UNIVERSITY OF CALIFORNIA

Santa Barbara

# Neural Activity Correlation Processor:
# Reduced Transmission Bandwidth Using Linear Estimation

A Dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in Electrical and Computer Engineering

by

Saeed Mirzaeian

Committee in charge:

Professor Luke Theogarajan, Chair

Professor Forrest Brewer

Professor Upamanyu Madhow

Professor Dmitri Strukov

December 2010

The dissertation of Saeed Mirzaeian is approved.

_____

Luke Theogarajan, Committee Chair

_____

Forrest Brewer

_____

Upamanyu Madhow

_____

Dmitri Strukov

December 2010

Neural Activity Correlation Processor:

Reduced Transmission Bandwidth Using Linear Estimation

Copyright © 2010

by

Saeed Mirzaeian

# ACKNOWLEDGEMENTS

[Begin typing your acknowledgements here. This would be the section in which you dedicate your manuscript to someone or acknowledge the people who helped you.]

VITA OF SAEED MIRZAEIAN
December 2010

EDUCATION

Bachelor of Science in Computer Engineering, Sharif University of Technology, Tehran/Iran, September 2002

Master of Science in Computer Engineering, Sharif University of Technology, Tehran/Iran, September 2004

Master of Science in Electrical and Computer Engineering, University of California, Santa Barbara, September 2008

Doctor of Philosophy in Electrical and Computer Engineering, University of California, Santa Barbara, December 2010 (expected)

PROFESSIONAL EMPLOYMENT

2004-2008: Research Assistant, Department of Electrical and Computer Engineering, University of California, Santa Barbara

Summer 2007: Summer Internship, Calypto Design Systems,

Summer 2008: Summer Internship, Cadence Research Labs,

2008-2010: Research Assistant, Department of Electrical and Computer Engineering, University of California, Santa Barbara

PUBLICATIONS

Saeed Mirzaeian, Feijun Zheng, and K. -T. Tim Cheng, "RTL Error Diagnosis Using a Word-level SAT-Solver," in Proc. of International Test Conference (ITC'08), Santa Clara, CA, Oct. 2009.

HONORS

Rank 1st in UCSB electrical and Computer Engineering screening exam (mathematics and Oral)

Rank 8th in Iran's graduate college entrance exam

Rank 1st among around 1,000,000 students in Iran's university entrance exam

Rank 21st among around 200,000 students in Iran's university entrance exam

FIELDS OF STUDY

Major Field: Low-power and Computational Analog Circuit Design

Studies in Verification with Professor Tim Cheng and Professor Shahin Hesabi

Studies in Test and Diagnostics with Professor Tim Cheng

ABSTRACT

Neural Activity Correlation Processor:

Reduced Transmission Bandwidth Using Linear Estimation

By

Saeed Mirzaeian

The limited power budget, in implantable brain activity recorders, constraints the bandwidth of transmission. Simultaneously, the raw data gathered from brain environment is relatively large. Therefore, an effective information extraction algorithm is required. State-of-the-art implantable neural signal recorders use pattern recognition as the solution. This method has two main phases; in the learning phase each signal is scanned for distinguishable action potentials, and they are clustered based on their properties. Then, a unique signature is assigned to each of these clusters. In the transmission phase, the action potentials detected in the signal are fit into one of these clusters. The transmitter then sends out an event-time packet representing the detected activity. However, the raw signal recorded on the most of the probes is a superposition of many action potentials. This is, mainly, due to the probe design and the density of the brain neurons. In current methods, the signals

from probes with detectable action potentials are used and the signals from remaining probes are discarded.

Signal from a single neuron coupled to more than one probe results in crosstalk noise. This is the main source of redundancy in the recorded analog signals. In this dissertation we develop a low power analog preprocessor which removes this type of redundancy. Minimizing redundancy reduces the transmission bit-rate for commonly used encoding methods. Consequently, this preprocessor opens the road for incorporating many well developed encoding algorithms without sacrificing efficiency. For example, the activity of a single neuron will appear in only one of the channels, resulting in decrease of the number of clusters in the recognition algorithm mentioned earlier.

One of the methods of signal compression for two closely correlated signals is trying to estimate one signal from other and define the error as the new signal. It is proven in information theory that if a signal has a Gaussian cumulative distribution function, the entropy of the signal is proportional to the log of variance of the cumulative distribution function. So, if we decrease the variance of a signal we can transport it with less number of bits. Therefore, if the error has a smaller variance compared to the original signal, it can reduce the bit-rate of transmission. One of the most common estimators is the linear minimum mean square error (MMSE) estimator. A MMSE estimator achieves minimum mean square error among all

estimators of the form “ka - b”. In which ‘a’ and ‘b’ are random signals and ‘k’ is a constant.

In this dissertation we devise a novel, hardware efficient, gradient descent based method of deriving the MMSE estimate for four signals of adjacent probes, and then the estimator is used to reduce the information embedded in the signal to decrease the bandwidth of the ADC and the transmitter in the next level of design. Two prototypes of the design were implemented and tested successfully using discrete analog and digital components. Finally equivalent sub-threshold component were designed and implemented in 0.13um IBM-CMRF process.

TABLE OF CONTENTS

LIST OF FIGURES

# I. Introduction

"The brain is an assembly of neurons that constantly receives information, elaborates and perceives it, and makes decision. (Nicholls, Martin and Wallace)" The building blocks of a brain are called nerve cells (neurons) which undertake these tasks on a large amount of information; an abstract representation of neuron is shown in Figure 1. The information processing in a brain occurs by signaling inside these cells. Due to the importance of their functions, many studies try to understand the meaning behind neurons' signalings. In addition, the interaction between cells should be traced; for example in image perception section of the brain, the signaling of neurons should be traced from the image projected on the subject retina to the perception of the image in cerebral cortex. This task is done by tracing the signal from nerves of the eye to higher layers of brain.

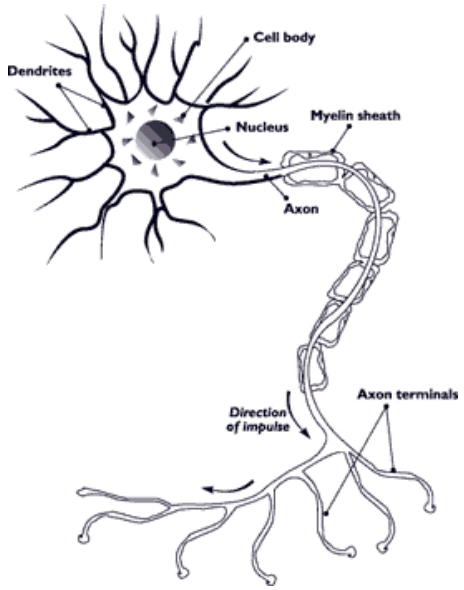



**Figure 1. An abstract representation of a neuron, source: www.sciencecases.org**

# 1. Neurons

The first step in understanding the function of brain starts with identifying of the functions of its building blocks (neurons). Based on the shape, neurons can be categorized in two main categories: stellate (Figure 2-b) and pyramidal cells (Figure 2-a) (Abeles). Stellate cells are neurons with several dendrites radiating from the cell body giving them a star shaped appearance. Pyramidal neuron has a triangular shaped soma[1]. Other key structural features of the pyramidal cell are a single axon, a large apical dendrite, multiple basal dendrites, and the presence of dendritic spines. Pyramidal cells are the primary excitation units of the mammalian prefrontal cortex and the cortico-spinal tract[2]. Stellate cells are found in layer IV. They receive excitatory synaptic fibers from the thalamus and process feed forward excitation to layer IV pyramidal cells. These layers would be discussed more thoroughly in the next section.

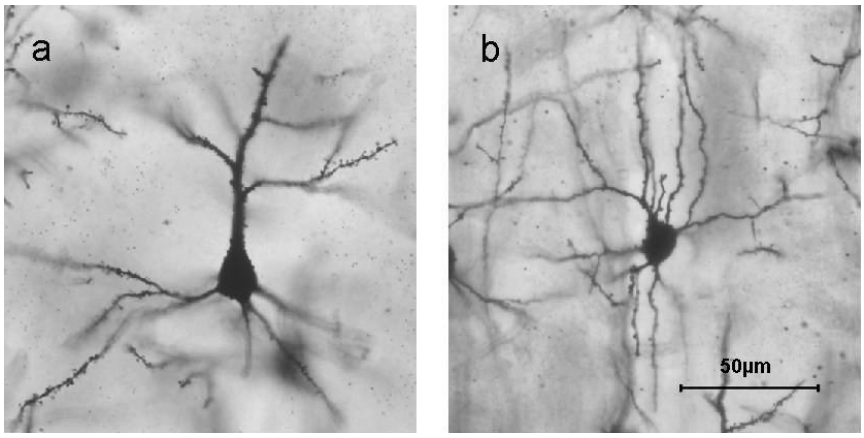



**Figure 2 (a) Pyramidal Cell (b) Stellate Cells source: www.Wikipedia.com**

[1] Cell body

[2] The cortico-spinal tract is a series of axons that cover a path from the cerebral cortex to the spinal cord.

A nerve fiber can be modeled as a tube filled with solution of salts and protein which is separated from an extracellular fluid by an almost impenetrable membrane. Throughout the membrane there are many channels which are permeable to specific type of ion. Figure 3 presents a cross-section of membrane. The ionic strengths of intra- and extra-cellular fluid are similar but they compose of different ions.

