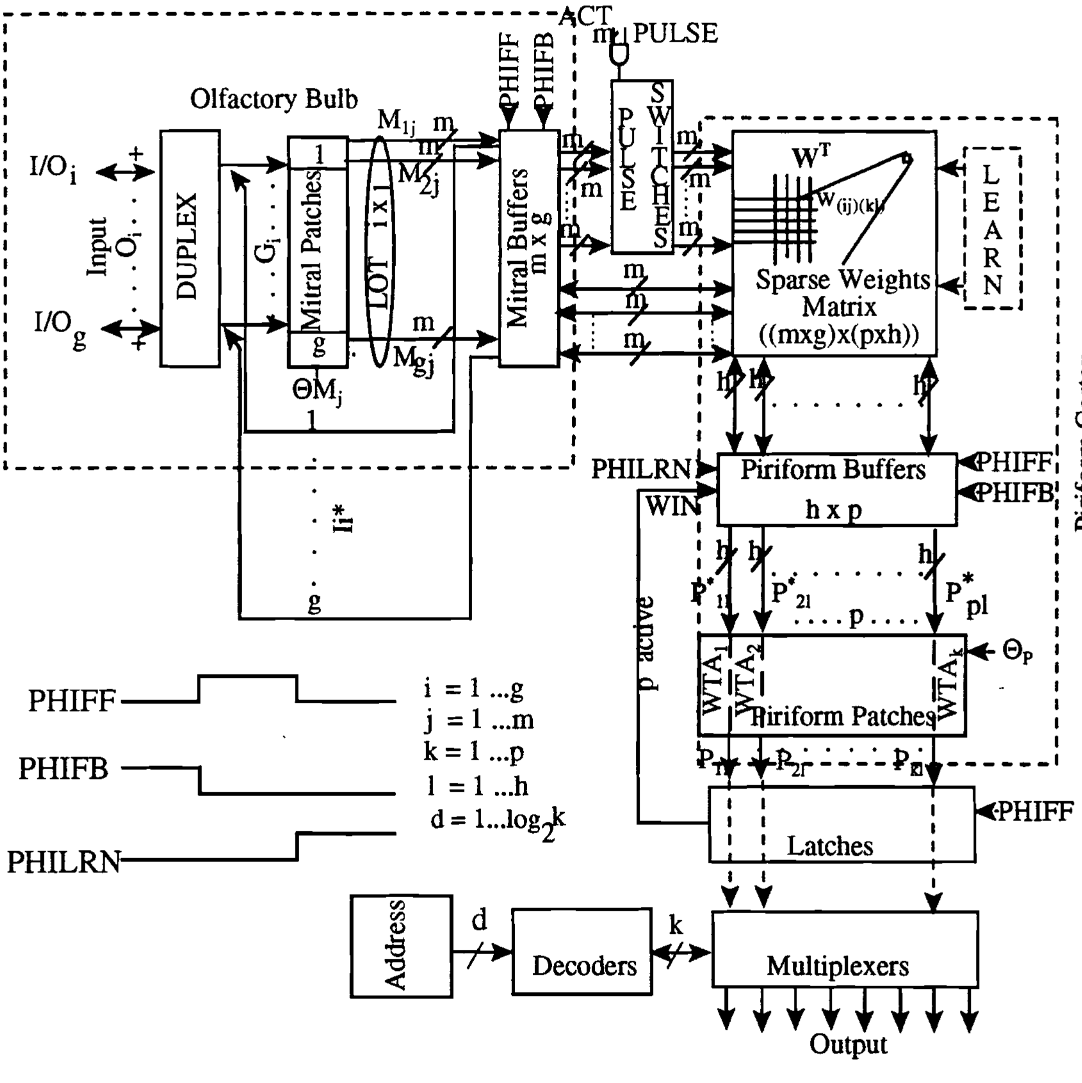


Figure 1. Block Diagram of Olfactory System

Olfactory Bulb

The olfactory receptor inputs contact mitral/tufted and periglomerular cells in bundles termed as glomeruli g. Each glomeruli is assumed to be associated with different types of receptor cells. Each glomerulus receives excitatory input from an ON collectively forming an vector O_i. It also receives an inhibitory feedback activation vector I_i from the PC through weights obtained after adaption. The excitatory inputs are combined with the inhibitory feedback signal and the resulting net inputs form the un-normalized activity G_i^* to the glomerulus.

Because of its dense inhibitory granule system, the bulb seems to be well suited to normalize inputs of different magnitudes from the receptor (i.e the greater the signal, the greater the inhibition it extracts). The resulting net inputs are then subjected to non-linear processing as well as global normalization mediated by the interaction between the excitatory and inhibitory cells of OB. Thus the bulb output is normalized such that the total number of mitral cells that are activated is reasonably constant across cues for different intensities and composition (the normalization process constrains the bulb so that only 20 % of the bulb cells are activated). The sum of the non-linearly mapped and scaled normalized activity remains nearly constant.

Each normalized glomerulus signal G_i is thermometer coded by the m mitral cells per mitral patch. Mitral cells have equidistant thresholds, $\theta_{Mj} < \theta_{M(j+1)}$ ($0 \leq j \leq m$), where θ_{Mj}, is the activation of the jth mitral cell in the glomerulus, globally generated by a resistor ladder. Electronically, this is equivalent to A/D conversion without encoding.

Sparse Weight Matrix

The binary voltage levels of the mitral cell M_{ij} in the OB are spatially projected on the hxp piriform cells in the piriform cortex via mxg LOT lines, forming the synapses between the OB and PC. The synaptic weights $w_{(ij)(kl)}$ are realized by floating gate, non volatile, analog programmable memory in conjunction with a MOS transistor, whose conductance is modulated by the charge on the floating gate. The weights are non-decremental, incremented in discrete steps (typically ~ 10% of their maximum weight), and saturated at a maximum value of $w_{\max}$ (~ two to three times their naive weights or greater is desired). The excitatory synapses $w_{(ij)(kl)}$ are sparse and they are randomly distributed within the PC with a sparseness on the order of 10%. The hardware sparse weight matrix, $W_{(mxg)(hxp)}$ consists of sparsely connected (2 of 20) synapses randomly arranged in the 4x5 submatrices. We believe that to restrict the PC random interconnection to a small local area is intuitively correct and biologically consistent, although unrealistically limited in area. However, the choice of a 4x5 area was selected for fabrication convenience and has no biological formulation. Each submatrix receives four consecutive LOT lines and five consecutive piriform input lines resulting in 20 cross junctions. The sparse (10%) pseudorandom connectivity within the submatrix is achieved by establishing two randomly chosen connections at these cross junctions via the placement of a weighing transistor. Within the submatrix any input LOT line may be interconnected with any piriform input line, with the exception that a double interconnection between a given pair of lines is excluded.

This architecture results in the uniform distribution of weights as opposed to the increasingly tapered distribution from caudal to rostral as reported for the anatomical model. Further, due to the restrictions imposed on the submatrix, there exists a zero probability of forming certain particular patterns of connectivity within a submatrix. The architecture does not appear limited by this effect. In networks which are sufficiently large, and with an increasing number of LOT lines the constrained distribution in the submatrix tends to be very similar to the unconstrained interconnection patterns of the anatomical model, with the exception of tapering.

Time multiplexing of the weight matrix W is used to compute, the weighted excitatory bulbar input currents to the PC in the forward phase, and the weighted inhibitory feedback currents from winning piriform cells to OB in the backward phase. Current Conveyor (CC) based Bidirectional Voltage/Current buffers (BiVI) permit such a bidirectional use of W[17].

Piriform Cortex

The currents produced by the innerproducts between LOT activity and sparse weights, are summed on the column of W. The total number of columns are organized into p patches with h neighboring columns per patch. The resulting innerproduct analog currents P_{kl}^* are amplified/scaled by the BiVI and fed into the PC. In the PC, the excitatory piriform cells P_{kl} are arranged into p disjoint winner-take-all piriform patches with h piriform cells/patch. The index k indicates the patch while l indicates the cell number within a piriform patch. Thus each column feeds only one corresponding piriform cell. The piriform patches exhibit a WTA competition within a patch which results in only a piriform cell or cells, associated with the highest current or greatest number of weight connections, to go high while rest of the cells remain low. The winning piriform cell is declared activated only if the input current to the corresponding piriform cell is equal to or greater than a piriform threshold θ_P. θ_P is used to support unassisted learning.

The WTA processed output P_{kl} ideally should have only *h* winners. But due to the finite resolution of the WTA circuit($\Delta I \sim 1uA$), it is not possible to avoid ties with the few near highest input currents . The tie resolver circuit has been added to the post WTA processing , thereby resolving ties digitally. Thus during the multisampling process, resolved WTA competition at each cycle results in a distinct output code used for clustering and forms the basis for feedback inhibition.

To implement feedback inhibition on the OB by the PC during the backward phase, binary outputs of the resolved winning piriform cell PW_{kl} are latched and reciprocally applied via the BiVI buffers to the multiplexed transpose (W^T) of the weight matrix, thus generating the inhibitory currents on the respective LOT lines configured for sinking the currents. The resulting inhibitory currents are amplified/scaled by the BiVI buffers. The inhibition on m consecutive LOT lines are summed by switching them together forming an aggregate unthreshold inhibition I_i^* associated with each glomeruli from which the respective forward LOT lines originated.

Multi-Sampling

The feedforward excitation and feedback inhibition cycles work in synchrony with a 4-7 Hz theta rhythm. Electronically this can be as high as a few Megahertz. After the first sniff a normalized OB output (with initial zero inhibitory feedback) will trigger the most active piriform cell in each patch on the basis of performance rules and random connectivity. The winning piriform cells in the PC are trained according to LTP which produces an inhibitory feedback. Thus the glomeruli with the most significant input components are more strongly inhibited and secondary components then elicit significant responses from their glomeruli in subsequent cycles.

In subsequent sniffs, the normalized activity of these secondary components has to increase in order to keep total normalized activity of glomerulus at a constant level. As a consequence, the spatial pattern of the mitral cell activity differs from the pattern generated in the first sniff. The mitral cells from glomeruli which are just inhibited do not fire whereas a larger number of mitral cells fire from the glomerulus whose normalized activity has just been increased. Hence a different activation pattern is generated which, in turn, produces a distinct bulbar-cortical output code. In short, secondary components are also inhibited and still weaker components are expressed in subsequent cycles and so on in a hierarchial fashion.

The process (bulb activation ⇒ normalization ⇒ cortical activation ⇒ inhibitory feedback) is repeated in each cycle until the bulb is sufficiently inhibited to be largely quiescent so that all the weaker stimuli are expressed. The process in which distinct bulbar-cortical responses are obtained by successively inhibiting components of the original stimuli is referred to as multi-sampling. During this multi-sampling process a hierarchial clustering takes place in which the initial output code indicates a main class or cluster membership, and subsequent codes indicate sub-clusters or subclass membership. Cluster and sub-cluster breadth in the input vector space appear to be dependent on weight increase, the ratio of saturated to naive weight values, and the data sample set on which the network learns.

The key to integrating the building blocks into a functional system is optimization of architecture dimensionality (g, m, p and h) and scaling of P_{kl}^{*} , I_i^{*} etc, which have a direct impact on transistor dimensions. This requires the development of a statistically-based system model to assist in the understanding and optimizing the design. Chapter II of this thesis presents a statistical software model of the modified GLA olfactory , Chapter III analyses the results of model simulations done using matlab and fortran to validate the models accuracy and Chapter IV deals with conclusion and future prospects of olfaction.

CHAPTER II

STATISTICAL MODELING OF AN ELECTRONIC OLFACTORY

Although modeling need not duplicate a biological system exactly, an accurate model is necessary to understand how a biological system functions in order to characterize its algorithm properly. In addition, an accurate model assists in the following:

(a) providing a better understanding of the paradigm,

(b) understand the process of learning,

(c) assists in the design or selection of the system architecture dimension (i.e the number of patches and number of cells per patch),

(d) assists in optimizing performance i.e. weight range and,

(e) provide a means to try new strategies.

The statistical modeling described in this chapter, in addition to the above mentioned general factors, assists the researcher in the determination of optimal network dimensionality, the feedback scaling of I_i^* and the determination of the distribution of the number of active synapses on a winning piriform neuron. This distribution, and its variance, contribute to exemplar subvector estimation error in I_i^* (the effective SNR). This knowledge is essential for simulation, analysis, transistor sizing and fabrication of the electronic olfactory system. It also provides a clearer picture of how clustering is achieved and the means by which I_i^* is estimated at each hierarchical level. An extensive analysis of how the device mismatch may affect the proposed hardware implementation is also completed.

Distribution of the winning piriform cells

The following discussion develops the distribution for the number of winning piriform cells which will be solicited during the feedback phase of each clustering cycle. Using ordered statistics [25], the distribution of the active synaptic connections (a_w) on a winning neuron in a winning neuron in a WTA piriform of patch size h can be calculated as:

$$g(a_{\max}) - hf(a_{\max}) \left[\int_{-\infty}^{a_{\max}} f(\,.\,)\, dp \right]^{(h-1)} \qquad (1)$$

for $-\infty < a_{\max} < \infty$

where f (.) is the synaptic distribution of the active synapses on a piriform neuron (weight matrix column)

The number of active synapses present on a piriform neuron can be estimated statistically by noting that a hypergeometric distribution [6] is formed by the interaction of A active out of N LOT lines, within the presence of n possible active synapses in N possible locations and it is written as follows:

$$f(a_w) = \frac{\binom{A}{a_w}\binom{N\text{-}A}{n\text{-}a_w}}{\binom{N}{n}} \qquad (2)$$

where

N is the number of LOT lines

A is the number of active LOT lines

n is the number of synapses on a piriform neuron

a_w is the number of active synapses on a neuron

and the mean and standard distribution are given by:

$$\mu = \frac{nA}{N} \qquad (3)$$

$$\sigma^2 = \frac{(nA/N)\,(1\text{-}A/N)\,(N\text{-}n)}{(N\text{-}1)} \qquad (4)$$

For a relatively large N and n A/ N > 4, a hypergeometric distribution can be accurately approximated by a normal distribution. Further considering the sparsity of the weights to be equal to 0.1, then n = 0.1 N synapses per column. This observation is biologically well founded and results from the probability of a LOT to piriform synapse occurrence which is in the range from 0.1 to 0.2 in the piriform cortex [26]. This results in the operational constraint that A be greater than 40. Further, given the biological observation that LOT activity (K_G) is limited to 10 to 20% then N must be greater than 200. This result also is very realistic biologically, since N is easily in excess of several thousands [26]. The distribution of the active synapses on a piriform neuron can now be written as follows:

$$f(a_w)=\frac{e^{-1/2(a_w-\mu/\sigma)^2}}{\sigma\pi\sqrt{(2)}} \tag{5}$$

where: μ and σ are from equation (3) & (4) and σ^2 now can be written for large N as:

$$\sigma^2=\frac{nA(N-n-A)}{N^2} \tag{6}$$

Substituting equations (5) into equation (1) and integrating, the distribution of the winning piriform neuron can be written as:

$$g(a_{\max})=\frac{-me^{F}}{\sqrt{\pi}\left|\sigma\right|\left[ERF\left(\left(\frac{\mu}{2\sigma}-\frac{a_{\max}}{2\sigma}\right)sign\sigma\right)-1\right]} \tag{7}$$

where

$$F=-\ln(2)+\ln\left(1+ERF\left(\frac{\sqrt{2}a_{\max}}{2\sigma}-\frac{\sqrt{2}\mu}{2\sigma}\right)\right)-\frac{\mu^2}{4\sigma^2}+\frac{\mu a_{\max}}{2\sigma^2}-\frac{a_{\max}{}^2}{4\sigma^2} \tag{8}$$

The mean and variance of the distribution in (7) are not readily determined in closed form. However, by plotting this distribution it is easily observed that the mode and mean μ_w increase as *h* increases, while the variance σ_w decreases with increasing *h*. This results in a better estimate of μ_w as *h* approaches infinity and corresponds well with intuition. Note, having won and with training the resultant winner will be shifted well out on to the tail of the distribution. This will be demonstrated in chapter III, figure 10.

Distribution of the Inhibition Response

Using the preceding nomenclature the expected value of the inhibition signal can be determined as follows. The winning synapses a_w are actively updated or trained during learning while the remaining, (n - a_w) remain unchanged. After training, a_w weights will have a value of w_{max} , while (n - a_w), have the initial or the naive value w_n . Each winning piriform patch (element of P) can be viewed as a binomial distribution trial when calculating the inhibition vector

$$I_i^{*}=W^{T}P \qquad (9)$$

Ideally during feedback in a trained network, where a pure or a noise free exemplar is applied to the input (template M_{ij} or G_i), the occurrence of an active trained synapse is mutually exclusive of a naive or a untrained synapse. This is also true when the P (and P_w after tie resolving) vector (which results as a direct application of M_{ij} or G_i) is applied to the W^T in equation (10).

In the calculation or estimation of I_i^* three potential sources of error exist: contributions due to naive weights, both the mean (μ_n) and standard deviation (σ_n); and standard deviation (σ_w) associated with the number of trained weights (w_{max}) on a winning piriform patch equation (7). The expected value of the estimated exemplar (I or I^*) is determined by μ_w. Therefore the expected values of inhibition for a fully trained W^T matrix at a currently excited, and unexcited mitral cell are:

$$\mu_{Ma} = \frac{\mu_w w_{\max}}{A} P \qquad (10)$$

$$\mu_{Mn} = \frac{w_n (n-\mu_w)}{N-A} P \qquad (11)$$

respectively, and their respective variances are

$$\sigma^2_{Mn} = \frac{W^2{}_n (n-\mu_w)}{N-A} \left(1 - \frac{n-\mu_w}{N-A} \right) P \qquad (13)$$

$$\sigma^2{}_{Ma} = \sigma^2{}_w \qquad (12)$$

The expected signal to noise ratio, " SNR " for the inhibition vector I_i^* or exemplar estimate can now be written as:

$$e_{SNR} = \frac{\mu_{ma}{}^2}{\sigma_{Ma}{}^2 + \sigma_{Mn}{}^2 + \mu_{Mn}{}^2} \qquad (14)$$

where σ_{Ma} is the noise term associated with the variation in the value of the number of trained synapses on a winning piriform synapse (7), while μ_{Mn} and σ_{Mn} are the results of the naive or untrained synapses being solicited during the feedback or the inhibition cycle. As previously, μ_{Mn} and μ_w are mutually exclusive of each other and this contribution may be easily eliminated by masking (biological inhibition) or disallowing summation of those LOT lines inactive during the present minor cycle to contribute to the inhibition of level I. Note that μ_{Mn} represents a "DC" term which also can be readily removed by capacitive coupling. Substituting (10) through (12) into (13).

$$e_{SNR}=\frac{\dfrac{\mu_w^2 w_{\max}^2 p^2}{A^2}}{\left(\dfrac{\mu_w p}{A}\right)\left(\dfrac{a-\mu_w}{A}\right)w_{\max}^2+p_2 w_n^2\left(\dfrac{n-\mu_w}{N-A}\right)^2+p w_n^2\left(\dfrac{n-\mu_w}{N-A}\right)\left(1-\dfrac{n-\mu_w}{N-A}\right.} \qquad (15)$$

where after simplification

$$e_{SNR}=\frac{\dfrac{K_r\mu_w^2}{A^2}}{\dfrac{K_r\mu_w}{A^2 p}(A-\mu_w)+\left(\dfrac{n-\mu_w}{N-A}\right)^2+\left(\dfrac{n-\mu_a}{N-A}\right)\left(\dfrac{N-A-n-\mu_a}{N-A}\right)\dfrac{1}{p}} \qquad (16)$$

where K_r equals $w_{\max}^2 / w_n^2$ is the trained to naive weight ratio.

For A = 0.2N and n = 0.1N and after further simplification

$$e_{SNR}=\frac{25K_r\mu_w^2}{\dfrac{25K_r\mu_w}{p}(0.2N-\mu_w)+\dfrac{(0.1N-\mu_w)^2}{0.64}+\dfrac{(0.1N-\mu_w)(0.7N-\mu_w)}{0.64p}} \qquad (17)$$

The following observations can be made from equation (17) regarding the noise terms in the denominator. The first and the third terms are reduced by increasing the number of trials or patches in the piriform cortex. The second term can be removed by thresholding. Finally it is possible to remove the second and third terms by masking with M (the mitral activity) as previously noted. In the following discussion we will look at the two possible cases in more detail.

case 1: Threshold removal of term two reduces equation (17) to

$$\epsilon_{SNR}=\frac{25K_r\mu_w^{\,2}p}{25K_r\mu_w(0.2N-\mu_w)+\dfrac{(0.1N-\mu_w)\,(0.7N-\mu_w)}{0.64}} \tag{18}$$

for μ_w approaching n (increasing *h* see (7)), increasing *p* and larger K_r

$$\epsilon_{SNR}=\frac{\mu_w p}{(0.2N-\mu_w)}=\frac{p}{\left(\dfrac{0.2N}{\mu_w}-1\right)} \tag{19}$$

case II: Mask removal of the second and third terms

$$\epsilon_{SNR}=\frac{25K_r\mu_w^{\,2}p}{25K_r\mu_w(0.2N-\mu_w)} \tag{20}$$

and after simplification becomes

$$\epsilon_{SNR}=\frac{\mu_w p}{0.2N-\mu_w}=\frac{p}{\left(\dfrac{0.2N}{\mu_w}-1\right)} \tag{21}$$

From equation (18) we can observe that to achieve a better signal to noise ratio the value of k_r should be high and from equations (19) and (21) we can say that by increasing "p" we can achieve an improved Signal to Noise Ratio. It is also of considerable importance for us to ensure that the transistor mismatch errors of the electronic olfactory implementation does not affect the results or the performance of the olfactory model. In the following sections we have analyzed the transistor mismatch errors of the proposed hardware implementation, which will provide a brief insight of the effects of mismatch errors on the model results and performance.

Modelling of Mismatch in MOS Transistors

The mismatch of MOS transistors must be taken into consideration to achieve precise modelling of the above model. In general, when considering the mismatch in MOS transistors in an analog integrated circuit process, two variations are to be considered. The first one is the global or the interdie variations which account for the total variation in the value of the component over a wafer or a batch. Second is the local variation mismatch or interdie variations which reflect the variations in a component value with reference to an adjacent component on the same chip. As the design of presicion analog integrated circuits is based on component ratios rather than their absolute values, we must concern ourselves with the android variations. The major effects of mismatch in a MOS transistor are: the drain current mismatch which are due to the offset voltage or overdrive and β mismatch [50]. In the following sections we will develop a mismatch model and analyze the device mismatch errors of the building blocks of the electronic olfactory. This should help us in deciding whether the device mismatch errors affects the system performance.

Bidirectional Voltage / Current Buffers

Bi-directional voltage/current (BiVI) buffers based on the current conveyor concept permit bi-directional access to the weight matrix. They provide the dual functions of serving as voltage drivers and current sources/sinks to isolate the weight matrix in forward and backward modes respectively. Figure 2. shows the BiVI conveyor. It is clear from the figure

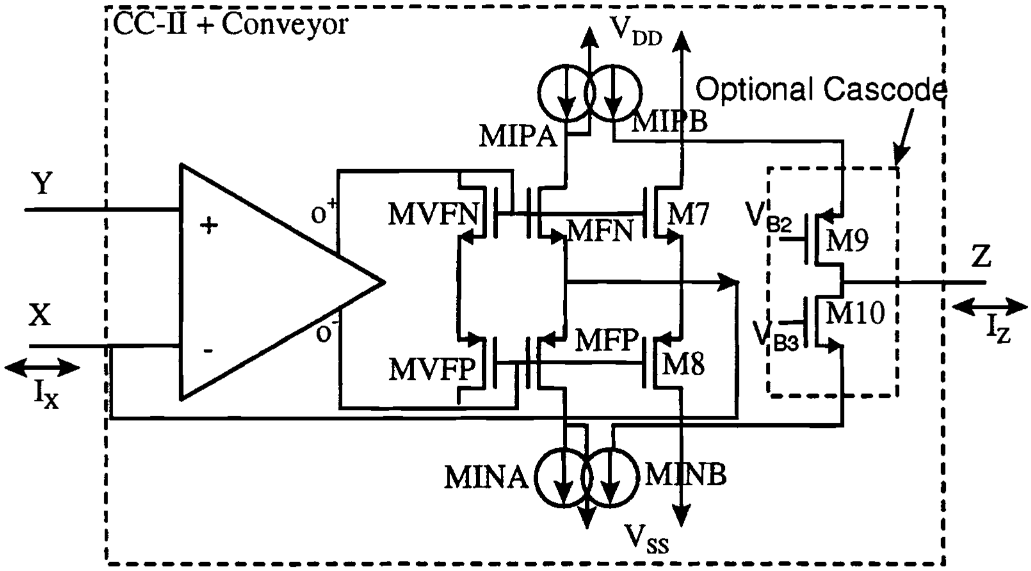


Figure 2. Bi-directional Voltage / Current Buffer

that the two major factors of the BiVI causing mismatch are the offset voltage of the amplifier and the current transfer error due to the current mirrors (MINA &MINB, MIPA & MIPB). Both the mitral and the piriform BiVI buffers contribute similar errors. These BiVI buffers perform inverse functions in the feedforward and feedback cycles which will be analyzed in the following sections.

Feed Forward Operation

In the feed forward mode the BiVI buffer at the mitral end acts as a Voltage controlled Voltage source (VCVS). Therefore the mismatch error associated with BiVI buffer when acting as a VCVS is just the offset voltage (V_{osm}) of the amplifier. Whereas the BiVI buffer at the piriform end acts as a Current controlled Current source (CCCS) in the feed forward mode. The mismatch error associated with the CCCS is the offset voltage due to the amplifier (V_{osp}) and the current transfer error of the current mirror (ΔI_p) as shown in Figure 3. The outputs of the mitral cells (M_{ij}), are projected onto the piriform cells in the piriform cortex via the LOT lines thus forming a connection matrix between the OB and the PC. The excitory synapses $W_{(ij)(kl)}$ have an associated mismatch error term ($\Delta W_{(ij)(kl)}$). Therefore the significant error terms in the feed forward operation are as follows:

(a) the offset voltage of the amplifier in the mitral buffers (V_{osm}) ,

(b) the offset voltage of the amplifier in the piriform buffers (V_{osp}),

(c) the current transfer error of the current mirrors in the piriform buffer (ΔI_p) and,

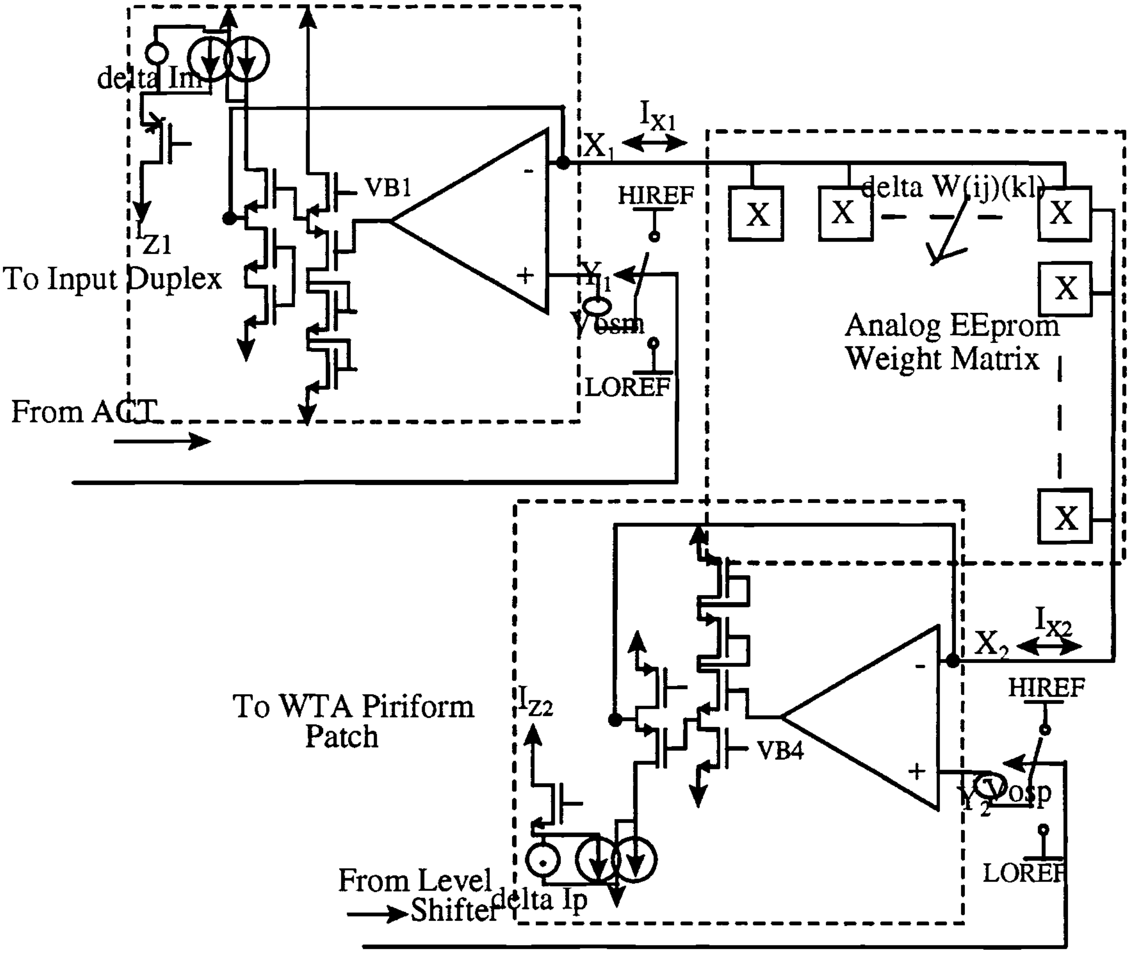


Figure 3. Bi-directional Voltage/Current Buffers across Weight Matrix

(d) the mismatch error term of the synapse ($\Delta W_{(ij)(kl)}$).

Feed Back Operation

In the feedback mode the BiVI buffers at the mitral side act as a current controlled current source (CCCS) and the BiVI buffers at the at the piriform side act as a voltage controlled voltage source (VCVS). The significant error terms of the BiVI buffers in the feed back mode from a similar analysis as previous are given as follows:

(a) the offset voltage of the amplifier in the piriform buffers (V_{osp}),

(b) the mismatch error term of the synapse ($\Delta W_{(ij)(kl)}$).

(c) the offset voltage of the amplifier in the mitral buffers (V_{osm}) ,

(d) the current transfer error of the current mirrors in the mitral buffer (ΔI_m),

Weight Matrix

The weight matrix sub structure of Figure 4 with four rows (M_{ij}) and five columns (P_{kl}) is shown. Each of the weight elements ($W_{(ij)(kl)}$) are modelled as resistors as shown in Figure 4. The equation for the weight interconnection transistor is as given below and it can be seen that there are two significant terms the mismatch in geometries ($\Delta\beta$), and the mismatch in their threshold voltages (ΔV_T).

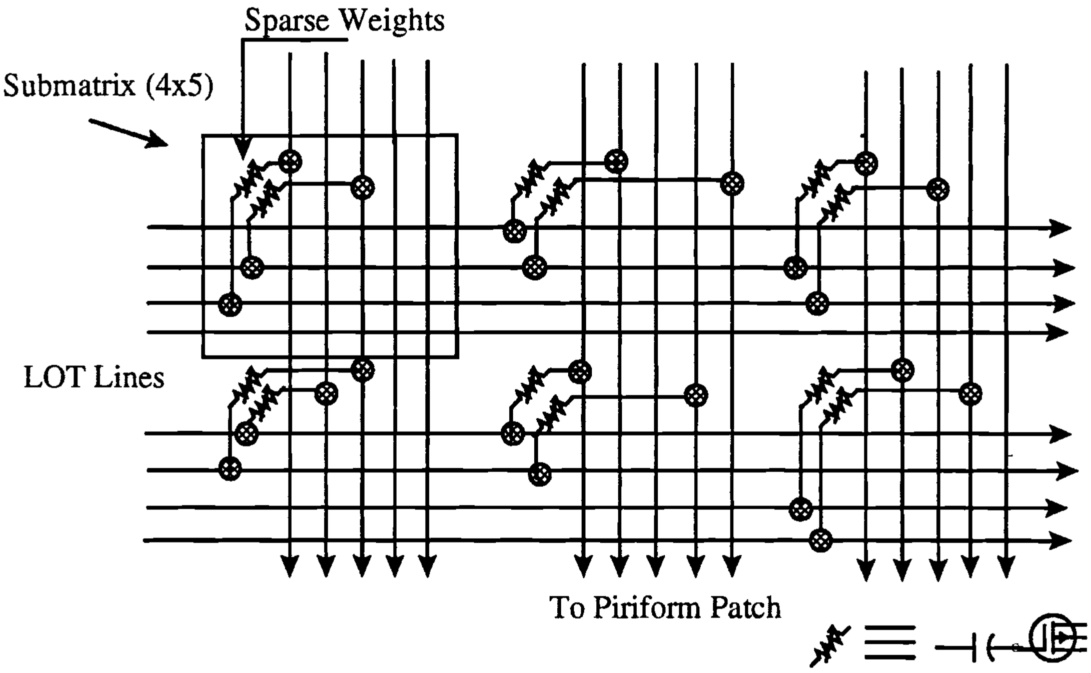


Figure 4. Weight Matrix Stucture

$$W_{(ij)(kl)}+\Delta W_{(ij)(kl)}=\frac{1}{(\beta\Delta V)\left(1\pm\frac{\Delta V_T}{\Delta V}\right)\left(1\pm\frac{\Delta\beta}{\beta}\right)} \qquad (22)$$

which can be simplified as follows

$$W_{(ij)(kl)}+\Delta W_{(ij)(kl)}=\frac{1}{\beta\Delta V}(1\pm\Delta V_T\pm\Delta_\beta) \qquad (23)$$

where

$\Delta\beta$ is the geometry mismatch of the interconnecting weight transistor ($W_{(ij)(kl)}$),

ΔV_T is the threshold voltage mismatch of the interconnecting weight transistor ($W_{(ij)(kl)}$).