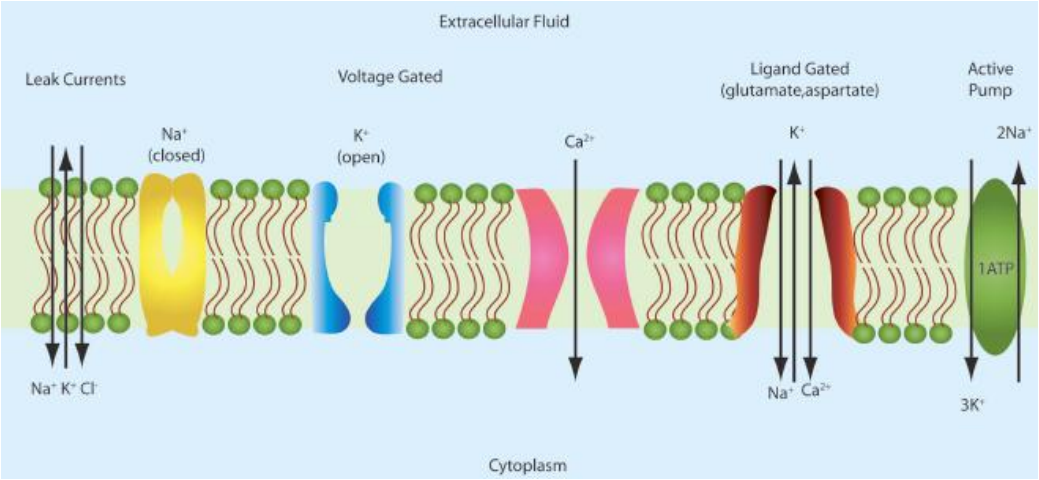



**Figure 3. Schematic of a neural membrane showing the various channels, source: ( Theogarajan)**

in the path to understand the functionality of one neuron studies showed some positive discoveries; such as first signaling of neurons is limited to only two different types; localized and Action Potential. Second, these signal interpretations depends only on the source and destination. Third, they are identical in different animals. Action Potential signals consist of brief signal pulse, about 0.1V (LLano and Gerschenfeld). they period is about 1 ms and move inside the nerve with speed of 120 m/s, and the intensity of stimulus an decoded into the frequency of pulses fired inside the neuron (Adrian). Figure 4 represents the general shape of an Action Potential signal.

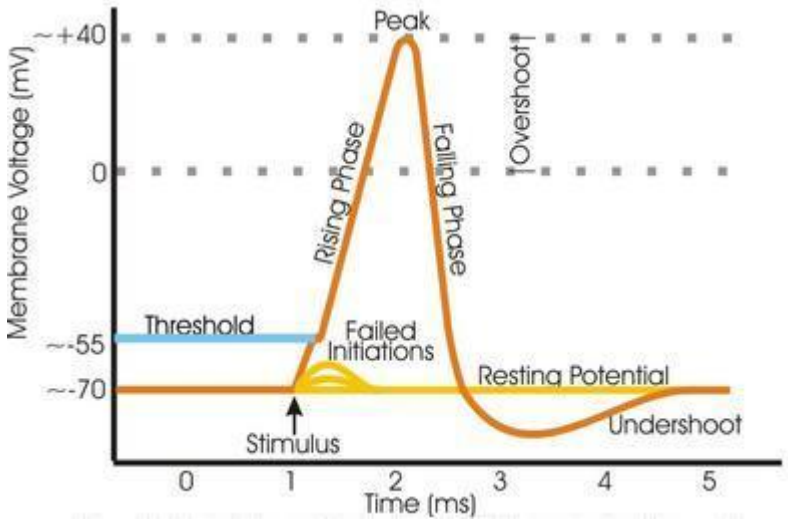



**Figure 4. General Share of Action Potential, source: Wikipedia**

In addition to the shape of neuron, its position, source and destination in neural network gives valuable information on its function. The brain's functions are related to a complex interconnect between neurons rather than a function of a single neuron. Thus, the quality and meaning of a signal depends on origin and destination of nerve; for example, in brain different optic neuron might convey the same pattern of signaling but they are fed into different parts of brain and result in different perception. Based on these assumptions, neurologist made relatively good progress in understanding the beginning stages of sense and final stages of control or execution. Nevertheless, the main concern appears when dealing with the complex scope of perception which happens in higher layers of brain. For example, when an image in presented to the eye the trace of neural activity to brain is mostly discovered, additionally how would a signal for moving the muscles is created and send to the target muscle is well examined but understanding how this picture is perceived deep

inside brain and related to an specific concept is the current challenge (Figure 5). This is similar to the case when we don't know a language and the only information at hand is the symbols used in that language. The only path to understand the complex higher function of the brain is by correlating the activity of individual nerve cells with complex behavior or perceptual activity.

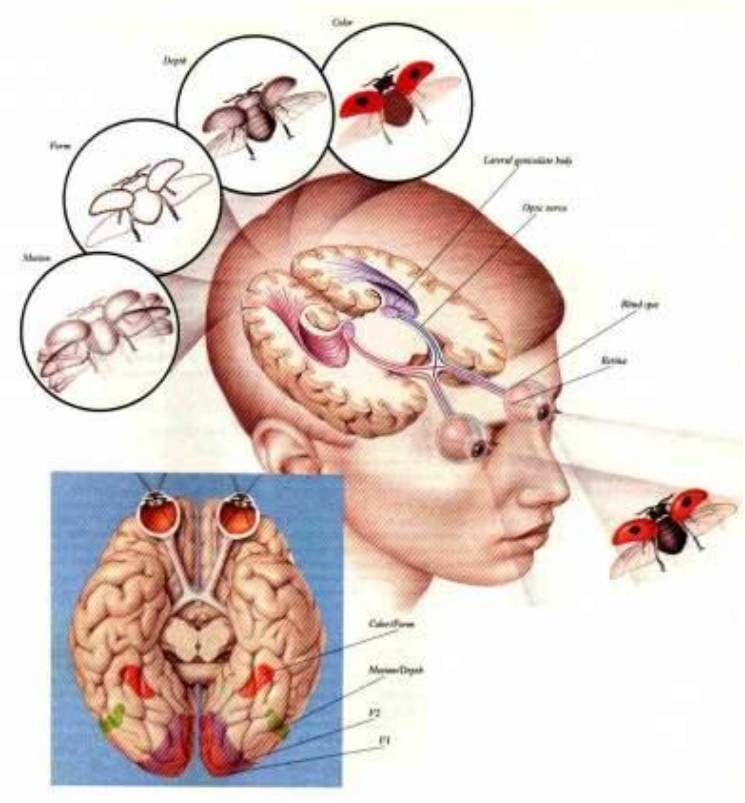


**Figure 5 Trace from Image on retina to final brain perception, source: http://eyemakeart.wordpress.com/**

## 2. Brain Structure and Cerebral Cortex

The brain only deals with symbols of events, which is a code made up of different frequencies. And quality or meaning of each signal depends on source and destination of the nerve fiber, that is, their connection, this was compared to

telegraphic wires which, although, they carry limited number of coded signals it conveys most different concepts by (Helmholtz). Researcher found different sensory modalities are linked to different parts of a brain; and inside each section different type of stimulus deals with different subsection of neurons. Another aspect of brain that helps it capability to do complex analysis such as perception and decision making is the high number of components, an average human brain's cortex consists of $10^{10}$ to $10^{12}$ neurons that uses few stereotyped cells that will be defined in future.

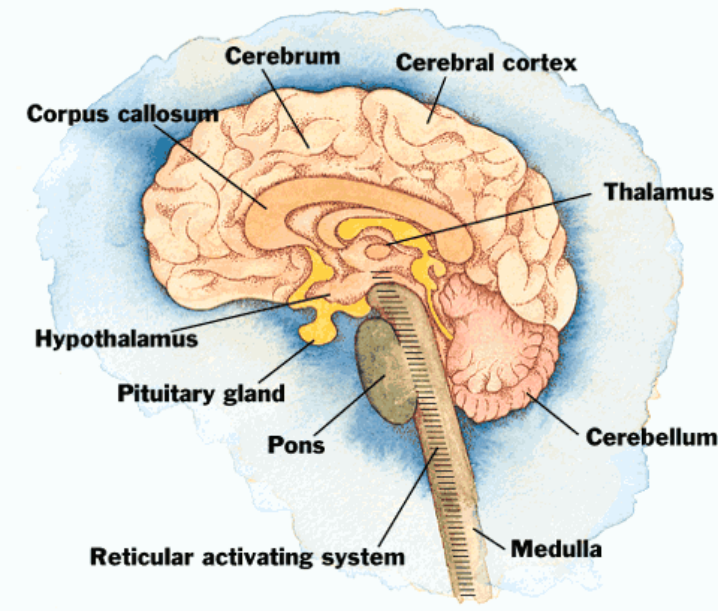



**Figure 6. The diagram for human brain, source: http://www.macalester.edu/psychology**

As it is shown in Figure 6 the cerebrum is the largest part of human brain. The outer most sheet of neural tissue in the cerebrum is the cerebral cortex which is commonly referred to as "grey matter", in contrast the inside of the cerebrum is filled with interconnect axons that are responsible for the interaction both among neurons in

the cerebral cortex and between the neurons in the cerebral cortex with the other parts of the body, this section is commonly known as “white matter”, Figure 7 shows a cross section of cerebral cortex which presents the white and grey matter section of brain. The cerebral cortex consists of two main parts neo-cortex and hippocampus; the neo-cortex consists of six layers of neurons which are connected to each other vertically, this section is limited to more evolved animals including humans. In contrast hippocampus has at most three layers of neurons. Cerebral cortex is considered as the section in the brain responsible for many of human capabilities such as thought, and memory. The input of cerebrum comes from various sensory structures in muscles, skin, and joints; from visual and auditory cortex, and so on.

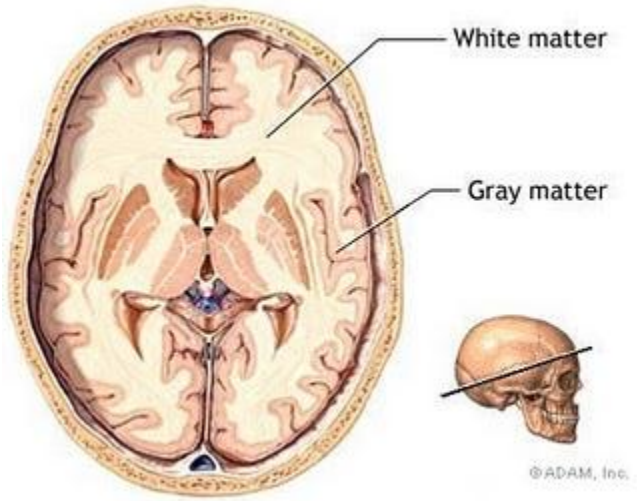



**Figure 7. white and grey matter in crebral cortex, courtesey of New York Times**

Sensory nerves are grouped together, separated from those neurons that respond to other stimuli. This concept of grouping based on function adds new physiological insight to anatomical concepts. For example there is no complete

mapping of retinal surface projecting onto visual cortex, but a series of cell clusters, which perform its special analysis and synthesis of information coming from retina which each section of retina might be transmitted into different parts of visual cortex in different times ( Van Essen).

Commonly, a trained anatomist uses a high resolution MRI scan in order to find the thickness of different layers in brain. However, this process could take days. Alternatively, authors in (Fischl and Dale) used an automated surface recognition algorithm, surface averaging techniques and powerful statistical methods in order to locate the pial surface and the white/grey border, the distance between these two surface results in the thickness of the cortical gray matter. In their study, the average of grey matter thickness in 30 different subjects varied between 2.4±0.3 mm to 2.9±0.3 mm in different sections of the brain. An extreme example is the thickness of the Brodmann area, which exceeds 4mm (Abeles).

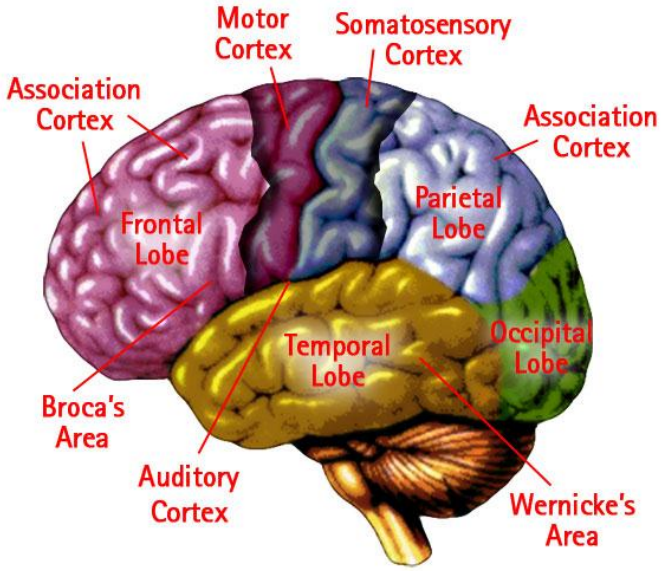



**Figure 8. different section of cerebral cortex, source: www.familyhopecenter.org**

In this thesis we direct our experiments on one of the main sections of cerebral cortex called the primary motor cortex shown in Figure 8, which is responsible for voluntary movement. The motor areas are located in both hemispheres of the cortex. This section has a specific type of neurons called Betz cells which are restricted to advanced primates and humans. These cells are considered responsible for movement of limbs, especially the hand, the large size of Betz cells make them an ideal target for current electrodes. Additionally, intensive research ( Rivara) has been done on the structure and density of primary motor cortex in the human brain. Figure 10 shows a cluster of four Betz cell among other neurons in human cerebral cortex. For these reasons, we will concentrate on this section of the cerebral cortex throughout this paper.

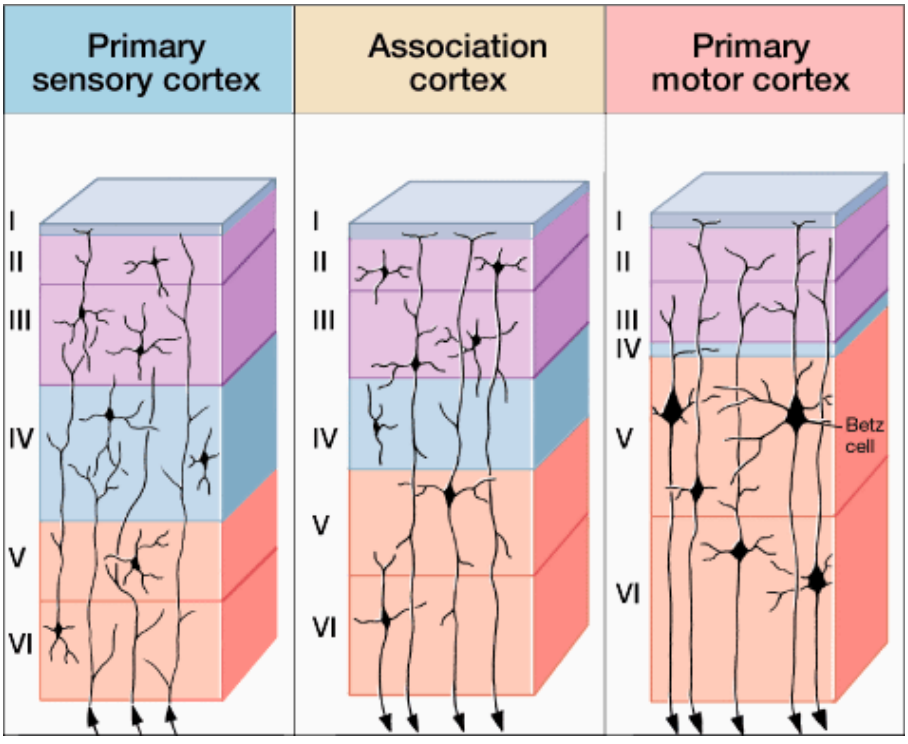



**Figure 9. Layers of Cerebral Cortex in different section of human cortex, source: brainmind.com**

As it shown in Figure 9 inside the cerebral cortex neurons are connected vertically i.e. in a columnar organization. As mentioned earlier, the cerebral cortex constitutes up to six horizontal layers; each layer is distinguished by different thickness and composition of neurons and connectivity among them.

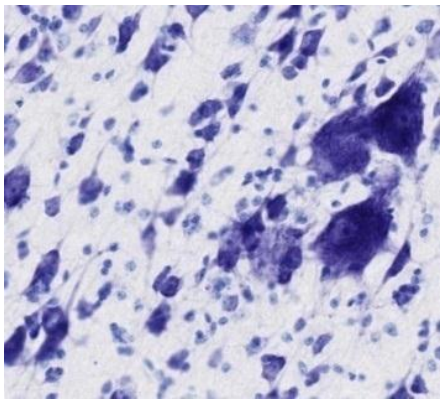


**Figure 10 Huge Betz Cell among other pyramidal cells source: http://brainmaps.org/**

Layer I has very few neurons. It mainly consists of terminal tuft of the apical dendrites of pyramidal cells and mesh of horizontal axons. This layer works as a feedback for associative learning and attention. It has been proven that many of terminal tufts come from thalamus, (Rubio-Gerrido, Perez-de-Manzo and Porrero) so; it is assumed that feedback happens through many parts of cortex through the thalamus.

Layer II of cortex is made of small pyramidal cells. Layer III composed of mostly larger pyramidal cells which are responsible for inter hemispheric cortico-cortical efferent. Layer IV can be divided into two main sub layers; first sub layer is a mixture of small and medium sized pyramidal and stellate cells. Second sub layer is

almost all stellate cells, responsible for intra-hemispherical cortoco-cortical afferent (Jones). Layer V has very large pyramidal cells and many satellite cells. Betz cells are located in this layer of cerebral cortex.

Layer VI has a very small amount of large pyramidal cells, and contains many small pyramidal cells and satellite cells. The responsibility of this layer is to adjust the gain for inputs between cortex and thalamus or in other words between the conscious and the unconscious.

## 3. Action Potential

The signals in neurons are produced to transmit information from one cell to another one. These signal are electrical by nature and consist of a potential produced by an electrical current flows across their surface membranes (Figure 3). The current is carried by ions, such as potassium and sodium, instead of electrons in insulated copper wires, resulting in lower speed of flow and lower concentration of carrier. In addition, the membrane used as an insulator in nerve fiber is $10^7$ times worse than normal wire insolation. Furthermore, the small width of neuron fiber and large size of carriers limits the amount of charge that can flow in a nerve. Consequently, neuron is poor electrical conductors compared to a wire.

There are only two types of signal in the neuron; localized signal and action potentials. The localized potential only spread for short distances and disappear as the result of attenuation by capacitive property of the membrane. These signals appear mainly in two sections of nervous system; first they are generated by a receptor, such

as heat receptor in the skin, as a result of being stimulated. This type of localized signal is called *receptor potentials*. The other location in which the localized potentials are generated is in post-synaptic nerve. *Postsynaptic potentials* are created by receiving excitory or inhibitory signals from pre-synaptic nerve. These two concepts are shown in Figure 11. The main responsibility of localized signals is to enable individual nerves cells to perform their integrative functions and initiate action potentials. Depolarization of the nerve fiber increases the nerve potential up to a point that it passes the defined threshold voltage. In that case nerve generates a pulse which is called an action-potential.

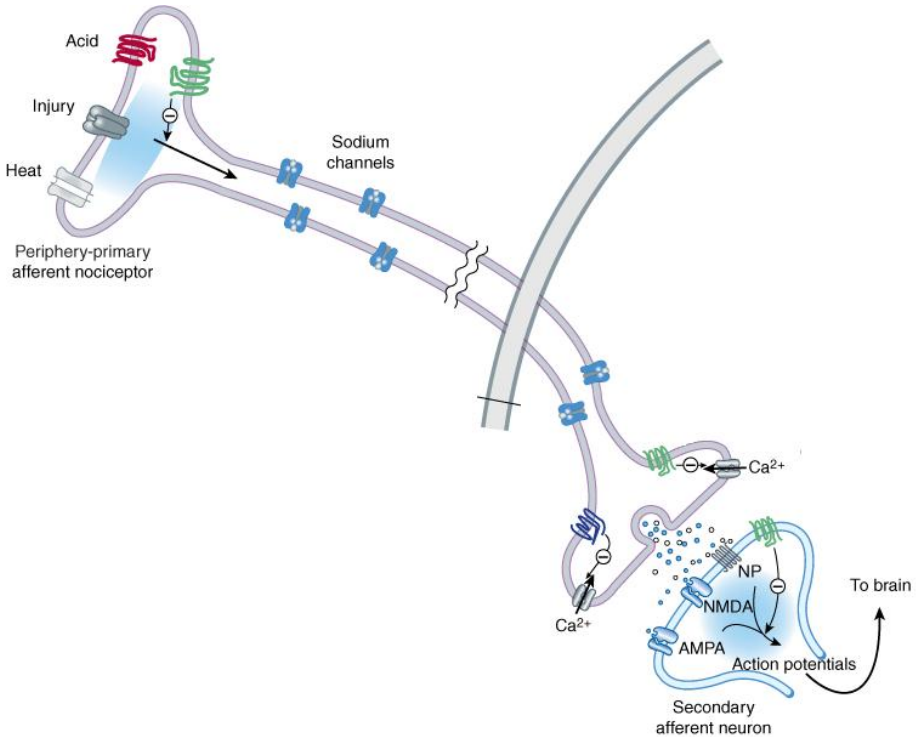



**Figure 11. On top, receptor which generates receptor potentials, bottom a synpse which creates postsynaptic potentials source: [http://www.accessmedicine.com](http://www.accessmedicine.com)**

The action potentials are regenerative impulses which are conducted rapidly over a long distances in the axon of the nerve without attenuation. These potential are similar between different types of neuron independent of their function and their triggering localized signal current or duration. Prolonged depolarization beyond threshold will result in a train of similar action potential in the nerve fiber, which cannot happen together, because a new action potential cannot be triggered unless the neuron passed its *refractory period* after each impulse, shown in Figure 4.