Winner Take All Circuit

Figure 5 shows the winner take all circuit that is implemented in the proposed hardware model. The most significant sources of mismatch in the winner take all cell are as follows:

(a) the beta and threshold mismatch errors due to the comparison transistor MP2 of the winner take all cell and,

(b) the offset voltage due to the comparator

This analysis ignores β errors (area and oxide insulator) of MP4 and λ mismatch errors can be ignored due to the use of cascodes. By converting all these mismatch error terms into an

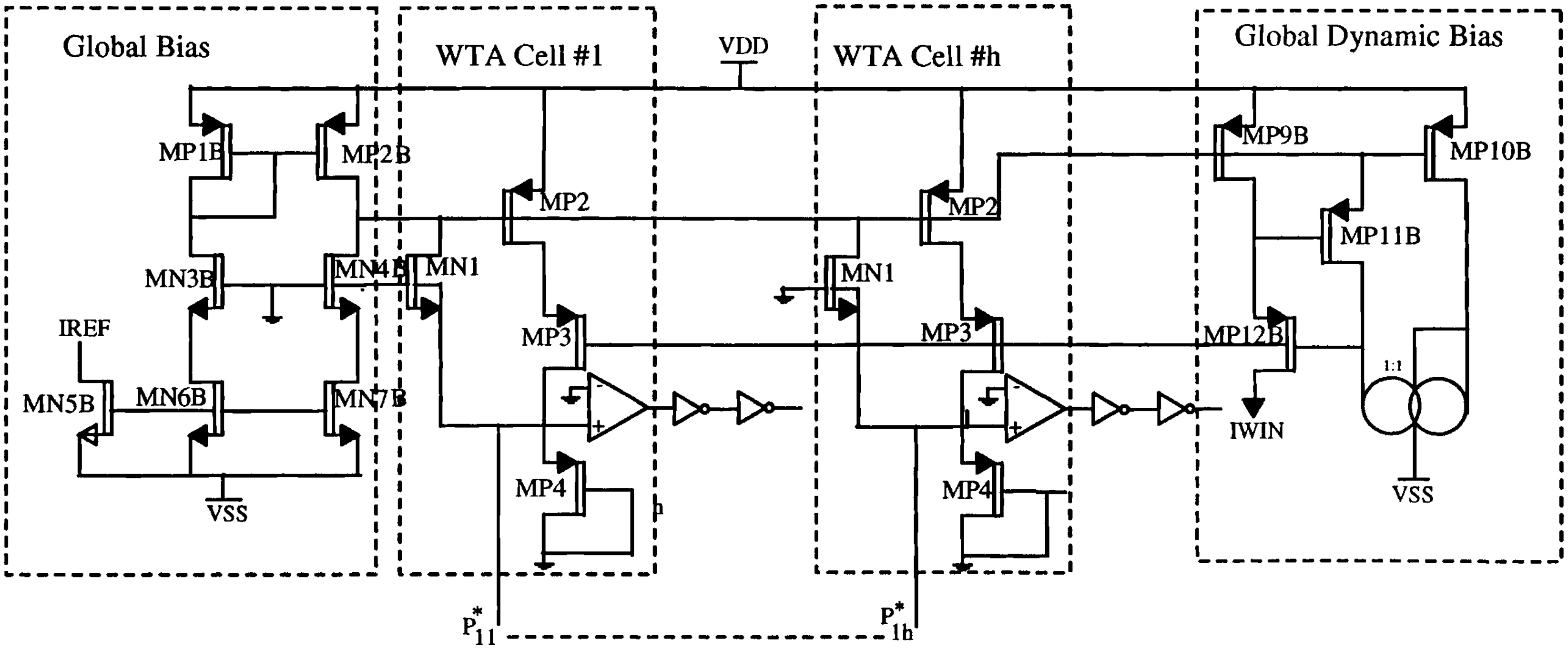


Figure 5. Winner Take All Circuit

equivalent input referred current error the equivalent error current of the winner take all circuit can be written as follows:

$$\Delta I_{WTA}=\left(\Delta I_{(mcc)}^{2}+\Delta\,\beta\,\Delta\,V^{2}+(V_{(os2)}^{\;2}g_{m2})^{2}\right)^{(1/2)} \tag{24}$$

where $g_{m2} = \surd\,(\,2\ \beta_2\ I_{in})$ and β_2 is of the transistor MP2

$\Delta I_{mcc} = \Delta I_{(mirror)}$ = current transfer ratio error of the current mirror composed of MP2 of the various winner take all cells.

Equivalent Current Error in Feed Forward Mode

The equivalent current error in the feed forward mode can be determined by taking into account the device mismatch errors of the bi-directional buffers of the mitral and piriform patches along with the weight matrix and winner take all mismatches. The equivalent current at the input of the piriform current conveyor can be derived from Figure 3 and is as shown below:

$$(I_p+\Delta I_p)=\left(\frac{W_{(ij)(kl)}}{W_n}+\frac{\Delta W_{(ij)(kl)}}{W_n}\right)\left(Vm_{(ij)}-Vp_{(kl)}+V_{osp}+V_{osm}\right)+\Delta I_{(p)} \tag{25}$$

where

$W_{(ij)(kl)}$ are the elements of the weight matrix

$\Delta W_{(ij)(kl)}$ are the error associated with the weight matrix

$Vm_{(ij)}$ are the voltage at the mitral end of the weight matrix

$Vp_{(kl)}$ are the voltage at the piriform end of the weight matrix

V_{osm} are the offset voltage due to the comparator at the mitral buffer

V_{osp} are the offset voltage due to the comparator at the piriform buffer

ΔI_p are the error associated with the current mirrors at the piriform buffers.

By defining a factor K as the number of times the weights have been trained, we can simplify equation (25) as follows:

$$(I_p+\Delta I_p)=KW_n\left(1+\frac{\Delta W}{KW_n}\right)V_{(MP)}\left(1+\frac{V_{osm}}{V_{(MP)}}+\frac{V_{osp}}{V_{(MP)}}\right)+\Delta I_{(mirror)} \tag{26}$$

where $V_{MP} = Vm_{(ij)} - Vp_{(kl)}$

The above equation gives the total current at the output of each piriform patch. The equivalent current mismatch in the feed forward direction can be obtained by adding the winner take all mismatch errors of (24) with the above equation. The equivalent current error in the feedforward direction and equivalent error term can be obtained by combining equations (24) and (26)

$$(I+\Delta I)_{FF}=KW_n\left(1+\frac{\Delta W}{KW_n}\right)V_{(MP)}\left(1+\frac{V_{osm}}{V_{(MP)}}+\frac{V_{osp}}{V_{(MP)}}\right)+\Delta I_{(p)}+\Delta I_{(WTA)} \tag{27}$$

By a similar analysis the equivalent current mismatch error can be derived for the feedback inhibition current and is as shown below

$$(I+\Delta I)_{FB}=KW_n\left(1+\frac{\Delta W}{KW_n}\right)V_{(MP)}\left(1+\frac{V_{osm}}{V_{(MP)}}+\frac{V_{osp}}{V_{(MP)}}\right)+\Delta I_{(m)} \tag{28}$$

For statistical modelling of the mismatch errors a normally distributed random number in the range of 2-5% of the original value is added to the original value. This 2-5% device mismatch error is the total error of the building block which takes into account the β error and also the typical threshold offset voltage errors of ±10mv. The values of 2-5% were chosen based on the results of previous device fabrications. The results of the simulations are analyzed in the next chapter, and conclusively demonstrate that transistor mismatch errors have no significant effect on the results and performance of the model.

CHAPTER III

SIMULATION AND ANALYSIS OF THE OLFACTORY MODEL

This chapter describes a software implementation of the statistical model presented in the previous chapter by software. Two source codes, one in Matlab and the other in Fortran with minor implementation differences were written for our implementation of the modified GLA model and validated. Analysis of the training/clustering results of the simulation are summarized along with the model validation.

In this chapter, the model is validated with respect to the following statistical properties

(a) Distribution of the synapses in the sparse matrix,

(b) Distribution of the number of active synapses in a piriform neuron,

(c) Distribution of the magnitude of the winning piriform neuron,

(d) Relative mean and variance of the trained winner,

(e) Distribution of the inhibition response and,

(f) Distribution of inhibition response with mismatch errors taken into consideration.

(g) Effective SNR of I*

Software Model

The flowchart of the Software implementation of the model is shown in Figure 6. From the flowchart we can infer how the function of the different modules in the software implementation of the GLA olfactory model interact. The table 1 on the following page lists the nomenclature used in the software code and its corresponding usage in the model.

Trial input vectors are the normally distributed random numbers generated between (0 - 1). These form the input vectors O (i) as represented in the flowchart. In our case the input O (i) consists of 40 uniformly distributed random numbers. In order to have 20% activity as specified in the model in chapter II, we sample without replacement 8 of the 40 vector elements, thus allowing a maximum of five minor cycles. The magnitude and position of the each grouping of 8 vectors were determined by selecting the eight largest in magnitude and replacing the others by a zero. In a similar manner for each cycle, the eight remaining largest input vectors are selected and all other vectors(elements) are set to zero. This process is continued until all five groups of sparse vectors (20%) have been presented to the mitrals. In the subsequent cycles (for 2,3,4,5) the input cycle vectors have to be normalized. This is done by selecting the maximum value element in each cycle(20%) and scaling it to 1 and the other elements are normalized or scaled by the reciprocal of the maximum value element. These inputs after normalization are then applied to the thermometer coder. Program *Input.m* shows how the input vectors are classified into different cycles and the logic by which they are normalized. In this program the trial input vectors O_i are classified based on 20% activation into 5 cycles. They elements of these

S.No	Model Nomenclature	Software Nomenclature
1	g	g
2	m	m
3	p	p
4	h	h
5	O_i	O_i
6	W	r1
7	Δw	dw
8	W_T	r3_v
9	θMj	theta
10	P^*_{kl}	pw
11	P_{kl}	is4
12	PW_{kl}	is7
13	G_i	G_i
14	$I^*(ij)$	iinhib
15	$I^*(i)$	fif
16	θ_I	thresI
17	i	i
18	j	j
19	k	k
20	l	l
21	ij	x
22	kl	y
24	w_{max}	w_{max}
25	w_n	w_n
26	μ_{Ma}	f1

27	σ_{Ma}	f2
28	μ_{Mn}	f3
29	σ_{Mn}	f4
30	μ_{w}	f5
31	σ_{Mn}	f6
32	G_i^*	G_i^*
33	a_w	pw
34	$\Delta w_{(ij)(kl)}$	sperr(x,y)
35	V_{osm}	m3
36	V_{osp}	pwerr
37	K_G	k

Table 1. Equivalent Nomenclature for software/model

cycles are called $G_{i.}$ These cycles are then normalized and stored as G_i^* (gistar).

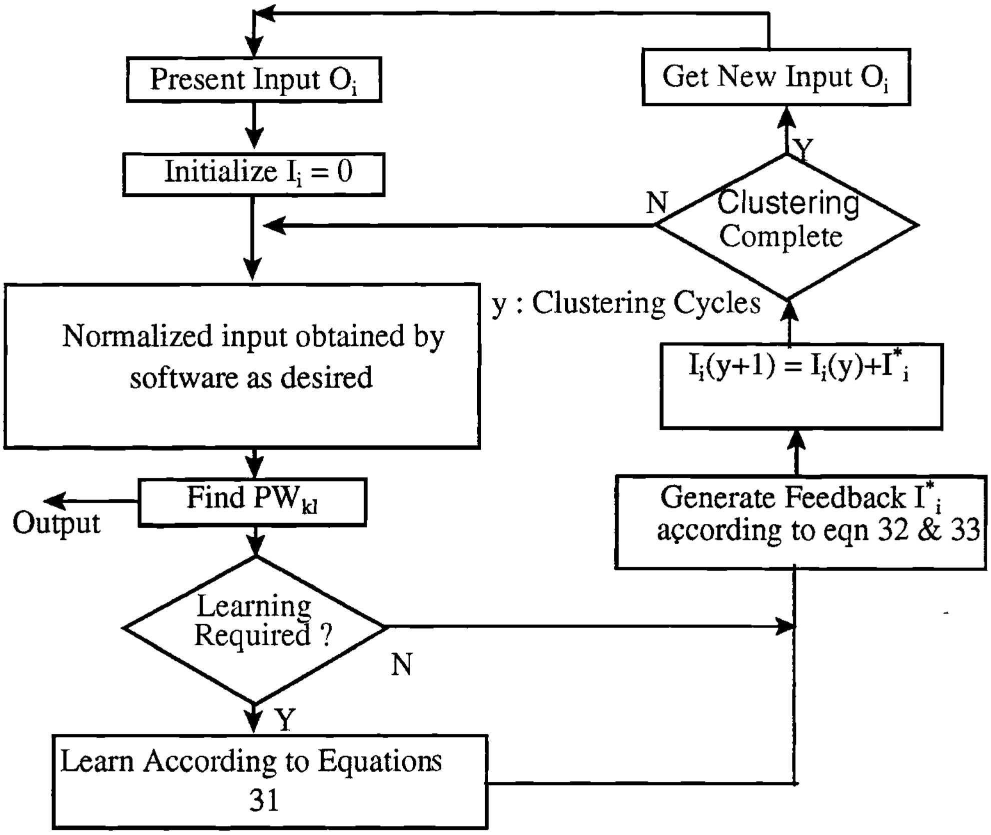
The initial conditions that were assumed for the software model of Figure 6 are as follows:

(a) $I_i = 0$ $(1 < i < g)$

(b) $P_i = 0$ $(1 < i < p)$

where I_i and P_i are the integrated inhibition signal into the glomeruli " i " from cortical feedback and the piriform output of the " i " patch respectively.

The complete program (*olfactory.m*) is written in such a way that by changing the values of the dimension parameters "g","m","p"and,"h", the program can be adapted to any size olfactory system. It automatically generates the properly dimensioned weight matrix and all other corresponding vectors. For our simulations we have chosen the size of "g", "m",



Present Input O_i
Get New Input O_i
Initialize $I_i = 0$
Y
N
Clustering Complete
y : Clustering Cycles
Normalized input obtained by software as desired
$I_i(y+1) = I_i(y)+I^*_i$
Find PW_{kl}
Output
Generate Feedback I^*_i açcording to eqn 32 & 33
Learning Required ?
N
Y
Learn According to Equations 31

Figure 6. Flow Chart of the Olfactory Model

"p", and "h" to be 40, 16, 40, and 16 respectively. These dimensions were selected based on the olfactory model by P. A. Shoemaker, C. G. Hutchens and S. B. Patil [4]. This results in a sparse weight matrix of 640 X 640 with a probability of interconnect (synapses) of 0.1. The placement of synapses is random and normally distributed with a naive value of 1. This is implemented by the matlab command "sprandn(640,640,0.1)", which results in a sparse matrix of dimension 640 x 640 with a probability of interconnect 0.1. The command "spones (sparse weight matrix)" will result in a sparse matrix with naive value of 1.

Initially, the first cycle of inputs, (20%) of the most significant input components are applied to the thermometer coder which is essentially A/D conversion without encoding. This is implemented in the software by having "m" number of equidistant values from 0 to 1. The matlab command for the above is Theta = 0:1/16:1. The mitral cells or excitory neurons are modelled as two state devices which are either quiescent (not firing) or active (firing at maximum rate) with glomerulus activity above or below its threshold activity. This results in a mitral output "m2" of dimension "g x m" which is either "0" or "1".

m2(i,j) = 1 if g(i) ≥ theta

= 0 otherwise

This mitral output is then applied as input to the weight matrix (r1). In the event, one is taking the transistor mismatch errors into consideration then a random number (called "m3") varying between 2-5% of m2 (output of the mitral patch) is added to the mitral outputs which accounts for the BiVI buffer errors as explained in chapter 2. Therefore the mitral output taking into considerations the device mismatch errors are m2=randn(0.02 - 0.05)*m2+m2. Another 2-5% of the naive weight value is added to the synapses (called "sperr" and stored in "sperr.mat"). Therefore the final value of the weight matrix will be equal to the sum of the sparse weight matrix (r1) and sperr. Finally, another 2-5% of the piriform input value (pw) is added as noise to the piriform inputs. The outputs of the mitral is then projected onto the piriform cells in the piriform cortex via the LOT lines thus forming a connection matrix between the OB and the PC. This is implemented in the program in the following manner:

$$pw=\sum_{i-1}^{g}\sum_{j-1}^{m} m2_{(ij)}r1_{(ij)(kl)} \tag{29}$$

See Appendix A for the program listing. For a detailed discussion of the transistor mismatch model see chapter II.

The output of the piriform cortex is then passed onto as input to the winner take all module. In this module, the program tries to resolve the winner among the patches which was done by grouping the piriform output in batches (batching of the patches is a software constraint) of "h" and selecting the one with maximum activation.