Nerve cells influence each other by excitation that is, they tend to produce impulse in another cell and by inhibition, that is, they tend to prevent impulses from arising in another cell. Each cell receives many excitory and inhibitory inputs from other cells and in turn supplies many other cells. The cell integrates the excitory and inhibitory inputs. If the difference between inside neuron potential and outside environment decreases the cell is depolarized and if it increases the cell is hyperpolarized. If depolarization potential is larger than a threshold voltage the cell would make an impulse. For example in sensory nerves if the stimulus is large enough, the cell will be depolarized and it will impulse. In response to depolarization, the pre-synaptic terminal of the chemical synopsis releases chemical transmitter substance. At the excitory synapse the chemical released by pre-synoptic end depolarizes post-synaptic cell, on the contrary at an inhibitory synapse the transmitter hyperpolarizes the post-synaptic cell.

As an example in a very simple vision process (Figure 5), an image is projected on to the subject retina; the receptors on the ends of visual nerves receive the stimulus which it is sensitive to. In this case light for cone cells in retina shown in Figure 12. Small localized potential are created in the nerve ending and increases the overall depolarization potential of the nerve up to a point of passing the threshold voltage, resulting in a train of action-potential impulse depending on the intensity of the light received from the image in the axon of the nerve. The vision nerve cells which has a synaptic connection to this cell receives the impulse and small localized potentials are created in the post-synaptic nerve, until another action potential is generated, the signal is propagated and analyzed using an series of neurons effecting each other to a point of visual cortex of brain the perception of the image in generated and analyzed.

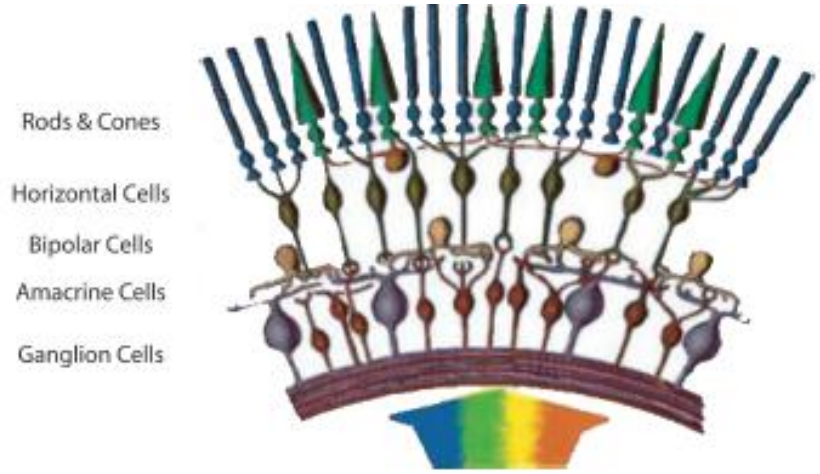



**Figure 12. schematic of human retina source: ( Theogarajan)**

To completely understand how an action potential is created in a neuron, one should understand the ionic fundamentals of neuron in resting and action potential

state. The potential between inside and outside of the neuron depends on concentration of ions, which is set by series of metabolic channels. These channels keep a well-defined potential between inside and outside of the cell by actively switching ions. The main ions included in the electrical property of a neuron are sodium, potassium and chloride. As it is shown in Figure 3 neuron membrane has different channels for different ions which result in different permeability for each of these ions. The differences between channels are because of the different protein gates inside them. These gates regulate the ion flow in and out of the cell. There are three types of gate effecting the action potential generation in a neuron; commonly known as m type and h type which are responsible for flow of the sodium and n type which controls the flow of the potassium.

In the resting phase of the action potential, the m gate is closed, while the h gate is open. Therefore, sodium is neither leaving nor entering the cell. The n gate is also closed, so potassium can neither leave nor enter the cell. The permeability of chloride and potassium is the result of metabolic channels which keep the concentration of ion different in two sides of membrane by using energy.

The concentration of each ion depends on two major constraints; first, the bulk solution of inside and outside of the neuron should be electrically neutral. Second, the total osmotic concentration of ions and molecules inside and outside neuron should be equal. In addition, the permuted ions concentration should follow two extra constraints; concentration gradient and electrical gradient. These two

constraint result in movement of ions in or out of the cell, till they reach an equilibrium state. For example, the potassium ions are more concentrated inside neuron than outside. In equilibrium or rest state of neuron, the potassium ions diffuse to outside while the negative potential of inside neuron results in same amount of ions to drift inside. This can be presented by Nernst equation (Equation 1). Which R is gas constant, T the absolute temperature, z the valance of the ion, and F is faraday. $I_o$ and $I_1$ represent the concentration of the ion outside and inside the membrane.

$$E_k = \left(\frac{RT}{zF}\right) ln(\frac{I_o}{I_i})$$

**Equation 1. Nernst Equation**

Other fully permutable ion in rest state is Chloride, but the difference in this case is that the concentration of chloride ion outside neuron is higher than the inside. On the contrary, membrane is not permutable for sodium ions in rest state of neuron, this prevent the diffusion of this type of neuron from higher concentration (outside neuron) to lower concentration (inside neuron), or drift of ions similarly from the higher potential (outside of the cell) to the lower potential (inside of the cell). One should remember than during action potential firing the membrane becomes permutable to sodium ions. This change of permeability results in the fact that the gradient of potential which during rest state is determined by equilibrium in chloride and potassium ions. In action potential state we would have a rapid change of gradient toward the equilibrium in all three type of ion, for example Nernst equation predicts in the rest state concentration of potassium (30:1) and chloride ions (1:30)

result in potential difference of -85 mV between inside and outside of the neuron. Similarly during action potential phase equilibrium concentration of sodium is +55 mV. Since the permeability of membrane is not zero for sodium ion and infinite for chloride ion and potassium in the rest state, the actual value of membrane potential should be derived through Goldman equation which takes account of the ratio of the permeability (Equation 2), in which $p_i$ is the permeability of membrane for each type of the ions.

$$E_k = \left(\frac{RT}{F}\right) ln(\frac{p_k K_o + p_{Na} Na_o + p_{cl} Cl_o}{p_k K_i + p_{Na} Na_i + p_{cl} Cl_i})$$

**Equation 2. Goldman Equation**

Using Equation 2 and the fact on ratio of permeability in rest state is 1.0:0.03:0.1 (potassium: sodium: chloride), the resting potential between inside and outside of the neuron would be -70mV. In action potential stage the permeability would change to (1:15:0.1) which changes the potential to +44 mV.

When localized potential increase the depolarization of the neuron, the m gates in sodium channels start to open. As a result the permeability of sodium ions increases and more sodium ion diffuse into the cell. At the same time the n gates are still closed. At the voltage around -45 mV this combination would result in a positive feedback which in turn results in a sudden increase of nerve internal potential which forms the raising edge of an action potential. Increase in potential slowly shuts the the

h gates in sodium channels and opens the n gate of potassium channels. Simultaneously, the positive voltage pushes the potassium which in turn pushes the cell potential back to resting equilibrium.

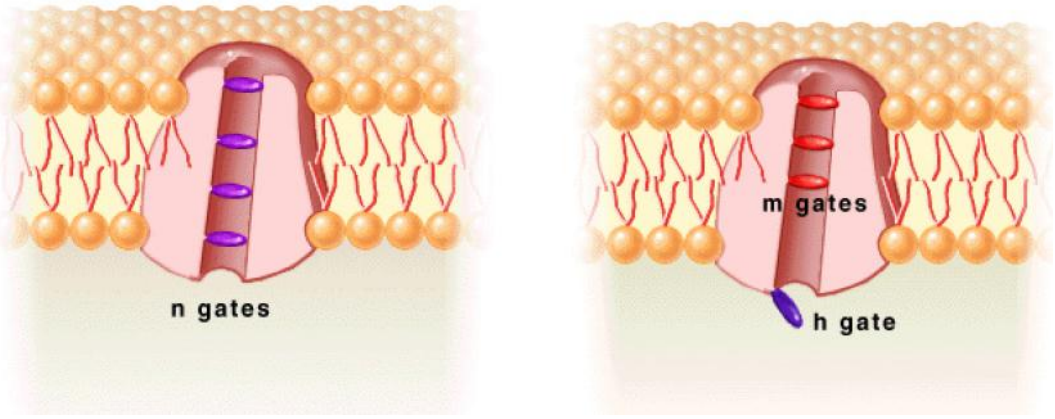



**Figure 13. left; a voltage gated potassium channel in rest state, right Voltages gated Sodium in rest state, Channel source: http://www.lib.mcg.edu**

A small hyperpolarization phase follows the sudden pulse of action potential; this is commonly due to the fact that high potential difference between input and output of the cell opens more potassium channels compared to number of open channels in the rest state, that is, voltage gated channels are open in addition to metabolic channels. These extra open channel do not close once the voltage reaches the resting voltage, giving potassium a higher permeability that resting state. Thus the potential of the inner-cell drops lower than resting state. After each action potential phase, there is a refractory phase in which the cell membrane channel to transit to the resting state, this prevents another action potential to fire. This time period prevent the action potential to travel in both directions inside the axon.

## 4. Intracellular and extracellular recording

The electrical activity of neuron can be monitored by an electrode placed outside the cell membrane (extracellular recording) or by microelectrode that actually penetrates the cell (intracellular recording). Intracellular recording is used to obtain information about the process of excitation or inhibition and the mechanism of how a neuron pulses. In this process the tip of electrode is inserted inside the cell, in this case the voltage difference of input and output of the cell is measured. In some cases intracellular probes is used to excite the cell with a current pulse. Intercellular neural signal recording is possible for a limited period of time, by inserting the probe inside the cell. However, this type of reading would eventually kill the cell. Therefore the main method of recording cell activity would be through extracellular microelectrodes reading. Figure 14 shows the conceptual difference between intracellular and extracellular probes.