Depending on the output of the winner take all "is7", the mitral outputs "m2", and the synapse placement learning is implemented, which occurs by adjusting the weights between two nodes (activation on both ends are mandatory for learning) with an initial value of w_n (naive weight), and maximum value of w_{max} (saturated value) by a value of dw (increment--al value of the weights). Synapses with strength zero cannot change and remain at zero there after. Learning of the weights is implemented based on the following conditions:

$$
\begin{aligned}
rl_{(ij)(kl)} &= \min\left(rl_{(ij)(kl)}+dw, W_{\max}\right) \quad \textit{if } rl_{(ij)(kl)} \neq 0,\ m2_{ij}>0,\ is7_{kl}>0 \\
rl_{(ij)(kl)} & \qquad \textit{otherwise}
\end{aligned}
\tag{30}
$$

The naive value of the weights was initially set to 1 and with their maximum value as 3.2. The value of the weights saturate once they have reached the maximum value. The weights are incremented in steps of 0.4. All three of the above parameters can be changed as need dictates.

The output of the winner take all is fed back to the trained transposed weight matrix. The feedback then selectively inhibits the mitral cells in those bulb patches which are most responsible for cortical output response via long lasting inhibition. The weighted inhibition (iinhib) on LOT line (ij) in the backward direction is implemented according to the following equation:

$$
iinhib_{ij} = \sum_{k=1}^{p} \sum_{l=1}^{h} PW_{kl} rl_{(kl)(ij)} \tag{31}
$$

Unthreshold feedback inhibition"iinhib" on consecutive m LOT lines in the backward direction is summed by grouping them together as given by:

$$
fif_i = \sum_{j=1}^{m} iinhib_{ij} \tag{32}
$$

The inhibitory feedback into the glomerulus is obtained by thresholding it with "thetaI". The program directly evaluates the mean (μ_w) and variance (σ_w) of the number of synapses in a winning piriform neuron, mean (μ_{Ma}) and variance (σ_{Ma}) of the signal and the mean (μ_{Mn}) and variance (σ_{Mn}) of the noise due to the naive weights.

Analysis of Simulation Results

Experiment 1-- Distribution of the Winning Piriform cells

(a) The sparse matrix generated by matlab program is first analyzed for its sparsity and distribution of synapses per column. This was done by summing up the number of interconnections in each column and then plotting a histogram of the same (as shown in figure 7). From the figure we can see that placement of synapses is random and is normally distributed, which corresponds with the assumptions made in the model. Further the number of non-zero elements of the sparse matrix was found to be 38,887 which is approximately equal to 0.1 x 640 x 640, which validates the condition for the 10% sparsity of the model.

(b) The 40 random numbers generated from 0 to 1 form the trial vectors O(i), which are then formed into 5 cycles (20% active) of Gi. In this experiment we apply the first cycle of inputs to the system and examine the outputs of the weight matrix. The weight matrix can be either a learned or naive, this does not affect the distribution of the number of active synapses or the magnitude of the winning piriform. All the other parameters g,m,p,h are the same through out the simulations unless otherwise specified. The output of the weight matrix

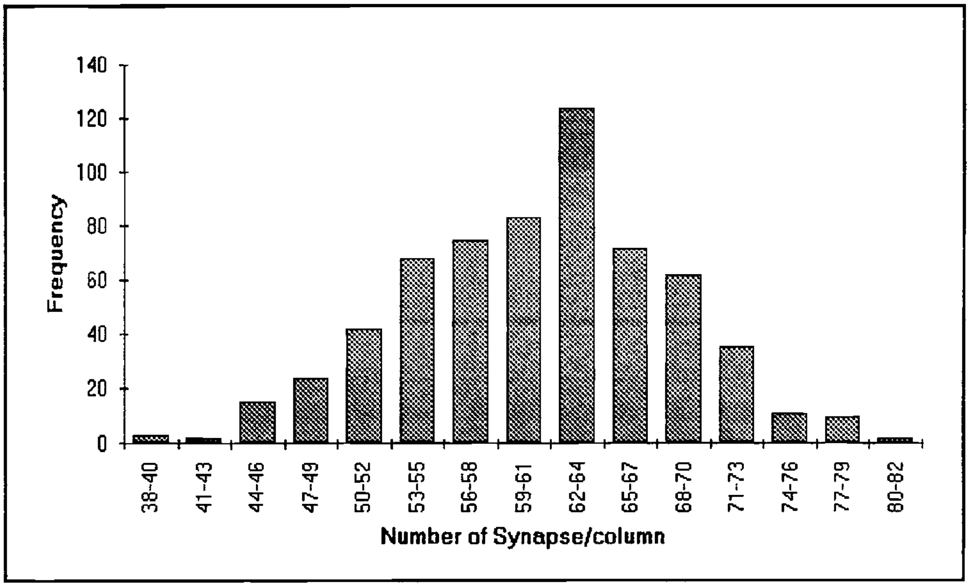


Figure 7. Distribution of active synapses in a sparse matrix

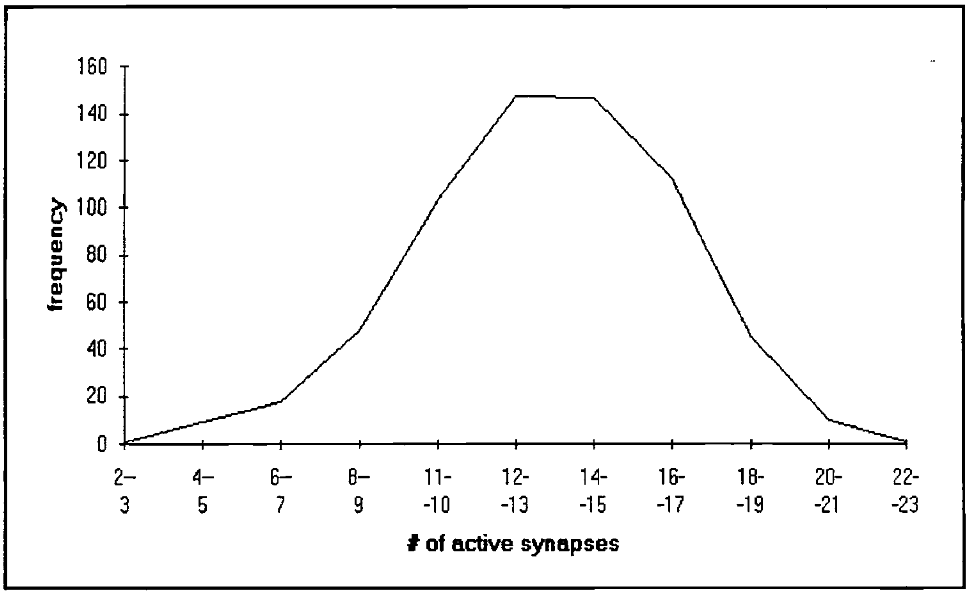


Figure 8. Distribution of active synapses in a piriform neuron

(pw) is obtained according to the equation (29) in chapter 3. The histogram of the result which is stored in temp5.mat is then plotted and is appears to be hypergeometrically distributed as shown in figure 8. From the equation (5) in chapter 2, we know that the number of active synapses follows a hypergeometric distribution when plotted and for a large N (number of LOT lines) it tends to be normally distributed. As the results of the model and simulation appears to be the same, the model is validated in this regard. The mean and variance of the active synapses of the piriform neuron, predicted by equations (3) and (4) are 14.10625 and 9.8259 respectively. The mean and variance of the active synapses in the piriform neuron from simulations were found to around 14.4 and 10.2 respectively from 30 trials constituting 40 samples in each trial.

(c) The above experimental data set and parameter values are again used for finding the distribution of the magnitude of the winning piriform. The output of the weight matrix whose results was analyzed previously is used as input to the winner take all. In this module, the outputs of the weight matrix is grouped in terms of "h", because we have "h" piriform cells per patch. Then as we are having "m" piriform patches with "h" piriform cells per patch, the winner for each of the "h" piriform patches are found. This is done by selecting the piriform cell with the maximum value in each patch. The results (is7) stored in the temp14.mat are then plotted. It is observed from the figure 9, that the distribution of the magnitude of the winning piriform follows a skewed distribution as predicted and we can see how it differs from the distribution of active synapses. From the equation (7) in chapter 2, we can observe that the magnitude of the winning piriform also appears to follow a skewed

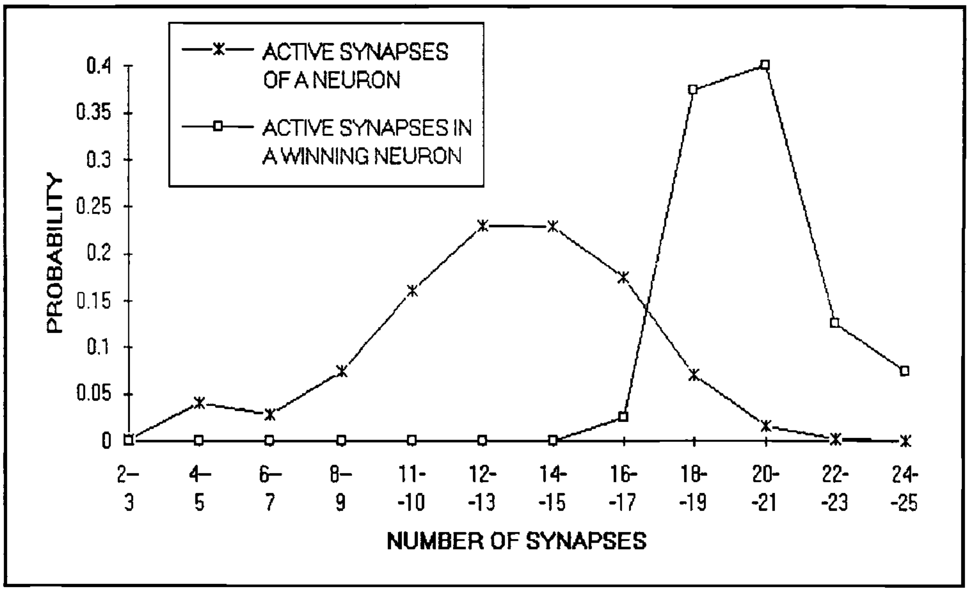


Figure 9. Comparison of the distribution of the active synapses in a neuron with winning neuron

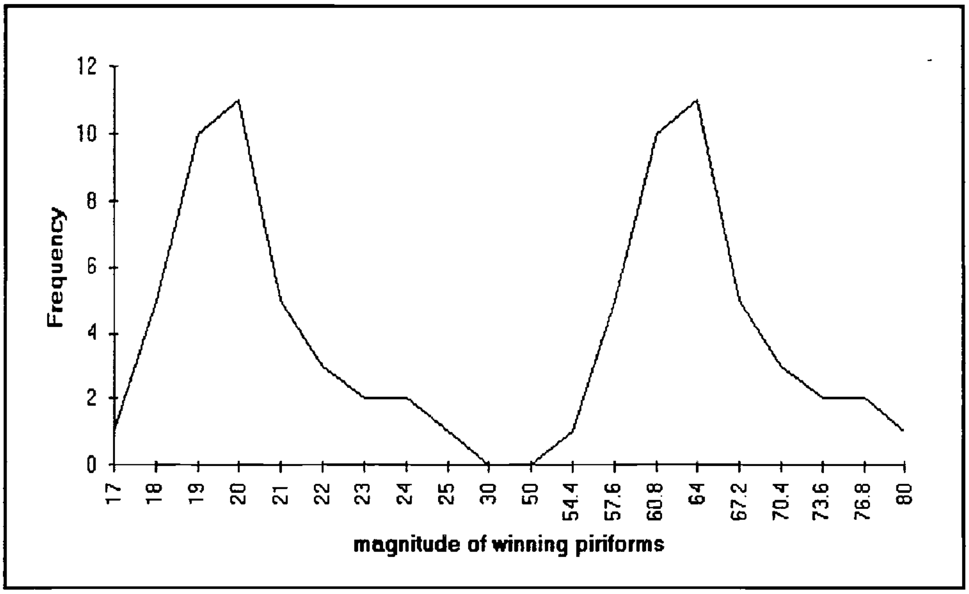


Figure 10. Comparison of the distribution of the magnitude of the winning piriform for naive/trained weight matrix

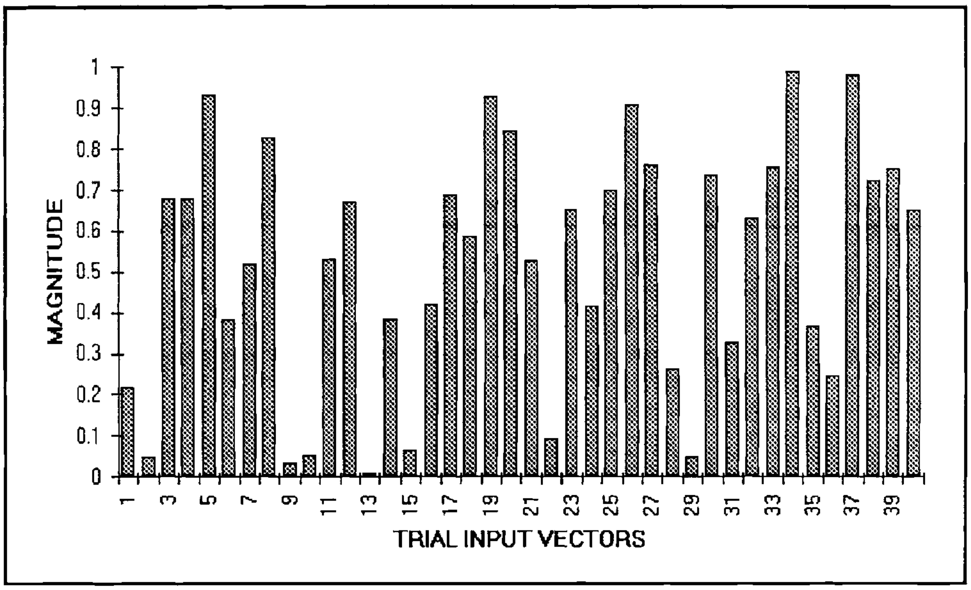


Figure 11. Trial input vectors

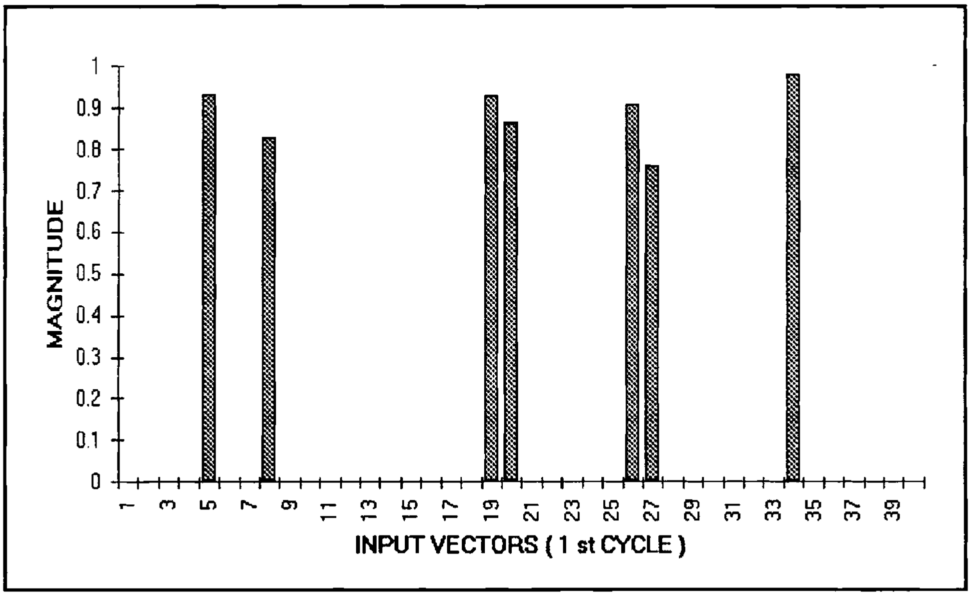


Figure 12. First cycle input vectors

distribution when plotted. The result seems to validates the model. However, as explained in the previous chapter, the mean and variance of the magnitude of the winning piriform neuron cannot be evaluated in a closed form. But, from the simulation results we calculated the mean and variance to be approximately 21.8 and 6.25 respectively. Some variation is due to the choice of the original sparse matrix. This was again calculated from 30 trials with 40 samples per trial. Figure 10, shows that the distribution of the magnitude of the winning piriform of a trained matrix is shifted and well out on the tail of the untrained one. Having analyzed the results of the winner in the feedforward direction in this section, the next experiment will analyze the distribution of the feedback inhibition response in the following sections.