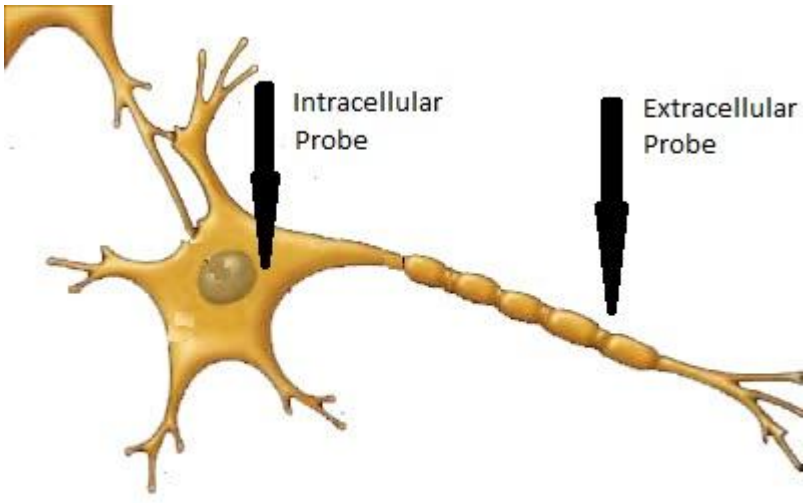



**Figure 14. Different between Extracellular and Intracellular Probes**

For extracellular recording a fine wire electrode is moved close to a nerve cell in the cortex, this technique supplies information about whether the cell is firing or is in quiescent, and records the rate of firing. The voltage produced by ionic current flow around neurons as their cell membranes depolarize (Wise, Sodagar and Yao). The fluid between the cell membrane and electrode surface works as a capacitance which is measured to be between 150pF–1.5nF in contemporary recording electrodes. APs of adjacent neurons appear as spikes in extracellular recordings and usually have durations of 0.3–1.0 ms (Johnston and Wu). The average firing rate of neurons in cerebral cortex is around 10 Hz and can be as high as 100Hz in some cases.

In addition to spikes, extracellular electrode recordings contain a combination of synchronous neural activity of cells further away. This signal, commonly known as local field potentials (LPFs), has a shape of random noise with low frequency (200 Hz) and is smaller in magnitude relative to the spikes. Commonly, the LFP and spike signals are separated in order to be analyzed separately. When a multi-electrode array is implanted in the cortex, some electrodes would detect spikes from one or more distinct neurons, while other electrodes may see no resolvable spikes.

There are many researches trying to deduct intracellular parameters from extracellular spike waveforms (Henze, Borgegyi and Csicsvari). For example it was shown that the width of the intracellular action potential corresponds to distance between distinct points on the extracellular spike. Or the amplitude changes of the intracellular action potential are reflected by changes in the amplitude of the initial

negative phase of the extracellular spike. Figure 15 shows the Diagram of waveform of intra cellular and equivalent observed extra-cellular signal recorded in a single neuron. As it can be seen the corresponding extracellular recording can be estimated as a first derivative of the intracellular recording. Therefore, Extracellular recordings are a good source for monitoring the brain activity.

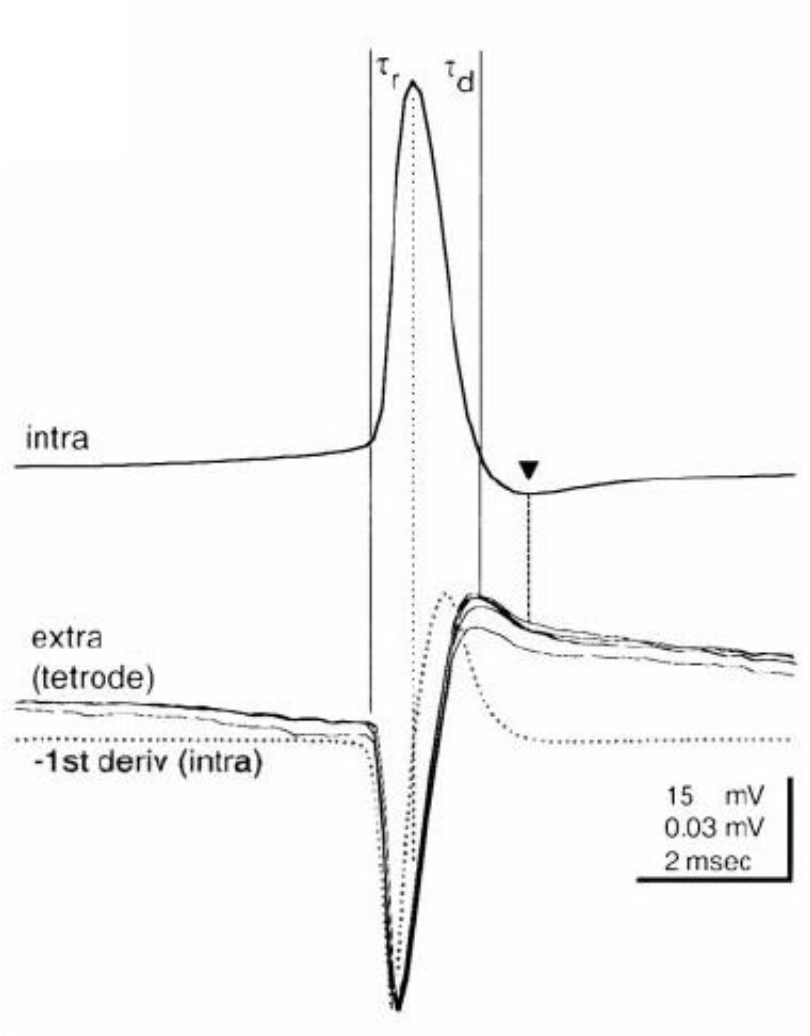



**Figure 15. Diagram of waveform of intra cellular and equivalent observed extra-cellular signal. Dotted line shows the first derivative of intracellular signal, $\tau_r$ and $\tau_d$ show the rising and falling phase, as it can be seen there is an equivalent phase in extracellular recording that can be used to detect each of the phases, source: (Henze, Borgegyi and Csicsvari)**

In both cases many complications arise when the tip is being aligned with the cell, Such as, during the initial alignment or misalignment as a consequence of movement of environment. Most of discoveries for the function of neurons are gathered through removing a thin layer of cells from a cerebral cortex and experimenting in laboratory environment. Throughout this dissertation, This alignment fact motivates us to assume that the alignment is not a given property of the implantable neural recorder and aim for transmitting the neural recording independent of the alignment.

## 5. Structure of the dissertation

The rest of this dissertation is organized as following; in the next section after reviewing the state-of the art probe designs and in vivo signal processing methods, a novel model for layer IV of the brain is introduced, then this effect of inserting a probe inside this model is studied. In the rest of the section the model is used to derive the realistic reading for probes based on the action potential activity of the modeled brain. These results are used to motivate the rest of the thesis toward devising a new method of compression of analog signal which is compatible with the brain environment limitations. In the third section, first we review some basics of information theory and compression techniques. Then, a new algorithm for minimizing the entropy of the signal is introduced. This algorithm is optimized to satisfy the hardware/environment limitations which were discussed in the second section. This algorithm is simulated in SIMULINK and verified. In the next step two prototypes of the design were implemented and tested successfully using discrete

analog and digital components. The Fourth chapter emphasizes on equivalent sub-threshold sub-components that were designed and implemented in 0.13um IBM-CMRF process. These components can be used in the signal processing and compression portion of the under develop low-power implantable neural activity recorder. Final chapter includes a brief conclusion and presents a roadmap for the future work.

# II. Effect of Probes Shape on On-Chip Signal Processing Algorithm

In this section the motivation behind the work is elaborated more comprehensively. This job is done by modeling a subsection of the brain using MATLAB. The MATLAB code is used to generate realistic neural signals. These modeled signals are used to investigate the effects of common in-vivo multi-probe designs. Additionally the effect of different signal-processing algorithms would be compared and contrasted. The primary motor cortex of human brain is used as the baseline for comparing the efficacy of each type of electrode, as a function of the size and shielding. Current methods of on-chip signal processing for each probe option are then discussed. Finally, a roadmap for the direction of the thesis based on these mutually dependent design constraints is drawn.

## 1. Introduction

Advancements in neurology, neural prostheses and cognitive sciences depend on development of new methods of gathering accurate human brain activity. Current methods are limited to distorted electroencephalography (EEG) recordings that are limited in their information content. An example of human EEG recording is shown in Figure 16. Alternatively, in-vivo recoding using chronic implants in animals is used for the purpose of observing brain activity; an electronic circuit is surgically implanted inside the animal's brain and the information gathered is transmitted to the processing computer. This method needs the subject to have an open wound (Mehta,

Ulbert and Schroeder, Intermodel Selective attention in Monkeys I: Distribution and Timing of Effects across Visual Area), thus, due to the fact that a serious infection can result in a death of the animal, this method is limited to sterilized laboratory environments and the animal is put under a restraint. An infection in most cases can result in death of the animal.

**Figure 16. An example of EEG signal, source: http://www.cjd.ed.ac.uk/investigations.htm**

Therefore, these limitations prevent neuroscientists from devising a version of an implantable neural signal recorder suitable for humans shown in Figure 17. Such device can help neurologists to observe their patients, neuroscientists to gather brain activity in normal environments, and even design prosthetic limbs that can be controlled by the patient's brain activity.

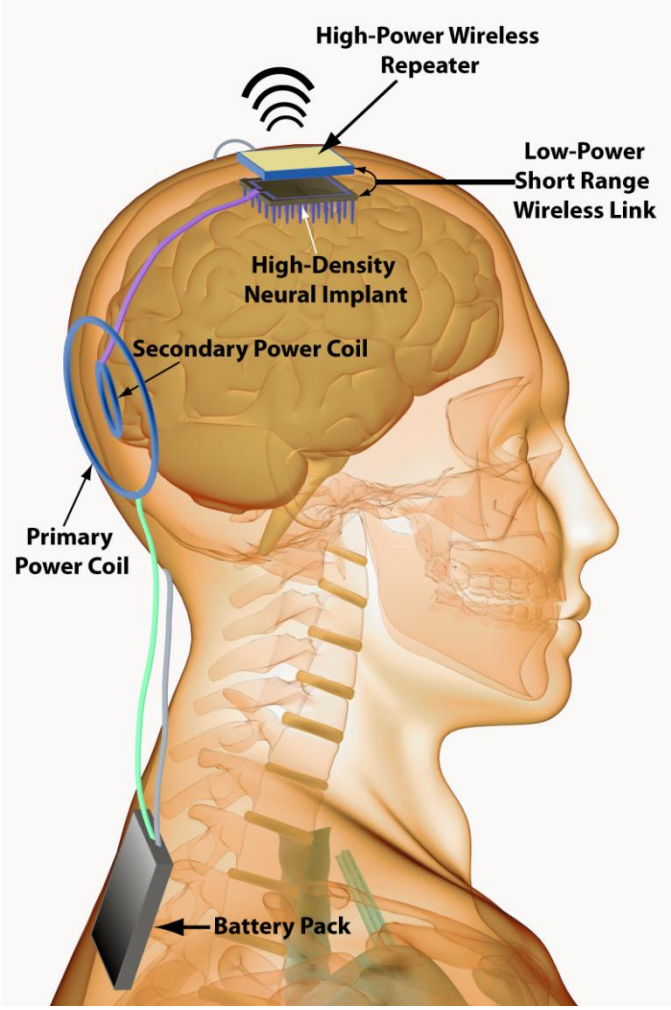



**Figure 17. A diagram of implantable neural activity recorder, source: http://mimetic.ece.ucsb.edu**

Devising an implantable neural signal recorder has many constrains: the high density of neurons and the indirect observation of neural signals due to the neuron insulation (myelin sheath), results in low signal strength and high level of crosstalk noise between recorded signals from adjacent neurons. As it was mentioned earlier any wired connection to the outside world would result in an open wound and a possible infection. Therefore, the desired module should be implanted inside the brain while being powered wirelessly. Another major limitation is that any excessive power

dissipation (> 20mW/cm$^3$) will damage neurons neighboring the implanted site. Furthermore the area of the chip is limited by the cost of fabricating a large area die due to yield limitations. There have been many different groups work on implementing circuits that satisfy some of these constraints (Chae, Liu and Yang) (Rizk, Obeid and Callendar) (Aziz, Abdelhalim and Shulyzki).

Two main aspects of proposed devices are the electrodes that extract the information from surrounding neurons and the signal-processing scheme that is implemented inside the implantable chip in order to reduce power consumption and wireless data-rate.

In the following sections we introduce the common methods currently used to tackle the electrode selection and on-chip signal processing. Then we analyze how different types of electrodes affect the signal fed to the processing unit and thereby dictate the choice of signal processing. Finally we introduce our motivation for the rest of this thesis based on our observations.

## 2. Signal Processing

The first problem associated with recording neural activity is the huge amount of raw data received from electrodes, which results in congestion of the wireless transmitter where the data rate is currently limited to 2.56 Mb/s (Yu, Olsson and Wise) due to the power requirements. Different models are used in order to decrease the amount of information transmitted. For example, one method is the use of time division multiplexing combined with spike detection (Olsson and Wise). Recently,

there has been considerable effort to decrease the bandwidth while keeping the power consumption and area in a limited range. In (Gosselin, Ayoub and Sawan, a mixed signal multichip neural recording interface with bandwidth reduction) authors presented an implant that uses different chips for different sections of the signal processing. These chips are mounted over each other for limiting the area of the whole implant. In this section, we review common methods of signal processing used in implantable recording systems.

### A. Spike detection and sorting algorithms

Spike detection and sorting is an important step in neural signal recording process because it limits the amount of data transferred to the receiver and can distinguish between different neurons that can be detected by one probe. Figure 18 shows the Block Diagram of spike detection and sorting algorithm.

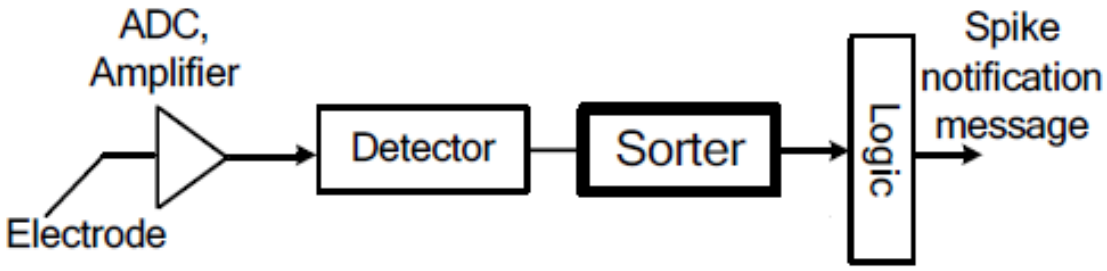



**Figure 18. Block Diagram of spike detection and sorting algorithm**

The first step in spike sorting algorithms is spike detection; the common methods are absolute value (I. Obeid, P.D. Wolf), Teager energy operator (TEO) (Kaiser), and stationary wavelet transform product (Mallat). Wavelet transformation

requires many clock cycles (S. Gibson, J. W. Judy, D. Markovic) and it does not have any apparent advantage over the other two methods of spike detection. In (I. Obeid, P.D. Wolf) the author rejects any complex algorithm in the favor of increasing the signal-to-noise ratio (SNR). The next step in spike sorting is extracting features from the signal that can be used in categorizing the signals. The most common method is Principal Component Analysis (PCA) in which the signal is expressed as a series of coefficients for orthogonal eigenvectors called principal components (PCs). Bayesian matching (Lewicki), template matching and wavelet (R. Q. Quiroga, Z. Nadasdy, Y. Ben-Shaul) are the other feature extraction methods. The major problem with PCA is that recorded signals usually carry a large low frequency noise and distortion, which prevents finding a high quality of PCs. Consequently, PCA will fail in noisy recordings, in these cases human supervision is needed to extract PCs by carefully observing the signals. The other methods rely heavily on arithmetic, which is impossible to implement in the limited budget associated with the power limitation of the implant. Another method relies on noise shaping and entropy of the signal (Z. Yang, Q. Zhao, and W. Liu) to identify the samples containing information these samples are then used for clustering the data, preventing the high power consumption due to including every sample in the clustering method.

The final step in spike sorting is to cluster the spikes based on features extracted in the previous step. Common automated clustering methods such as valley–seek (M. A. L. Nicolelis, D. Dimitrov, J. M. Carmena, R. Crist, G. Lehew, J.

D. Kralik) is used as a guiding point for the clusters, further information on clustering can be found in (J. Han, M. Kamber) .

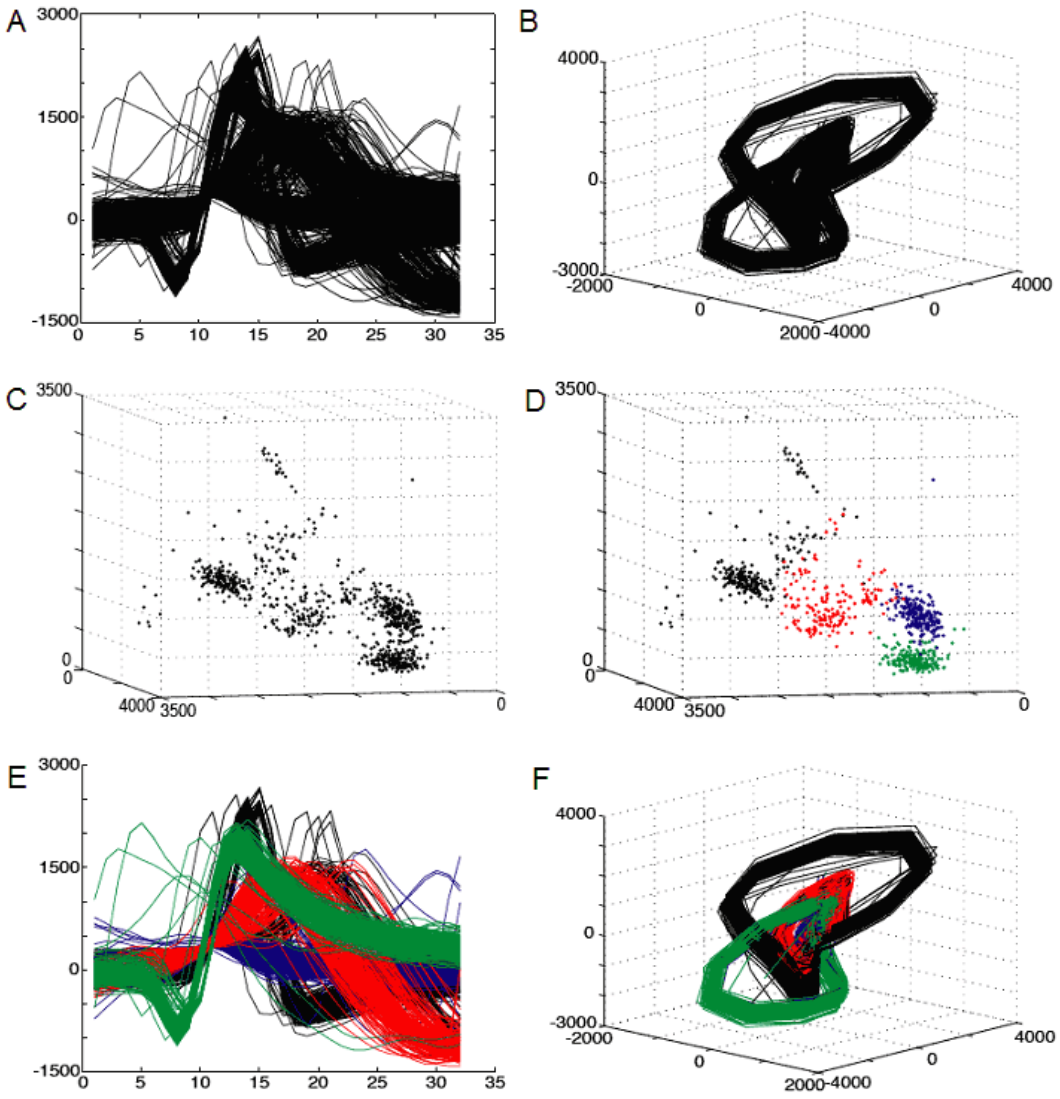



**Figure 19. An example of spike sorting algorithm. A) all detected spikes in the Spike Detection block. B) Diagram of derivative of each spike over 3 PC C) 3-D scatter plot of the 7 sample point in each derivative D) Clustered 7 point sample of the derivative of spikes. E, all detected spikes color-coded according to results of clustering. F, same plot as B but color-coded according to results of clustering, source: http://www.synopsys.com/community/universityprogram/pages/neuralspikesort.aspx**

An example of spike sorting algorithm is shown in Figure 19; in section A of the Figure 19 all detected spikes are presented, in the next section the derivative waveforms of each spike is plotted. A seven point sample of the derivative waveform is shown in section C of Figure 19. Using clustering algorithm the points are clustered into four different clustered that the equivalent spikes are shown in section E which are color coded based on the cluster they were assigned to. The black, green and red cluster are distinguished as spikes of three different neurons and the blue cluster is the noise of the spike detection algorithm used.