Experiment 2 -- Distribution of Inhibition Response

(a) Naive Matrix The trial Oi vectors (40) were divided into a maximum of five cycles, each cycle containing 8 vectors to maintain 20% activity. The values of mph and all other parameters are maintained the same as experiment 1. Inputs used are the cycle 1 input vector data and the naive weight matrix. The feed forward winner take all outputs obtained (similar to experiment 1) are applied to the transposed weight matrix, which results in a output as given by the equation (30) in chapter 3. The Unthreshold feedback inhibition I_i^* on consecutive LOT lines are summed by grouping them together as given by equation (31) in chapter 3. The simulation results for the trial vectors O(i), first cycle inputs, and the corresponding Unthreshold feedback inhibited currents outputs are shown in figure 11 to

SNR = 4.2DB

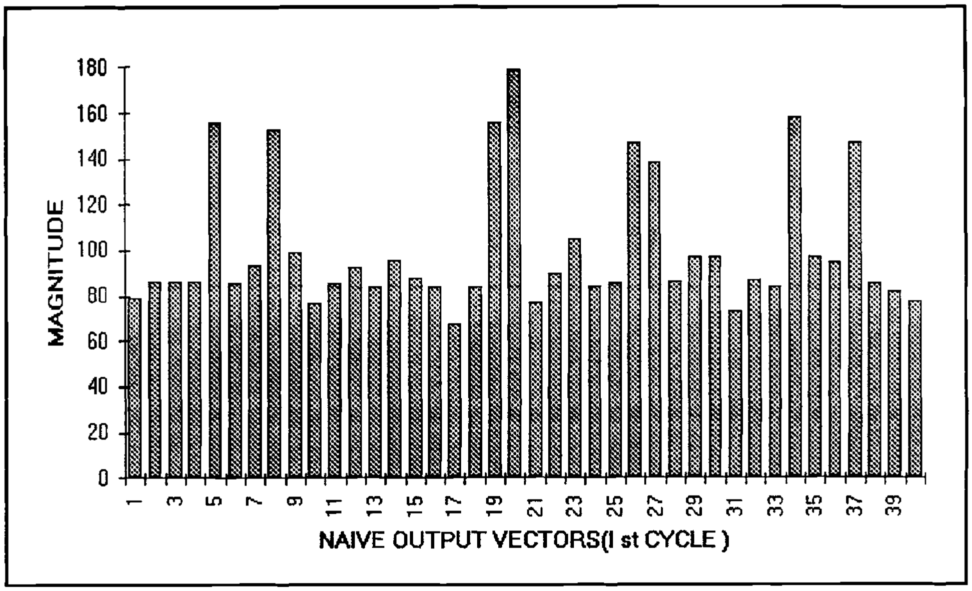


Figure 13. Output inhibited vectors (1 cycle -- naive)

SNR = 11.8DB

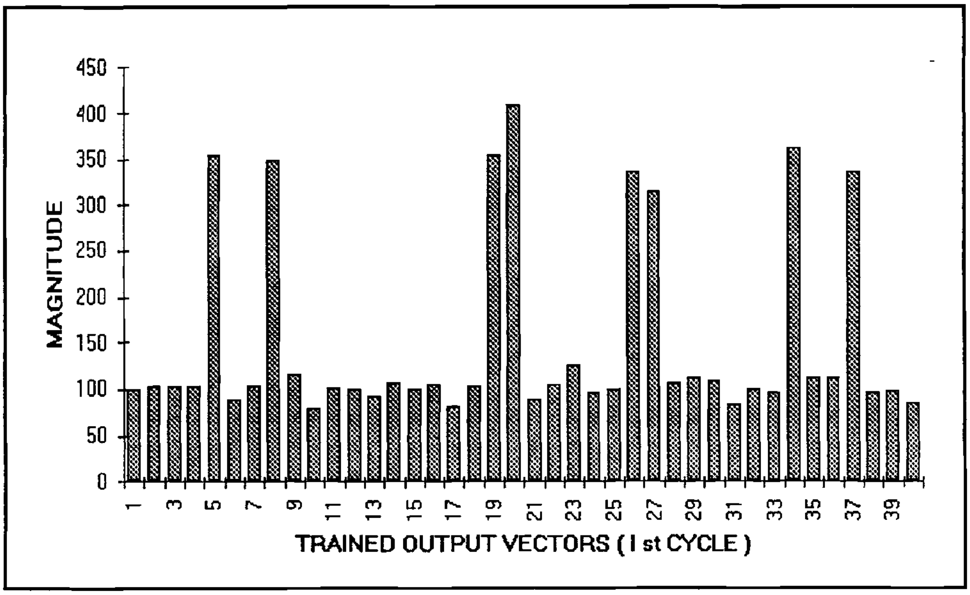


Figure 14. Output inhibited vectors (1 cycle -- trained)

SNR = 144.2DB

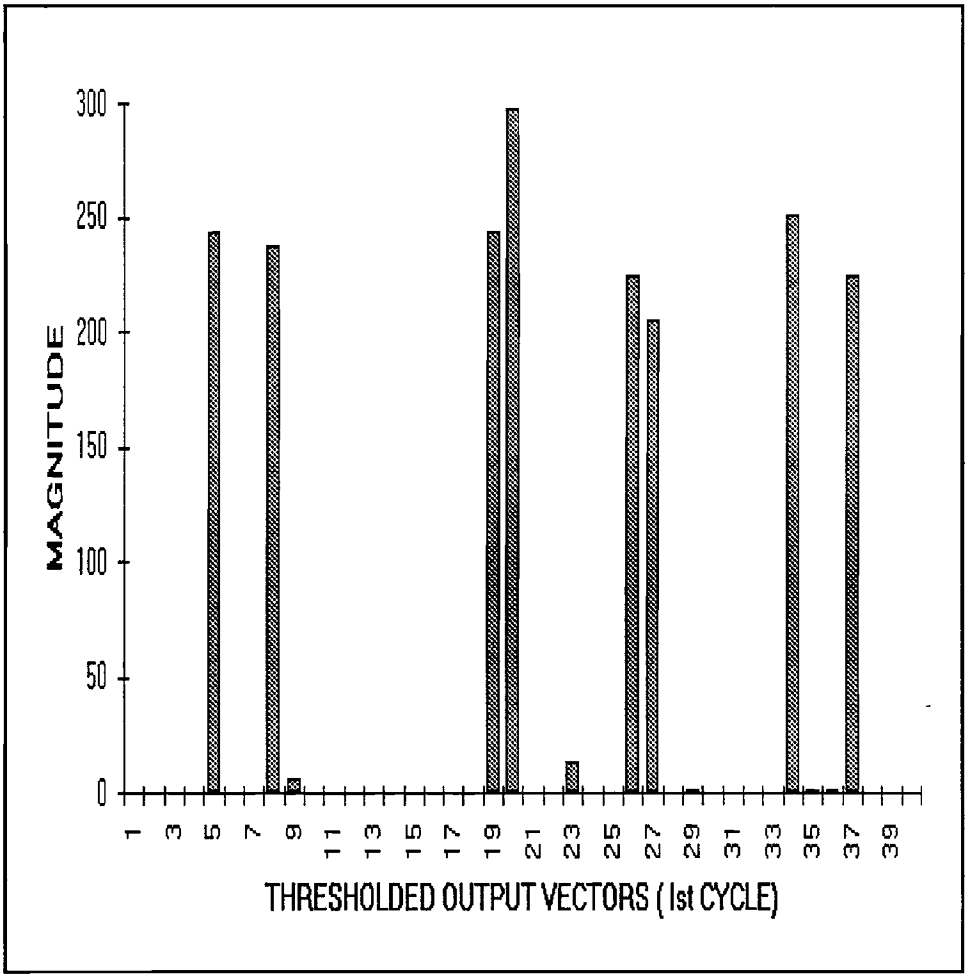


Figure 15. Thresholded feedback inhibited output vectors (1 cycle)

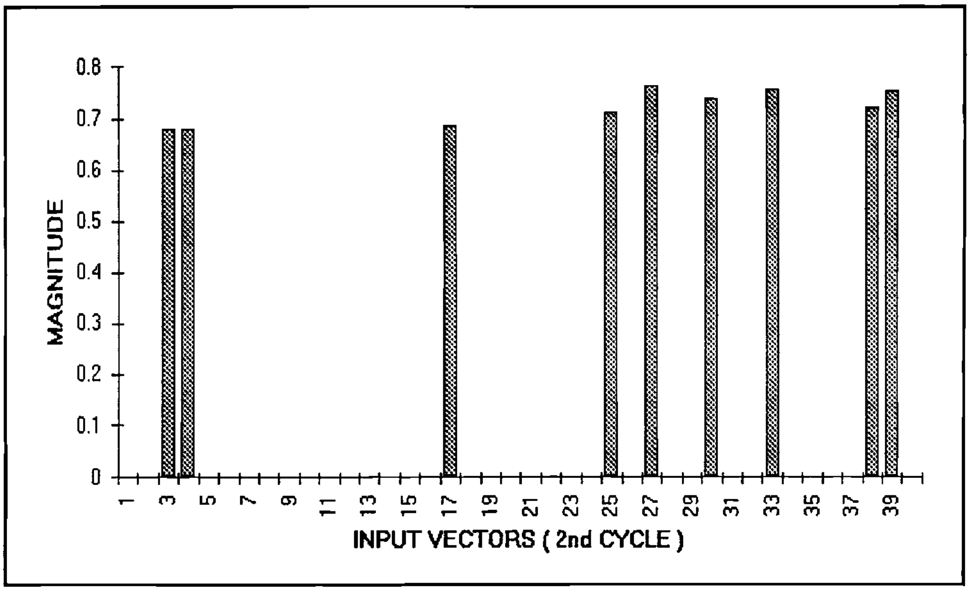


Figure 16. Input Vectors (2 cycle)

SNR = 3.6DB

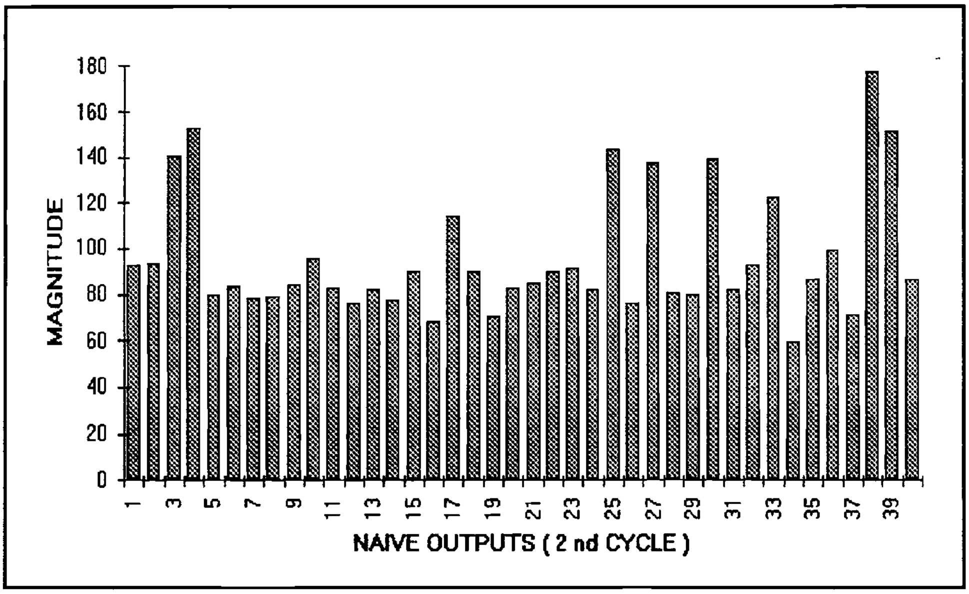


Figure 17. Feedback inhibited output vectors (2 cycle -- naive)

figure 13. The signal to noise ratio calculated from simulation results according to equation (15) in chapter 2 is 4.8DB. Although the system is able to recognize and classify the patterns a minimum signal to noise ratio of 6DB is desired.

(b) Trained Matrix In this experiment all the parameters and values are the same as experiment 2a, except that instead of a naive weight matrix we use the trained weight matrix (the updated weight matrix of the previous experiment 2a.). For the same input trial and cycle vectors the simulated output results are shown in figure 14. The signal to noise ratio for the output vectors were calculated as before, and were found to be 12.8DB. So from the above observation we are able to establish the fact that the matrix training results in a improved signal to noise ratio and that the SNR is directly proportional to the ratio of W_{max} / W_n . This corresponds with the equation (18) of chapter 2, which states that SNR is proportional to K_r .

(c) Thresholding The same parameters, trial vectors and input cycle vectors are used as the previous experiments. A fully trained weight matrix is used in this experiment. The simulations are carried out as before and the unthreshold feedback inhibition summed vectors are then thresholded, and thereby eliminating the DC noise that was present in the previous outputs. From the thresholded feedback inhibitory outputs shown in figure 15, we see that the total noise term present in the previous outputs is totally eliminated. This was done by arbitrarily setting the threshold value equal to $\mu_{Mn} + \sigma_{Mn}$. No attempt has been made to optimize SNR. We see that the total DC noise due to the naive weights is eliminated by

SNR = 10.3DB

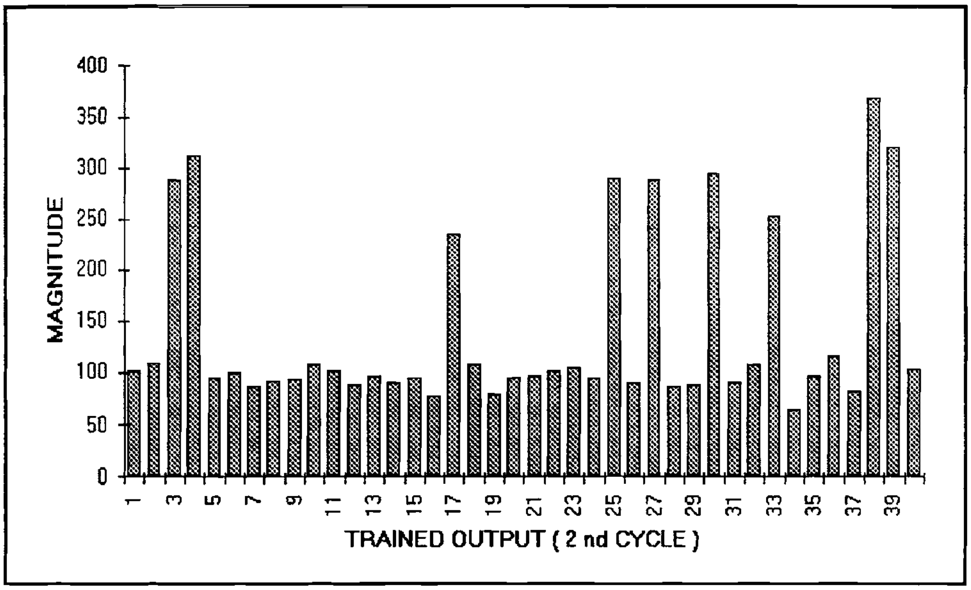


Figure 18. Feedback inhibited output vectors (2 cycle -- trained)

SNR = 52.1DB

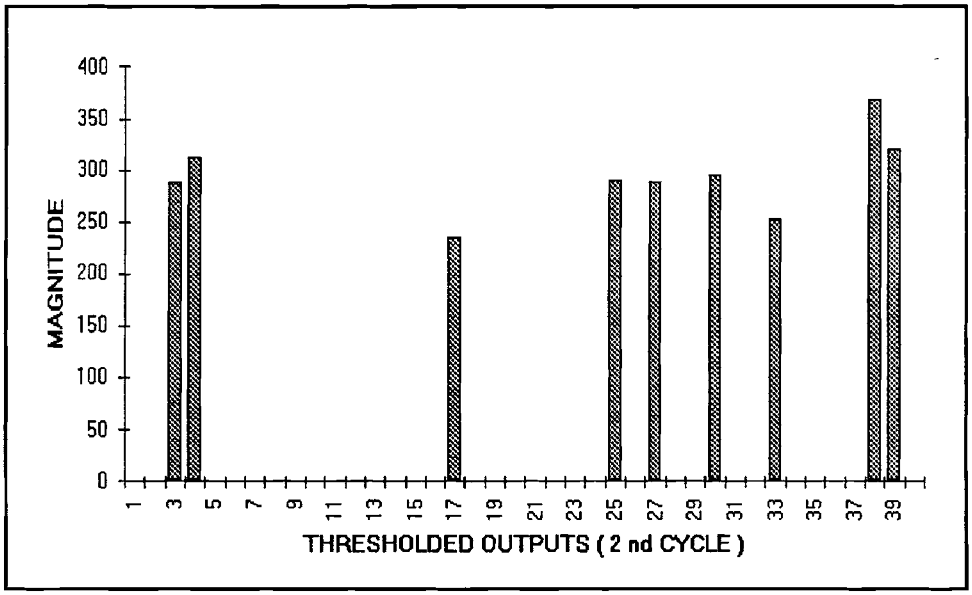


Figure 19. Feedback inhibited output vectors (2 cycle -- thresholded)

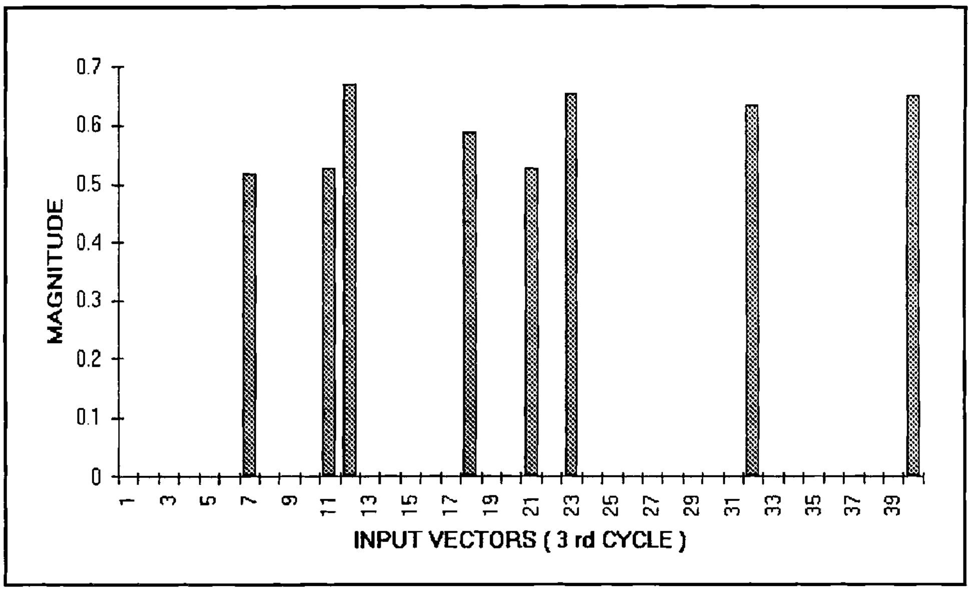


Figure 20. Input vectors (3 cycle)