Low power analog action potential detection can also be implemented (Gosselin and Sawan, an ultra low power cmos action potential detector). AP isolation was achieved using an energy-based preprocessor that emphasizes the AP shapes. A 9th-order linear-phase delay filter is used to keep a copy of the AP waveforms in their entirety, this method results in better discrimination and improved performance in prosthetic applications. The drawback of the analog delay element is the size of capacitors in VLSI design. Analog computation modules are then used to implement TEO (I. Obeid, P.D. Wolf) energy based spike detection.

## B. Noise Reductions and Compression

However, in many cases the electrode is incapable of distinguishing and clustering spikes and the information received by these probes have the shape of LFPs. To handle these probes different methods were introduced to minimize the

amount of redundancy and increase the relevant information transmitted by these electrodes.

One promising approach that could be implemented using low-power circuits, is a novel sigma-delta analog to digital converter (Chakrabartty, Gore and Oweiss) where the data from other channels are used to remove the crosstalk noise from target signal. This process is done in time domain consequently no transformation is needed for the noise removal. Removing the spatial noise in multichannel recording, using pre-whitening filters and array discrete wavelet transform has also been implemented (Oweiss, Anderson and Papaefthymiou, Optimizing Signal Coding in Neural Interface System-on-a-Chip Modules).

Significant differences in the time constants, current/voltage relations, and repetitive firing behaviors of the different neuron classes' action potential specifications (Mason and Larkman) can be used to distinguish activity of a specific type of neurons from LFPs. The use of this method has not been explored.

## 3. Different Probe Designs

In order to perform neural recording in the cerebral cortex, the main mechanical stepping stone is devising a reliable and small probe array. Small size of the neurons and comparatively large area of the unshielded section in the electrode results in recording of many different neurons on each probe. On the other hand scaling probe size to smaller dimensions will be limited by strength needed to penetrate the cortex. In this section we review the current state of the art electrodes.

## A. Utah array

The Utah Intra-cortical Electrode Array (UIEA) is one of the longest chronic implants (Wise, Sodagar and Yao). This two-dimensional silicon array is as shown in Figure 20. UIEA is developed at the University of Utah is based on glass reflow, sawing, and etching from p-type silicon. UIEA comprises of a 10 X 10 grid of needles projecting 1.5 mm above a 4.2 mm X 4.2 mm X 0.2 mm thick substrate ( Owens, Denison and Versnel). Each electrode is approximately 80 µm wide, and has a sharpened tip. They are insulated with polyimide and the approximately 50 µm at the tip is metallized with platinum to form the active electrode region for recording or stimulation.

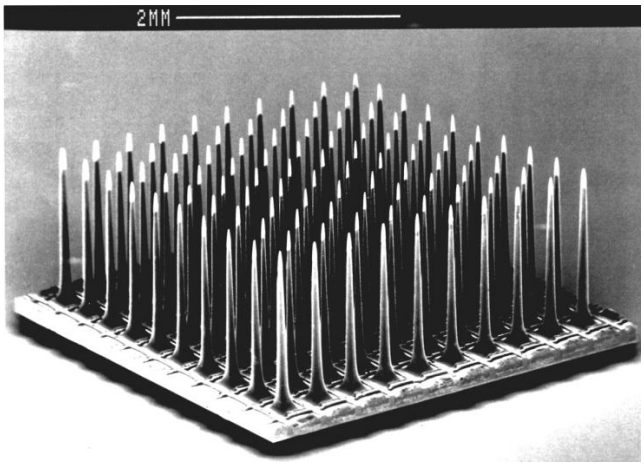



**Figure 20. Utah array, source: http://www.sci.utah.edu/**

Each electrode is capable of reading only one site, thus it is called a 2D array. Because of high density of the electrodes in the UIEA pushing the implant inside cortex will result in damages in cortical tissues, consequently, the array is inserted in

the cortex using a high-speed insertion tool (Nordhausena, Maynarda and Normann). In one experiment ( Owens, Denison and Versnel), they were capable of receiving distinguishable spike recording from an average of 58.6% of the electrodes in the array with an average of 3.4 separable spikes for each of those electrodes.

### B. Michigan array or 3D site reader

Michigan Probe provided a two-dimensional approach to multi-electrode recording by using semiconductor batch processing techniques the. Each probe in this array provides multiple recording sites on a single penetrating pin.

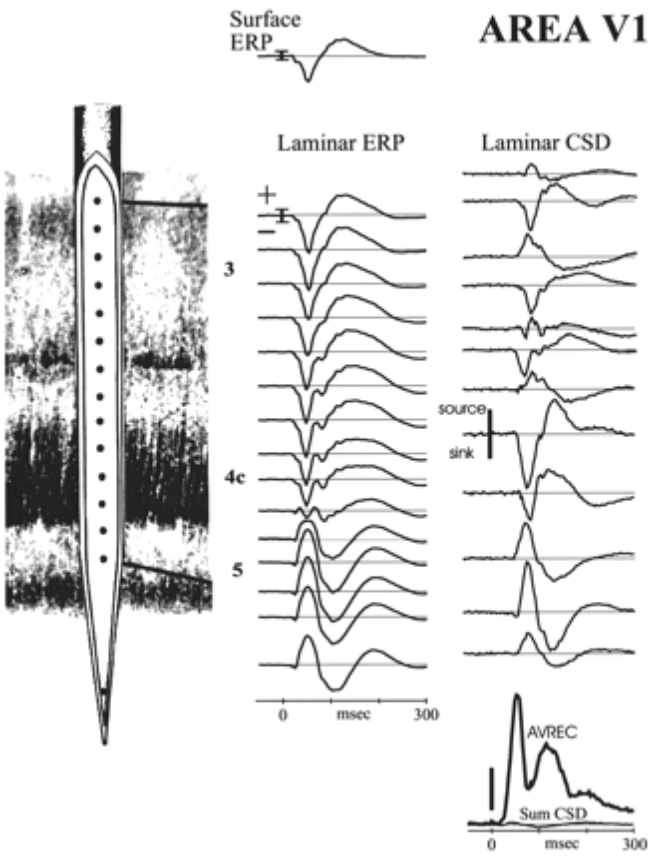



**Figure 21. A 3D probe with 15 reading site and the reading from them source: (Mehta, Ulbert and Schroeder, Intermodal Selective Attention in Monkeys. II: Physiological Mechanism of Modulation)**

Several such probes are assembled to form three-dimensional (3-D) electrode arrays, for example four probes of 16-channel can create a 256-site array (Wise, Sodagar and Yao).

In (Mehta, Ulbert and Schroeder, Intermodel Selective attention in Monkeys I: Distribution and Timing of Effects across Visual Area) and (Mehta, Ulbert and Schroeder, Intermodal Selective Attention in Monkeys. II:Physiological Mechanisim of Modulation) a probe is used which has 14 equally spaced (150um) recording sites and one probe electrode 1 mm below the array, this probe enables the receiving of the reading from 15 different depth consequently different layers of cortex. Figure 21 shows that different layers will result in different action potential shapes.

### C. Flexible array

Using a combination of processes for surface micro-machining and folding (Suzuki, T. ; Mabuchi, K. ; Takeuchi, S.) devised a completely flexible array of electrode. The flexibility will result in more stable and less-invasive method of gathering neural recordings over a long duration in soft and constantly moving neural tissue. The array was fabricated on a 10µm-thick parylene substrate. It consists of six flexible needle-shaped probes (1.2" in height, 160µm in width), each with three recording sites. Figure 22 shows the concept of how the flexible array inserted with a nonflexible soluble shield at the beginning and would change its shape as a result of degradation of the nonflexible shield.

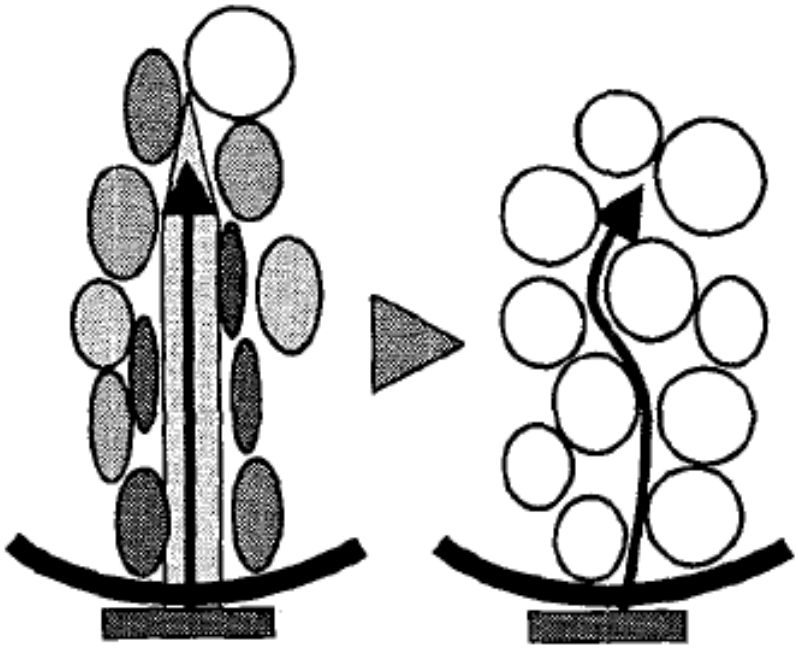


**Figure 22. Concept of flexible array source: (Suzuki, T. ; Mabuchi, K. ; Takeuchi, S.)**

### D. Array of microwires

Nicolelis et al. (M. A. L. Nicolelis, D. Dimitrov, J. M. Carmena, R. Crist, G. Lehew, J. D. Kralik) used an array of 32 S-isonel (or Teflon) coated tungsten (or stainless steel) microwires that were secured by a custom designed jig, the distance between electrodes were between 100um to 300um. The chips were implanted in three rhesus monkeys. These electrodes were blunt to subsequently implant the microarray, slowly at approximately 100um/min, during continuous electrophysiological monitoring. The array was screwed into the skull. The wired connection of the electrodes to outside world lends itself to enormous signal processing power using outside systems. Eight different principal components (PCs) were provided to the user from which 2-3 PCs were used for clustering the spikes.

Since the implanted were mounted on the skull most of the recording and clusters (around 80%) were valid even after 8 days of implementation. The final result for this experiment was 54% of the implanted micro-wires in the three monkeys yielded at least one isolated cell, but since some electrodes could detect more than one neuron, 576 electrodes could distinguish 421 neurons.