SNR = 3.1DB

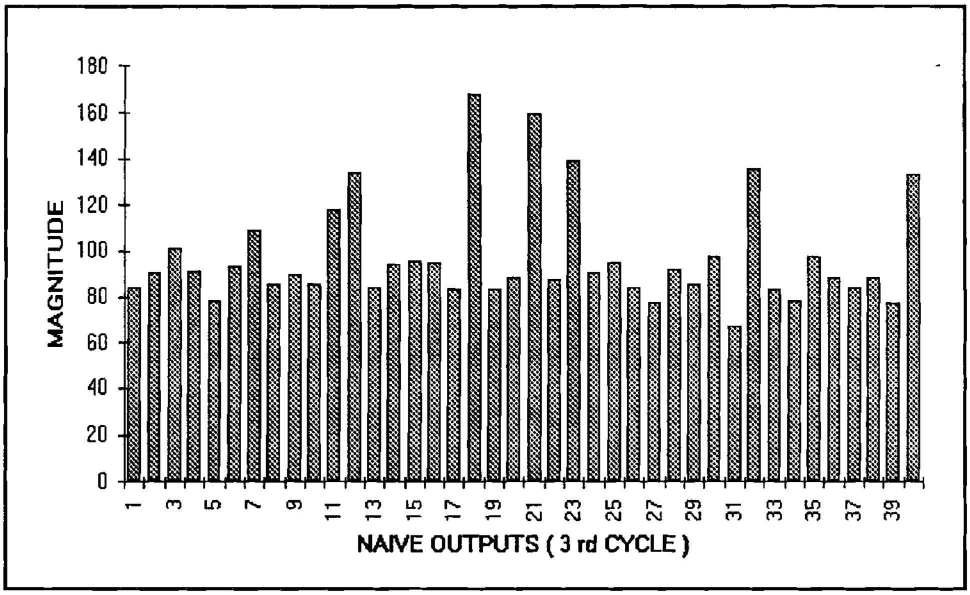


Figure 21. Feedback inhibited output vectors (3 cycle -- naive)

SNR = 8.2DB

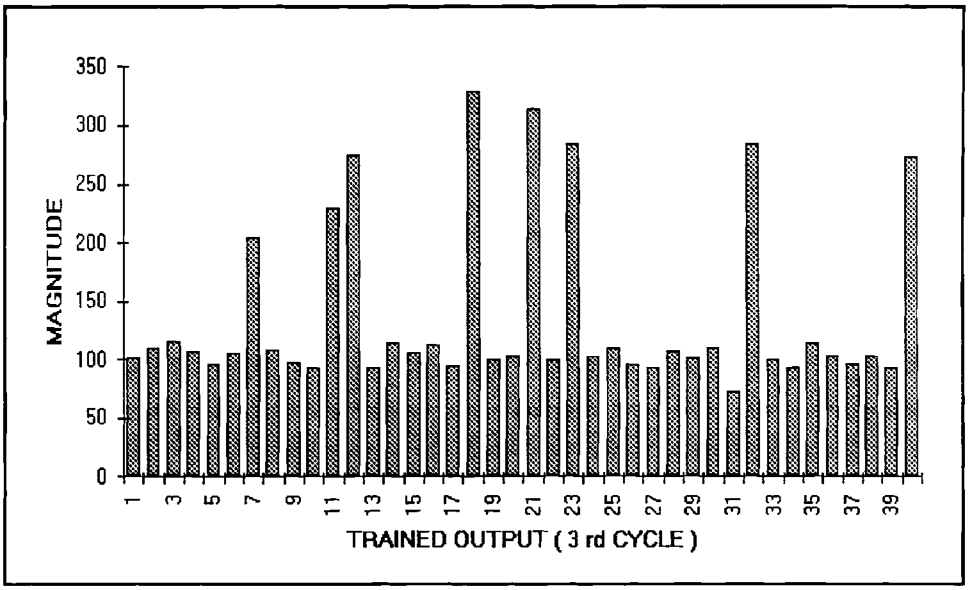


Figure 22. Feedback inhibited output vectors (3 cycle -- trained)

SNR = 44.8DB

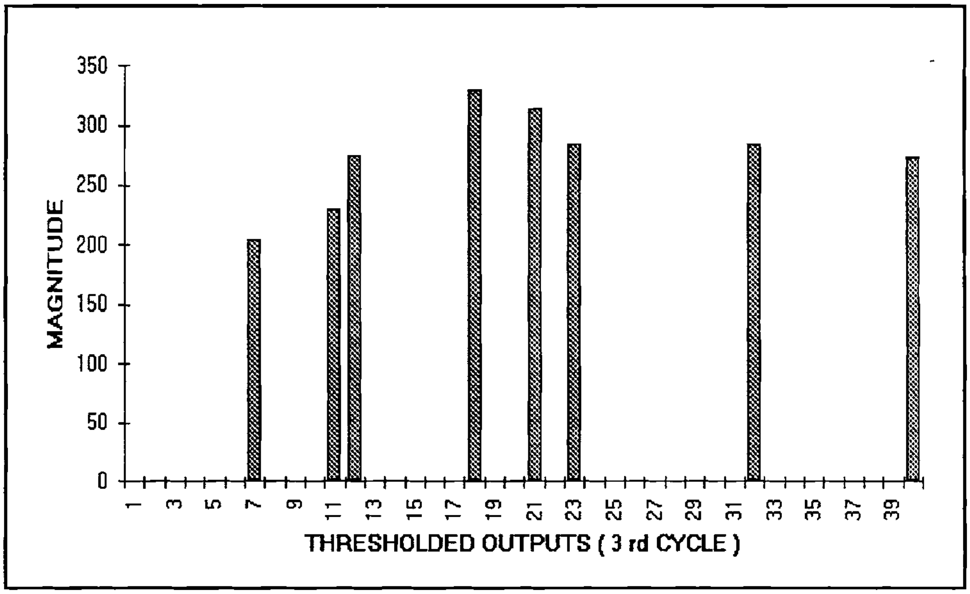


Figure 23. Feedback inhibited output vectors (3 cycle -- thresholded)

fixing the threshold value as specified by the model. The noise due to the standard deviation of the magnitude of the winning piriform can be reduced only by increasing the number of piriform patches "p" or by increasing the number of trials.

The above experiment is repeated for cycles two and three and the feedback inhibitory outputs for both naive and trained matrix and thresholded are shown in figures 16 to 23. The corresponding signal to noise ratios are calculated and are shown in table 2. From the values we can see that the signal to noise ratio continues to decrease with the increase in the number of cycles, for a given sparse weight matrix and other fixed size dimensions. The reason for the decrease in SNR with the increase in number of cycles is not clearly understood. Additional statistical experiments and analysis have to be done in this regard.

Cycle Number	Signal to Noise Ratio
1	12.9
2	10.38
3	8.2

Table 2. Cycle Number Vs SNR Values

To statistically validate the results that were obtained from simulations, a 90% confidence interval test was done. This was done to ensure that the simulated mean of the number of active synapses of a winning piriform is within one standard deviation of the theoretical value 90% of the time, 30 iterations were done. The results are tabulated in the appendix c. The iterations were done by taking random sparse matrix and random first cycle inputs every time. The table gives the corresponding SNR values obtained for every iteration. It can be observed from the tabulated values that the mean of the number of trained weights on a winning neuron is almost constant, hence statistically validating the results.

Another important observation that can be made from the simulations is that for a desired SNR, number of exemplars, and a given memory capacity (total number of weights), it is essential to find out what is the optimal value of the piriform patches and number of piriform cells per patch. Finding the optimum value of the piriform patches and piriform cells is beyond the scope of this thesis. However the observation that the percentage of ties increase with the decrease in number of piriform cells/patch can be made from the limited experiments conducted and summarized in table 3. Further statistical experimentation must be completed with the model to confirm the variation in the percentage of ties and, SNR.

# of Piriform Patches	# of cells/Patch	% of ties
64	10	17
40	16	7.5
32	20	6.25

Table 3. Sizing of Piriform Patches/Cells

Experiment 3 -- Mismatch Error Analysis

The device mismatch error analysis was done by adding a random number which is

2-5% of the value for the Bidirectional voltage/current buffers (all the error terms lumped together), and a random value of 2-5% of the naive value to the sparse matrix according to the model presented in the previous chapter. The same trial vectors and first cycle input vectors were presented as inputs. The mismatch error terms were added to the mitral outputs and to the sparse weight matrix. These changes were implemented in the program errors.m. The other parameters and dimension values are the same as the first two experiments. Figures 24 and 25 shows the feedback inhibited output values with and without transistor mismatch. A comparison of the thresholded outputs for a particluar cycle with and without mismatch are shown in figure 26. It is clear from the figures that the model with transistor mismatches is able to recognize and classify patterns unaffected by the random mismatch. It was further observed that, the location of winners in the piriform patches were also unaffected. The SNR is sligntly degraded in the case of device mismatch errors taken in to consideration.

In the case of ties in the winner take all of the piriform patch (with device mismatch errors taken into account), there is a chance that a different winner might be selected. We will be analyzing the effects due to this and how to rectify it below. This may result in a feedback inhibited current that may not be according to our expectations. One such case out of 40 trials was analyzed and the number of winners in the case of mismatch was found to be 40 and 48 without device mismatch errors. This is due to the fact the there are no ties in the winner take all when simulated with mismatch errors and hence it gives a different value. If this result is obtained by using naive weights then it doesn't affect the performance but whereas the tie results from a trained weight matrix is very remote. Infact in our 40 trials it failed to occur. By the time it becomes a problem the network pattern recognition function

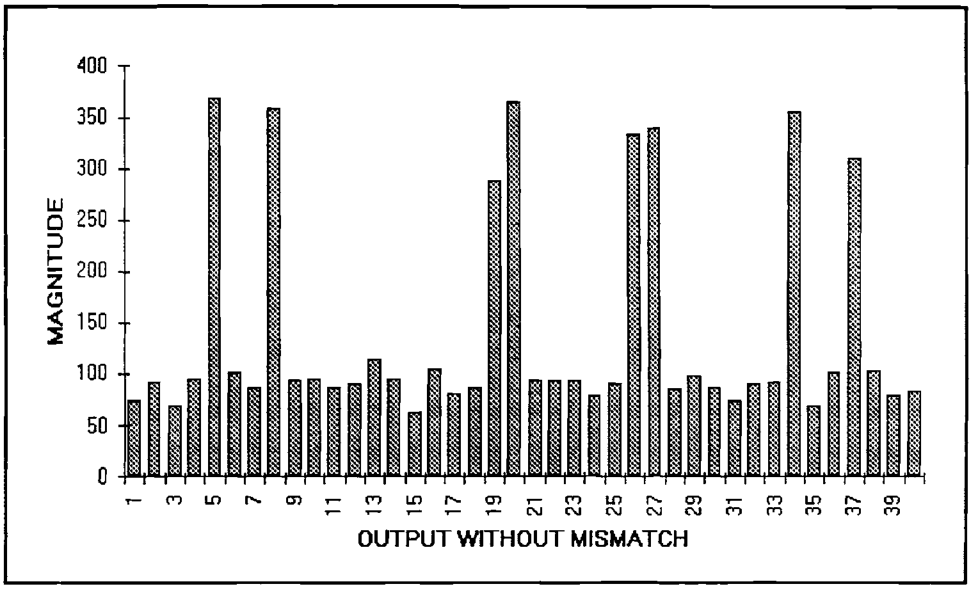


Figure 24. Feedback inhibitory output vectors (without mismatch)

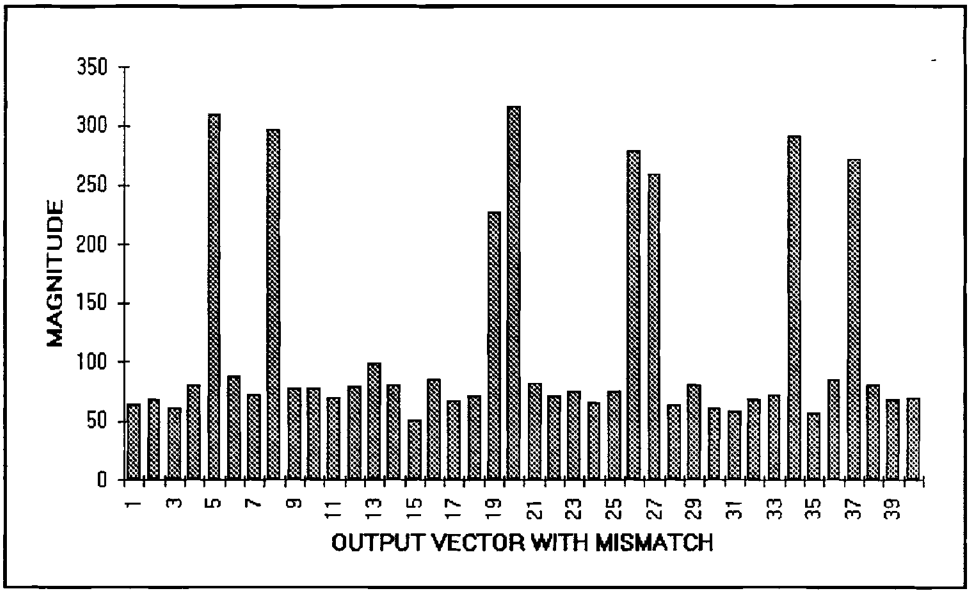


Figure 25. Feedback inhibited output vectors (with mismatch)

SNR (with mismatch)=92.83DB
SNR (without mismatch)= 144.2DB

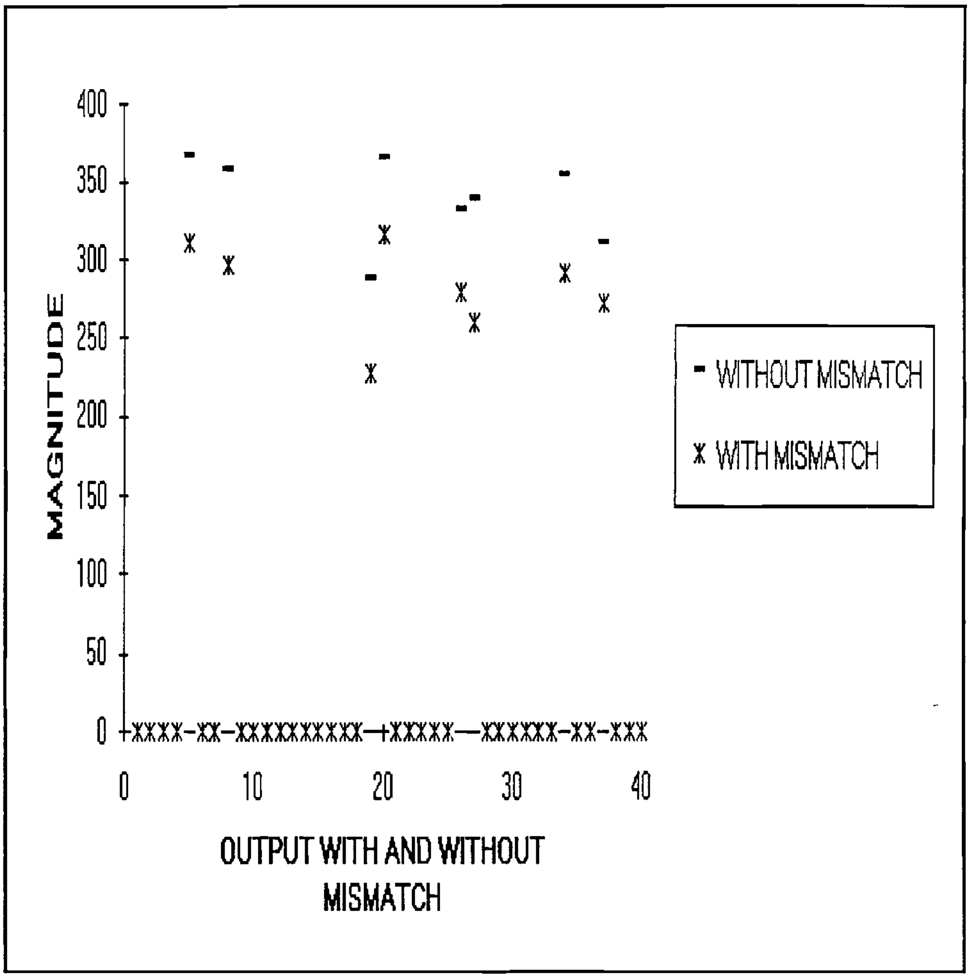


Figure26. Comparison of thresholded feedback inhibitory output vectors with and without mismatch

would have failed.

This problem can be overcome by modifying the model presented in the previous chapter. According to the model the winner is decided by selecting the piriform cell with the highest current. But if we take all the piriform cells within 2-3% of the winner, and select the winner using tie resolver approach, that is by selecting the leftmost (numerically least significant) piriform cell which is within 2-3% of the winner, then this problem is solved. This approach can be carried out for both naive and trained matrices.

Therefore, we can conclude that the transistor mismatch errors have a minimal effect on the model, but still performance remains high. Hifidelity audio is around 72-96DB and the SNR from the simulation results with thresholding and mismatch is 90 DB, which implies that the performance is still high.

CHAPTER IV

CONCLUSIONS AND FUTURE PROSPECTS

Modeling and fabrication of olfactory is a difficult task since olfaction theories are still in the developmental stages. On the one hand, a computer simulation of a too detailed anatomical olfactory model may result in huge volume of data which is difficult to analyze, while on the other hand, too much abstraction and simplification of the anatomical olfactory may loose its relevance to biology with the potential loss of computational power for the anatomical model. Thus the effort towards the moderate level of abstraction is necessary. The correct choice of model detail helps to understand the model while preserving the essential features of the model.

The modified GLA model described in chapter I and chapter II is most definitely biologically inspired, while the basic idea in the minds of the original investigators initially may not have been its hardware implementation, it is well suited for the hardware implementation of an associative processor. The original GLA model has required additional simplifications for hardware implementation but retains, the essential clustering properties of the olfactory bulb (OB) and paleocortex as verified by the simulation results presented in chapter 3. Computer simulations of the model have demonstrated attractive computational properties, such as hierarchial clustering, the extensibility to unsupervised learning, and the ability to detect weak odor obscured by a strong one [16] and further confirmed in this thesis or identifying the significant component of a complex odor. Future statistical experiments will have to be completed on the model of this thesis to confirm hierarchial clustering, and the ability to detect a weak odor.

The GLA model is a statistical model based on long term synaptic potentiation. Such statistical network models do not require high precision. The normalization was achieved using software algorithms. The simulation of the statistical mode presented in chapter II and validated in chapter III, allows the hardware designer to address the complex issues of ; normalization and scaling of feedback inhibition current, performance optimizing, and will assist in ; the selection of patch dimensions, assists us in understanding the learning process, and finally quantifies the effects of device mismatches on the performance of the system. From the simulation results, we can conclude that

(1) the distribution of the active synapses of a piriform neuron and the distribution of the magnitude of the winning piriform neuron appears to correspond with the model,

(2) the "DC" noise term can be removed by thresholding the feedback at a fixed value of $\mu_{Mn} + \sigma_{Mn}$ (for a particular dimension and parametric values) are obtained statistically and,

(3) As the system learns and the weights reach saturation value the SNR of the feedback inhibition current increases, this emphasizes the fact that ratio of saturation to naive weight should be maximized but cant be so large that they do not allow an untrained exemplar to emerge.

REFERENCES

1. R. L. Shimabukuro, An Analog Memory for Artificial Neural Networks PhD. Dissertation, University of California, San Diego, 1992.

2. A. F. Murray, and A. Smith, " Asynchronous VLSI Neural Networks Using Pulse-Stream Arithmetic," IEEE Journal of Solid-State Circuits, Vol. 23, pp. 688-697, 1988.

3. D. O. Hebb, The Organization of Behavior, Wiley, New York, 1949.

4. P. A. Shoemaker, C. G. Hutchens and, S. B. Patil, " A Hierarchical Clustering Network Based on a Model of Olfactory Processing," Analog Integrated Circuits and Signal Processing, Vol. 2, pp. 297-311, 1992.

5. K. Plattig, "Electrophysiology of Taste and Smell," Institute of Physical Sciences in Medicine, pp. 91-115, Aug 1989.

6. J. Ambros-Ingerson, R. Granger, and G. Lynch, "Simulation of Paleocortex Performs Hierarchical Clustering," Science, Vol. 247, pp. 1344-1348, 1990.

7. J. Ambros-Ingerson, Computational Properties and Behavioral Expression of Cortex-Peripheral Interactions Suggested by a Model of the Olfactory Bulb and Cortex, Ph.D. Dissertation, University of California, Irvine, 1990.

8. G. Lynch, and R. Granger, "Simulation and Analysis of a Simple Cortical Network," Psychology of Learning and Motivation, Vol. 23, pp. 205-241, 1989.

9. P. Treleavan, M. Pacheco, and M. Vellasco, "VLSI Architectures for Neural Networks" IEEE Micro, pp. 8-25, Dec. 1989.

10. R. P. Lippmann, "An Introduction to Computing with Neural Nets," IEEE ASSP Magazine, pp. 4-22, April 1987.

11. Jian-Kang Wu "Neural Networks and Simulation Methods," Marcel Dekker, Inc. New York, 1994.

(4) Simulations were performed taking device mismatch errors into account. From these simulation results we can conclude that the system hardware performance and results will have no significant effect on the electronic olfactory performance.

The Signal to Noise ratio gets degraded, but it is able to classify and recognize patterns, which is shown in figures 24 and 25. The simulation results for a mismatch network and a thresholded exemplar were a very respectable SNR of 92.83.

The following studies, statistical experiments and design investigation must be completed to achieve a optimal performing electronic olfactory:

(1) determining the relationship between memory capacity, I* SNR, exemplar length, mitral and piriform dimensions,

(2) determining the relationship between mitral and piriform dimensionality and a fixed weight array,

(3) more statistical experiments have to be completed to understand why the SNR decreases with increase in the number of cycles,

(4) the upper limit on weight saturation,

(5) the effect of noise on μ_w,

(6) determining the relationship between the number of piriform cells per patch and the percentage of ties and,

(7) redesigning hardware as appropriate.

12. S. M. Gowda, B. L. Sheu, J. Choi, C. Hwang,and J. S. Cable, "Design and Characterization of Analog VLSI Neural Network Modules," IEEE Journal of Solid-State Circuits, Vol. 28, No. 3, pp. 301-313, 1993.

13. S. Patil, "VLSI Implementation of Olfactory Cortex Model, " Master Thesis, Oklahoma State University, 1992.

14. J. Choi, and B. L. Sheu, "A High-Precision VLSI Winner-Take-All Circuit for Self-Organizing Neural Networks," IEEE Journal of Solid State Circuits, Vol. 28, No. 4, pp. 576-583, 1993.

15. K. Wagner, and T. M. slagle, "Optical Competitive Learning with VLSI Liquid-Crystal Winner-Take-All Modulators," Applied Optics, Vol. 32, No. 8, pp. 1408-1435, 1993.

16. C. A. Mead, X. Arreguit, and J. Lazzaro, "Analog VLSI Model of Binaural Hearing," IEEE Transactions on Neural Networks, Vol. 2, pp. 230-236, 1991.

17. S. V. Gala, "System Level Integration of Olfactory Cortex Model, " Master Thesis, Oklahoma state University,1993.

18. C. A. Mead, and Mahowald, "A Silicon Model of Early Visual Processing," Neural Networks, Vol. 1, pp. 91-97, 1988.

19. H. V. Shrumer, P. Corcoran, and J. W. Gardner, "Integrated Arrays of Gas Sensors Using Conducting Polymers with Molecular Sieves," Sensors and Actuators, Vol. B 4, p. 29-33, 1991.

20. H. V. Shrumer, "Basic Limitations for an Electronic Nose," Sensors and Actuators, Vol. B1, pp. 48-53, 1990.

21. I. Lunderstrom, R. Erlandsson, U. Frykman, E. Hedborg, A. Spertz, H. Sundegren, S. Welin, and F. Winquist, "Artificial 'Olfactory' images from a Chemical Sensor using a Light-Pulse Technique," Nature, Vol. 352, pp. 47-50, July 1991.

22. J. Bailey, D. Hammerstrom, J. Mates, and M. Rudnick, "Silicon Association Cortex. In S. F. Zornetzer," J. L. Davis, and C. Lau, editors, An Introduction to Neural and Electronic Networks, Academic Press, August 1989.

23. J. J. Hopfield, "The effectiveness of Analogue 'neural Network' Hardware," Network, Vol. 1, No. 1, pp. 27-40, Jan. 1990.

24. S. P. Eberhardt, R. Tawel, T. X. Brown, T. Daud, and A. P. Thakoor, "Analog VLSI Neural Networks: Implementation Issues and Examples in Optimization and Supervised Learning" IEEE Transactions on Industrial Electronics, Vol. 39, No. 6, pp. 552 -564, 1992.

25. S. Y. Foo, and L. R. Anderson, "Analog Components for the VLSI of Neural Networks, " IEEE Circuits and Systems, pp. 18-26, July 1990.

26. C. Schneider, and H. Card, "CMOS Implementation of Analog Hebbian Synaptic Learning Circuits, "IEEE Transactions on Neural Networks," pp. I437-I442, 1991.

27. C. Toumazou, J. Lidgey, and D. Haigh "Introduction," Ch. 1 in Analogue IC Design: The Current-Mode Approach, C. Toumazou, F. J. Lidgey, and D. G. Haigh, Eds., Peregrinus, London, 1990.

28. J. G. Taylor, "Model of Olfactory Learning," Department of Mathematics," King's College London, pp. 277-280.

29. L. B. Haberly, and J. M. Bower, "Olfactory Cortex: Model Circuit for Study of Associative Memory," TINS, Vol. 12, No. 7, 258-264, 1989.

30. Y. Yao, and W. J. Freeman, Pattern Recognition in Olfactory Systems: Modelling and Simulation,"

31. Z. Li., and J. J. Hopfield, "Modeling the Olfactory Bulb and its Neural Oscillatory Processing," Biological Cybernetics, Vol. 61, pp. 379-392, 1989.

32. Z. Li., "A Model of Olfactory Adaptation and Sensitivity Enhancement in The Olfactory Bulb," Biological Cybernetics, Vol. 62, pp. 349-361, 1990.

33. C. Linster, C. masson, M. Kerszberg, L. Personnaz, and G. Dreyfus,"Computational Diversity in a Formal Model of the Insect Olfactory Macroglomerulus," Neural Computation, Vol 5, pp. 228-241, 1993.

34. G. Lynch, Synapses, Circuits, and the Beginnings of Memory, MIT Press, Cambridge, MA, 1986.

35. P. E. Allen, and D. R. Holberg, CMOS Analog Circuit Design , HRW Inc., 1987.

36. S. B. Patil, and C. G. Hutchens," A Novel Squashing Function for Electronic Implementation of Neural Networks," 5th Oklahoma Symposium of Artificial Intelligence, 1991.

37. K. Bult, and H. Wallinga, "A Class of Analog CMOS Circuits Based on the Squre-Law Characteristics of an MOS Transistor in Saturation," IEEE J. Solid-State Circuits, Vol. SC-22, pp. 357-365, June 1987.

38. R. L .Geiger, P. E. Allen, N. R. Strader, "VLSI Design Techniques For Analog and Digital Circuits," McGraw-Hill, 1990.

39. P. R. Gray, and R. G. Gray, "MOS Operational Amplifier Design- A Tutorial Overview" IEEE Journal of Solid State Circuits., Vol. SC-17, No. 6, pp. 969-981, 1982.

40. A. S. Sedra, and G. W. Roberts, " Current Conveyor Theory and Practice," Ch. 3 in Analogue IC Design: The Current-Mode Approach, C. Toumazou, F. J. Lidgey, and D. G. Haigh, Eds., Peregrinus, London, 1990.

41. D. A. Durfee, and F. S. Shoucair, "Comparison of Floating Gate Neural Network Memory Cells in Standard VLSI CMOS Technology," IEEE Transactions on Neural Networks, Vol. 3, No.3, pp. 347-352, 1992.

42. J. Holt, and T. Baker, "Back Propagation Simulations using Limited Precision Calculations," IJCNN, pp. II 121-126, 1991.

43. J. Lazzaro, S. Ryckebusch, M.A. Mahowald, and C. A. Mead, "Winner-Take-All Networks of O(N) Complexity", California Institute of Technology Technical Report Caltech-CS-TR-21-88, 1989.

44. S. Bibyk, and Mohammed Ismail, "Issues in Analog VLSI and MOS techniques for Neural Computing," Analog VLSI Implementation of Neural Systems, Kluwer Academic Publisher, pp. 103-133, 1989.

APPENDIX -- A

MATLAB -- SOURCE CODE

OLFACTORY.M

```
% This program olfactory.m is used for both feedforward and feedback cycles.
% Intially the sparse weight matrix ( temp1.mat ) and input vectors ( inp1.mat ) are loaded
% by these two files
load temp1
load inp1

% the following command functions as a thermometer coder
the = 0:0.0650:1

% the following line converts the normalized inputs to mitral outputs spatially
for i=1:40
        for j=1:16
        if o(i)>=the(j)
                m(i,j)=1;
                else
                m(i,j)=0;
        end
end
end
m1=m'
m2 = m1(:)
% this above command gives the output of the mital cells
m3=m2';
% the above command is executed so matrix multiplication could be done
pw=m3*r1;
save temp5 pw
% the variabe "pw" is the output of the weight matrix and input to the piriform patches
% the following command helps us to select a winner
% "ca" acts as a counter and that is initialized with a value 1

% the following code diveides the piriform output in batches of 16 ( for each piriform
patch )
ca=1
for i=1:40
        for j=ca:ca+15
        k=j-(i-1)*16;
        is(i,k)=pw(j);
        end
ca=ca+16
end
% this code helps to select a winner among the piriform patch
for i=1:40
        for j=1:16
        if is(i,j)<max(is(i,:))
        is(i,j)=0;
        end
end
end
is4=is';
is5=is4(:);
is6=is5'
is7=spones(is6);
% variable "is7" is the winner take all output
save temp14 is6
is8=pw-is6;
save temp15 is8
```

```
% let u=i*j the no of mitral outputs
% let v=k*l the no of piriform inputs in feed forward mode
% the following steps find the output of the piriform patches
% the synapses have a naive value of 1 and a saturated value of 3.2 and it is increased in
steps of
% of 0.4, in case if the varible values have to be changed the next three lines have to be
edited.
wmax=3.2
wnaive =1
% the following lines are used for learing purposes
% this modelling of the olfactory uses unsupervised learning
dw=0.4
% this lines reserves the space for sparse matrix and hence speeds the process
r2_v=spalloc(640,640,640.*640.*.1);
[x1,y1,r1_v]=find(r1);
% these are nothing but arrays and pointers
for v=(1:length(r1_v))
  x1p=x1(v);y1p=y1(v);
 if m2(x1p)>0 & is7(y1p)>0
   r2_v(x1p,y1p)=min(r1_v(v)+dw,wmax);
 else
   r2_v(x1p,y1p)=r1_v(v);
         end
end
% transpose of the weight matrix after learning is done
r3_v=r2_v'
save temp3 r2_v
% the learned weight matrix is stored in file "temp3.mat"
% the following equations are the feedback current generating equations
```

```
in=is7*r3_v;
iinhib=in';
% the feedback unthreshold inhibiting currents are generated using the learned transpose
weight % matrix
z=1
for y=1:40
fif(y)=0
        for x=z:z+15
                fif(y)=(iinhib(x)+fif(y));
        end
z=z+16
end
% the above lines are done for grouping the feedback inhibited currentd based on the
number of % mitral cells
save temp13 fif
% variable "fif" represents the feedback inhibited current
for i=1:40
if fif(i)>250
t(i)=fif(i);
else
t(i)=0
end
end
t1=fif-t;
% variabe "f " gives the value of muMa
f=spfun('mean',t)
% variabel " f1 " gives value of sigmaMa
f1=spfun('std',t)
% variable " f2 " gives mean of muMn
```

```
f2=spfun('mean',t1)
% variable " f3 " gives value of sigmaMn
f3=spfun('std',t1)
% variable " f4 " gives value of muw
f4=spfun('mean',is6)
% variabel " f5 " gives sigmaw
f5=spfun('std',is6)
```

APPENDIX -- B

FORTRAN -- SOURCE CODE

Written by Dr. Patrick Shoemaker NRaD, San Diego.

OLFACTORY.FOR

```
C*  modified from OLF.FOR on 15-16 APR 92

C*********************************************************************
C*     olfactory algorithm simulation
C*
C*     Dimensioned for architecture/parameters to match
C*     network in Ambros-Ingerson dissertation.  However,
C*     note the following differences in our algorithm:
C*       1. no associational collaterals in piriform.
C*          (total number of synapses/piri.cell may match that
C*          of A-I model, though all come from bulb)
C*       2. inhibitory feedback piri.-to-bulb is graded
C*          (vs. thresholded (0 or max) in original model)
C*
C*          ARRAYS
C*     LABEL = label for input data
C*     EXIN = external inputs to bulb patches (glomeruli)
C*        (NOTE: range 0-1 initially; normalized to 0-NBC)
C*        1st index indicates patch, 2nd ranges over data set
C*     BINH = inhibitory feedback to bulb patch from piriform
C*     BCINH = component of piri. FB. corresp. to ea bulb cell
C*     BIN = total net input to bulb patch (w/o normalization)
C*     BN = normalized bulb patch input
C*     IB = # winning cells in a bulb patch
C*     BULB = vector of bulb states (winners are .TRUE.)
C*     PIN = net input to piriform cells
C*     IP = index of winning cell in a piriform patch
C*     PWIN = indicates piri.winners (.TRUE.) in a sniff cycle
C*  Weights are not double-indexed (since matrix is sparse); rather
C*  they are single indexed.  With respect to their piriform cells
C*  their numbers are in ascending order and in contiguous blocks.
C*     NW = # times + 1 that weight #(index) has been updated
C*     NB = index of bulb cell associated w/ weight #(index)
C*     NPW = index of 1st wt. associated w/ piri.cell #(index)
C*        (final element is total # weights + 1)
C*     W = value of (any) weight after (index-1) updates
C*
C*          SCALARS
C*     NBP = # bulb patches (glomeruli); NBC= # cells per patch
```

```
C*     NBT = total # bulb cells
C*     NPP = # piriform patches; NPC = # cells per patch
C*     NPT = total # piriform cells
C*     PBULB = fraction of bulb cells allowed to be active
C*     TOL = tolerance with which PBULB criterion must be met
C*     C = parameter for normalization of inputs
C*     NSNIFF = # "sniffs" between resets
C*     CINH = scaling constant for piri. inhibition of bulb
C*     LEARN = flag for learning
C*     NFLAG = flag for mode of net input normalization
C*     PINH = flag for inhibition (AHP) of piriform winners
C*     BINHLIM = flag to limit FB inhibition to active LOTs
C**********************************************************************

      DIMENSION LABEL(8)
      DIMENSION EXIN(40,8),BINH(40),BIN(40),BN(40)
      DIMENSION BCINH(400)
      DIMENSION PIN(1000)
      INTEGER*2 IB(40),IP(50)
      LOGICAL*1 BULB(400)
      INTEGER*2 NW(120000),NB(120000)
      DIMENSION NPW(1001)
      DIMENSION W(20)
      CHARACTER*1 FLAG
      CHARACTER*12 SETFL, WEIGHTS, WTSAT, INFL
      CHARACTER*12 OUTFL
      LOGICAL LEARN, PINH, PWIN(1024), BINHLIM

      DATA NW/120000*1/
      DATA IB/40*1/ IP/50*1/
      DATA BULB/400*.FALSE./
      DATA PINH/.FALSE./ BINHLIM/.FALSE./
      DATA ZERO/0.0/ HALF/0.5/ ONE/1.0/
```

```
C********************** SETUP *************************

C*      read setup file name
   WRITE (*,'(/" INPUT NAME OF SETUP FILE:")')
   READ (*,200) SETFL
200  FORMAT (A12)
   OPEN (10,FILE=SETFL)
   READ (10,*) NBP, NBC, NPP, NPC
   READ (10,*) PBULB, TOL, NSNIFF, CINH
   READ (10,200) WEIGHTS
   READ (10,200) WTSAT
   READ (10,200) INFL
   CLOSE (10)

C*      Ambros-Ingerson model parameters:
C*      NBP=40, NBC=10, NPP=50, NPC=20
C*      PBULB=5/40=12.5%, CINH: parameter not in original model
C*      WEIGHTS: there are 117 synapses (out of 400 LOT lines)
C*        per piriform cell
C*      WTSAT: weights go from .2 to .4 in increments of .04
C*        (these figs are all double for assoc. synapses but
C*        note many fewer of those fibers are active)

   NBT = NBC*NBP
   NPT = NPC*NPP
   FNBC = FLOAT(NBC)
   FNBP = FLOAT(NBP)
   FNBT = FLOAT(NBT)
   BMAX = PBULB*FNBT + TOL
```

```
      BMIN = PBULB*FNBT - TOL

C*        read interconnection sites
      OPEN (10, FILE=WEIGHTS)
      NPW(1) = 1
      J0 = 0
      DO 90 I=1,NPT
        READ (10,*) IX,NP
        READ (10,*) (NB(J0+J), J=1,NP)
        J0 = J0+NP
90      NPW(I+1) = J0+1
      CLOSE (10)

C*        read weight saturation characteristics
      OPEN (10, FILE=WTSAT)
      READ (10,*) NWMAX
      READ (10,*) (W(I+1), I=0,NWMAX)
      CLOSE (10)

C*        read input data for clustering
      OPEN (10, FILE=INFL)
      READ (10,*) NDATA
      DO 91 ND=1,NDATA
C*        (BN is scratch here)
        READ (10,*) LABEL(ND), (BN(J), J=1,NBP)
        DO 91 J=1,NBP
91        EXIN(J,ND) = FNBC*BN(J)
      CLOSE (10)
```

```
C*     read bulb normalization mode
       WRITE (*,'(/" BULB NORMALIZATION MODE:"/
     *            "  1 = LO & HIGHEST SAT (AGC + MAX)"/
     *            "  2 = BIOLOGICAL (AGC + SIGMOID)")')
       READ (*,*) NFLAG

       IF (NFLAG.EQ.1) THEN
        C = FNBT*(PBULB-ONE)
       ELSE
        C = FNBT*PBULB
        WRITE (*,'(/" INPUT NONLINEARITY CONSTANT:")')
        READ (*,*) CG
       ENDIF

C*     read piriform inhibition flag
       WRITE (*,'(/" INCLUDE PIRIFORM INHIBITION (AHP)?  Y OR N:")')
       READ (*,201) FLAG
 201   FORMAT (A1)
       IF ( (FLAG.EQ.'Y') .OR. (FLAG.EQ.'y') ) PINH = .TRUE.

C*     set bulb inhibition mode
       BINHLIM = .TRUE.
       WRITE (*,'(/" OUTPUT DATAFILE:")')
C*     read output datafile name
       READ (*,200) OUTFL
       OPEN (10,FILE=OUTFL)
C*     learn or evaluate mode
       WRITE (*,'(/" LEARN OR EVALUATE? L OR E:")')
       READ (*,201) FLAG
```

```
98  IF ( (FLAG.EQ.'L') .OR. (FLAG.EQ.'l') ) THEN
      LEARN = .TRUE.
C*       input # learning cycles
      WRITE (*,'(/" HOW MANY LEARNING CYCLES?")')
      READ (*,*) NLEARN
      NL = 0
    ELSE
      LEARN = .FALSE.
    ENDIF
C*       write headers
    WRITE (10,'("       SNIFF    LABEL"/
    *          "             IPI"/)')
    WRITE (*,'("        SNIFF    LABEL"/
    *          "             IPI"/)')
C********************** NETWORK FUNCTION *********************
C*       loop on data
99  DO 150 ND = 1,NDATA
C*       reset piriform inhibition of bulb
    DO 152 I=1,NBT
152   BCINH(I) = ZERO
    DO 153 I=1,NBP
153    BINH(I) = ZERO
C*       reset piriform self-inhibition (if feature enabled)
    IF (PINH) THEN
      DO 155 I=1,NPT
155     PWIN(I) = .FALSE.
    ENDIF
C*       loop on "sniffs"
    DO 151 NS=1,NSNIFF
```

```
C************** normalize inputs & set bulb states ******************
      XMAX = -1.0E3
      BTOT = ZERO
C*        loop on bulb patches to:
      DO 100 I=1,NBP
       NC0 = (I-1)*NBC
C*        reset all bulb neurons;
       DO 101 J=1,IB(I)
101       BULB(J+NC0) = .FALSE.
C*        compute current inputs;
C*        (net input = external input - patch inhibition)
       X = EXIN(I,ND) - BINH(I)
C*        find largest net input
       IF (X.GT.XMAX) XMAX = X
C*        find sum of inputs.
       BTOT = BTOT + X
100   BIN(I) = X


C*        normalization of inputs
C*        (iterative due to nonlinearity)
      IF (NFLAG.NE.1) GO TO 170
C*        mult + add normalization (w/ saturation high)
C*        shift net inputs by XMAX
      DO 184 I=1,NBP
184   BN(I) = BIN(I)-XMAX
C*        shift BTOT too
      BTOT = BTOT - FNBP*XMAX
C*        iterative multiplicative normalization
181  A = C/BTOT
```

```
      BTOT = ZERO
      DO 182 I=1,NBP
       BNI = AMAX1(A*BN(I),-FNBC)
       BTOT = BTOT+BNI
182    BN(I) = BNI
      IF (BTOT+FNBT.GT.BMAX) GO TO 181
C*        set integer output values (# active cells/patch)
      DO 183 I=1,NBP
183    IB(I) = INT(BN(I)+FNBC+HALF)
      GO TO 102
C*        biological model normalization
170   A = ONE
171   A = (HALF + HALF*C/BTOT) * A
      BTOT = ZERO
      DO 172 I=1,NBP
       BNI = G(A*BIN(I),CG,FNBC)
       BTOT = BTOT+BNI
172    BN(I) = BNI
      IF ( BTOT.GT.BMAX .OR. BTOT.LT.BMIN ) GO TO 171
C*        set integer output values (# active cells/patch)
      DO 173 I=1,NBP
173    IB(I) = INT(BN(I)+HALF)
C*        set winning bulb cells, BULB array
102   DO 104 I=1,NBP
       NC0 = (I-1)*NBC
       IBI = IB(I)
       IF (IBI.GT.0) THEN
        DO 114 J=1,IBI
114        BULB(J+NC0) = .TRUE.
```

```
      ENDIF
104   CONTINUE
C************** compute piriform inputs & set piriform states **********
C*      reset piriform inputs
     DO 105 I=1,NPT
105     PIN(I) = ZERO
C*      compute new piriform inputs
C*      (ie, W * LOT vector)
     DO 106 I=1,NPT
      PINI = ZERO
      DO 116 J=NPW(I),NPW(I+1)-1
       IF (BULB(NB(J))) PINI = PINI + W(NW(J))
116     CONTINUE
106   PIN(I) = PINI
C*      find and set winner, ea piriform patch
     DO 107 I=1,NPP
      NC0 = (I-1)*NPC
      PMAX = ZERO
C*      if inhibiting winners, exclude previous winners,
C*      pick the cell w/largest input, and set its inhibit flag
      IF (PINH) THEN
       DO 156 J=1,NPC
        NC = J+NC0
        IF (PWIN(NC)) GO TO 156
        IF (PIN(NC).GT.PMAX) THEN
         NMAX = NC
         PMAX = PIN(NC)
        ENDIF
156      CONTINUE
```

```
      PWIN(NMAX) = .TRUE.
     ELSE
C*       or just pick largest input if not inhibiting
      DO 108 J=1,NPC
       NC = J+NC0
       IF (PIN(NC).GT.PMAX) THEN
        NMAX = NC
        PMAX = PIN(NC)
       ENDIF
108      CONTINUE
     ENDIF
107   IP(I) = NMAX
C*       if done with "sniffs", skip computation of inhibition
    IF (NS.EQ.NSNIFF) GO TO 149
C************** compute FB inhibitions, piriform to bulb ************
C*       compute W(transpose) * piriform vector
    IF (BINHLIM) THEN
C*       limited inhibition - sum for bulb winners only
     DO 157 I=1,NPP
      IPI = IP(I)
      DO 157 J=NPW(IPI),NPW(IPI+1)-1
       N = NB(J)
       IF (BULB(N)) BCINH(N) = BCINH(N) + W(NW(J))
157      CONTINUE
    ELSE
C*       otherwise sum inhibition for every LOT line
     DO 109 I=1,NPP
      IPI = IP(I)
      DO 109 J=NPW(IPI),NPW(IPI+1)-1
```

```
         N = NB(J)
109       BCINH(N) = BCINH(N) + W(NW(J))
      ENDIF
C*        sum inhibitions over ea bulb patch
      DO 110 I=1,NBP
       NC0 = (I-1)*NBC
       BI = ZERO
       DO 111 J=1,NBC
111      BI = BI+BCINH(J+NC0)
110     BINH(I) = CINH*BI
C**************      learn (if flag is set) ****************
149  IF (LEARN) THEN
       DO 112 I=1,NPP
        IPI = IP(I)
        DO 112 J=NPW(IPI),NPW(IPI+1)-1
         IF (BULB(NB(J))) THEN
          NW0 = NW(J)
          IF ( NW0.LE.NWMAX ) NW(J) = NW0+1
         ENDIF
112      CONTINUE
C*        or write results if just evaluating
      ELSE
       WRITE (10,*) NS, LABEL(ND)
       WRITE (*,*) NS, LABEL(ND)
       DO 188 I=1,NPP
        IPI = IP(I)
        WRITE (10,'(13X,I4)') IPI
188      WRITE (* ,'(13X,I4)') IPI
      ENDIF
```

```
C*      close loop on "sniffs"
151  CONTINUE

   IF (LEARN) GO TO 150
C*      write spacer
   WRITE (10,'(/)')
   WRITE (*,'(/)')
C*      close loop on data
150  CONTINUE
   IF (LEARN) THEN
C*      loop on learning cycles
     NL = NL+1
C*      if learning complete, do an evaluation run
     IF (NL.GE.NLEARN) LEARN = .FALSE.
     GO TO 99
   ELSE
C*      if at end of learning run, option of continuing
     IF ( (FLAG.EQ.'L') .OR. (FLAG.EQ.'l') ) THEN
       WRITE (*,'(/" MORE LEARNING?  Y OR N:")')
       READ (*,201) FLAG
       IF ( (FLAG.EQ.'Y') .OR. (FLAG.EQ.'y') ) THEN
         WRITE (10,'(/)')
         FLAG = 'L'
         GO TO 98
       ENDIF
     ENDIF
   ENDIF
   CLOSE (10)
   STOP
```

```
END
FUNCTION G(Y,CG,FNBC)
X = Y/FNBC - 1.0
IF ( X.LT.0 ) THEN
 G = FNBC * EXP(-CG*X*X)
ELSE
 G = FNBC
ENDIF
RETURN
END
```

APPENDIX -- C
SIMULATION RESULTS

mean ma	std ma	mean mn	std mn	mean w	std w	snr
321.60	21.23	86.89	11.24	21.98	2.06	12.73
320.40	31.00	84.21	14.95	22.11	2.09	12.40
339.20	22.30	89.86	10.70	22.00	1.75	13.24
353.60	50.36	94.97	12.26	21.75	2.64	10.68
388.80	23.89	105.75	11.05	21.26	2.41	12.73
371.20	40.66	100.77	10.54	21.41	2.26	11.56
333.60	22.08	86.99	12.78	21.98	1.93	13.54
342.80	34.08	95.63	11.06	21.63	2.62	11.27
333.20	26.90	84.41	8.88	22.26	2.12	14.00
348.80	39.67	91.28	10.85	21.62	2.51	12.14
337.60	27.63	85.50	12.83	21.75	2.23	13.83
321.60	21.26	86.89	11.24	21.98	2.02	12.72
340.40	35.51	89.13	10.65	21.76	2.22	12.44
311.60	38.13	83.61	10.40	22.09	2.88	11.35
353.60	29.22	93.33	10.56	22.02	2.71	12.92
356.40	26.22	97.31	12.94	21.64	2.20	12.30
393.60	31.54	106.74	10.54	21.81	2.52	12.39
339.60	26.13	92.35	11.43	21.84	2.25	12.34
336.80	25.47	88.33	12.21	21.77	2.19	13.19
326.00	30.62	82.25	9.62	22.07	2.26	13.63
376.40	45.96	96.11	14.12	21.94	2.35	12.27
355.60	30.52	91.51	12.48	22.02	2.30	13.36
356.40	26.61	95.76	9.66	21.63	2.16	12.74
311.60	38.13	83.61	10.40	22.09	2.89	11.35
416.40	30.62	114.23	11.55	21.18	2.65	12.28
344.00	22.63	91.43	11.58	21.28	2.25	13.14
343.20	22.42	94.45	8.54	21.82	2.00	12.40
362.80	44.73	100.21	9.72	21.02	2.38	10.84
334.00	45.89	95.83	9.77	21.17	2.52	9.80
344.40	38.24	92.63	12.51	21.44	2.82	11.63

VITA

Meyyappan Ramanathan

Candidate for the degree of

Master of Science

Thesis: STATISTICAL MODELING OF AN ELECTRONIC OLFACTORY

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Devakottai, India, On February 29, 1968, son of Ramanathan Chettiar and Parvathi Achi.

Educational: Graduated from Sidda Naidu Matriculation School, Coimbatore, India, 1983; received Bachelor of Engineering degree in Electrical Engineering from Bharathiar University, Coimbatore, India in May 1989; completed requirements for the Master of Science degree at Oklahoma State University in May, 1995.

Experience: Research Assistant (Jan 1994 to present), Dept. of Electrical and Computer Engineering, OSU; worked at NRaD, San Diego, CA on Master's thesis.

Teaching Assistant (Aug 1994 to present), Dept. of Electrical and Computer Engineering, OSU.

Senior Technical Support Engineer (May 1991 - Jan 1993), Nexus Computers Pvt Limited, Coimbatore, India.

Customer Support Engineer (May 1989 - April 1991), Aurelec Data Processing systems, Pondicherry, India.