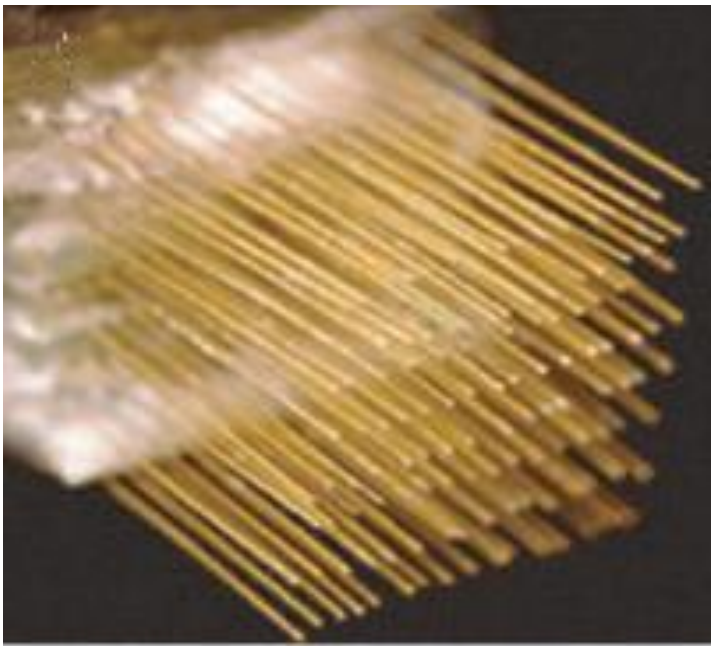


**Figure 23. Picture of array of micro-wires, source: (M. A. L. Nicolelis, D. Dimitrov, J. M. Carmena, R. Crist, G. Lehew, J. D. Kralik)**

## E. Unshielded array and custom-made probes

One significant problem with current electrode arrays is the calibration of the probes in a way that they are aligned with the exact type of neurons or layer of cortex that neurologists need to observe. Current solution includes the insertion of the probe slowly while electrophysiological monitoring, but Utah array is inserted in one move

and this method would not work. Other method is using unshielded arrays which can interact with many layers and neurons at the same time. Nonetheless because of the large area of adjacency of a completely unshielded probe and many different neurons, no group has been motivated to devise an unshielded probe. Figure 24 represent a series of unshielded probe design and implemented in Mimetic Lab of UCSB.

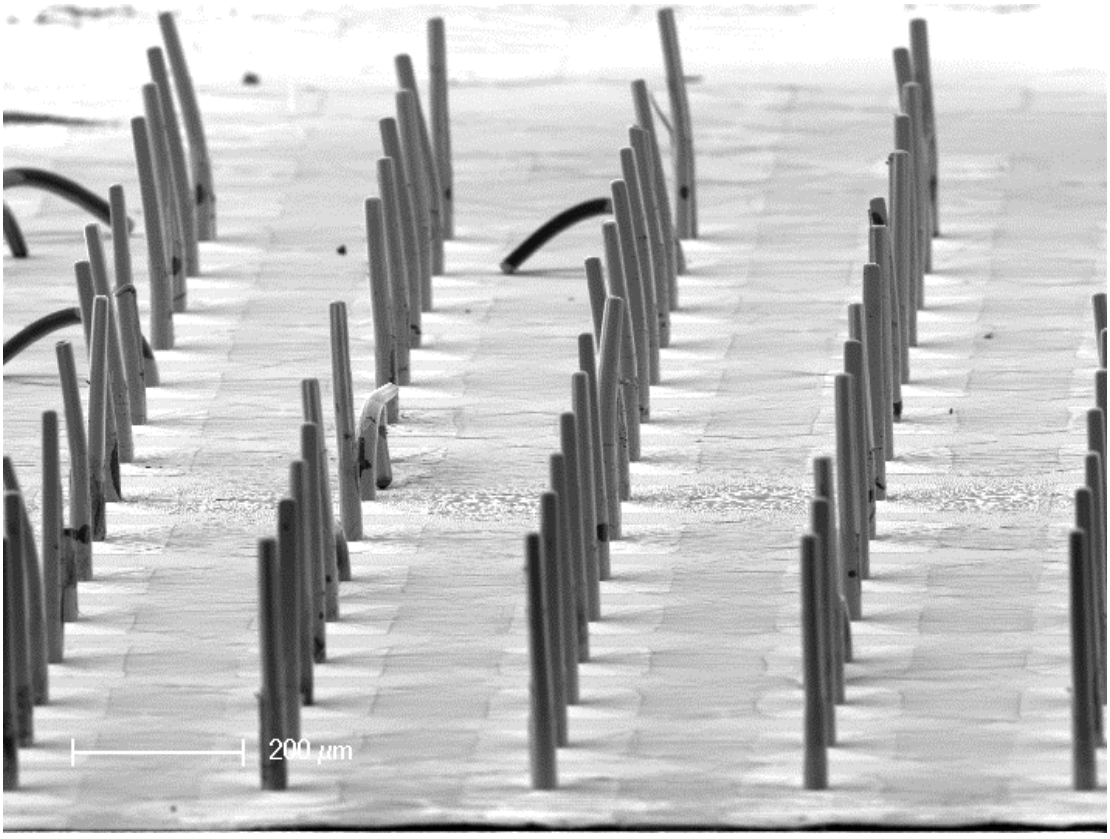



**Figure 24. Image representing the grown unshielded probes in, source: http://mimetic.ece.ucsb.edu**

As discussed earlier the thickness of the cerebral cortex differs drastically among different people. So, devising an unshielded array would be a general solution for the problem of accurate insertion of the shielded probes. Additionally, due to the

fact that the movement of brain in skull the implant might move in place and a fully shielded probe cannot calibrate itself with its new location. We mentioned earlier that different neuron types have different spiking and timing sequences that can be used to distinguish activity of a specific type of neuron in a reading of many different ones. This method of designing probes is still an open-ended question because of the huge amount of signal processing and transmission needed to separate different neural activities. As discussed in the introduction, MRI can be used in mapping and finding the thickness of cortex. Therefore custom-made probes would be a very simple solution for this problem.

## 4. Simulation Using Different Criteria

In order to show how different probes shape and sizes would affect the recorded signal properties, we designed a simulation experiment. As it is shown in Figure 25, since the extracellular voltage resulting from action potential around the soma is almost 4 times bigger than dendritic and axonal voltage (Gold, Henze and Koch), we model the neurons only as a spherical soma in our paper. The test data was generated based on the macaque monkey's cortical action potentials, which were adapted from (I. Obeid, P.D. Wolf). In our simulations each neuron activity is modeled as a memory-less Poisson process with a mean of 14.63 milliseconds as a period between spikes of a single neuron ( Moehlis, Shea-Brown and Rabitz), consequently, the time between each two occurrences of spikes is generated using an exponential distribution.

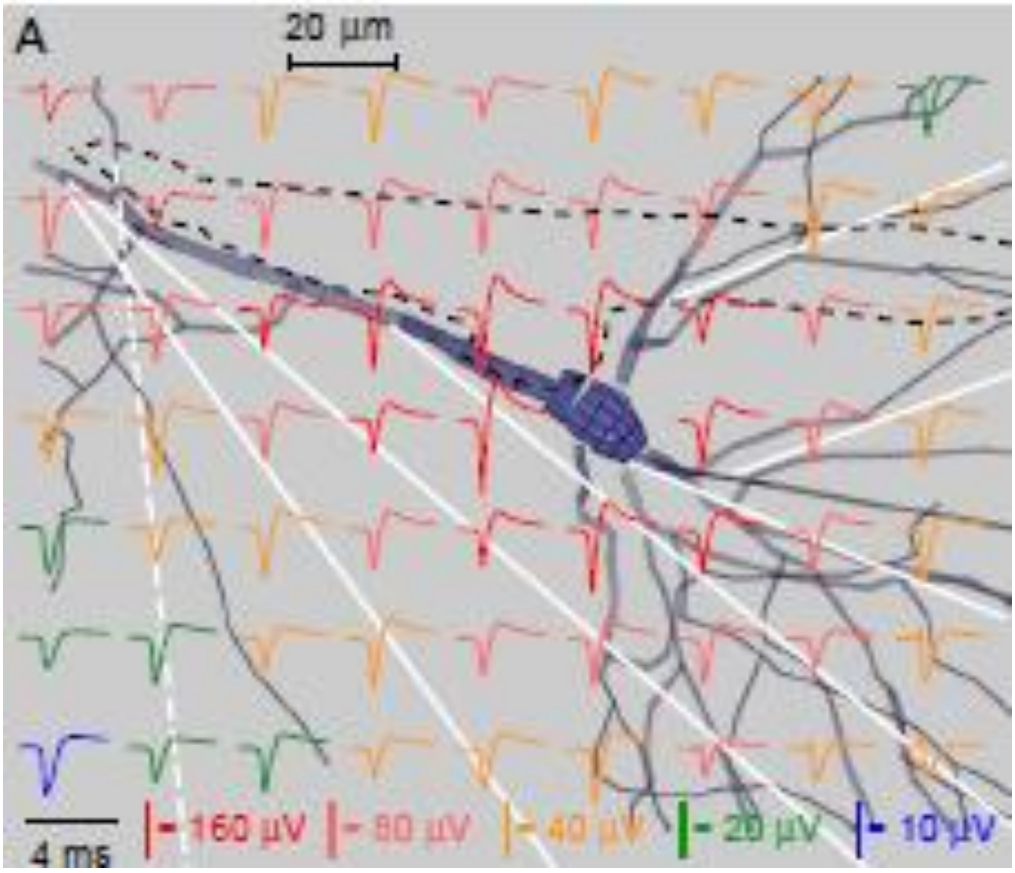



**Figure 25. The difference in magnitude of the extracellular activity recorder by probe depending on the distance of the probe source: (Gold, Henze and Koch)**

Based on the studied conducted in (I. Obeid, P.D. Wolf), the SNR is a more important measure in effectiveness of each spike detection algorithm, rather than the nature of algorithm itself. Thus, we use normal threshold based spike detection algorithm for our experiment and put our effort on improving the SNR.

### A. Effect of cell clustering

In the first step of this experiment we show how the arrangement of neurons would affect signal's shape. As we know the detectable voltage of each neuron on a

specific electrode is inversely related to square of the distance between the two. Therefore, if distances of two cells from the electrode differ drastically, the probe would pick up closer neural spikes while it would regard the farther cell's activity as a small background noise.

We simulated readings of an electrode for different cases in which neurons were clustered in different group sizes. Figure 26 represents the results in different cases. In Figure 26a the neurons are not clustered, so one neuron's spikes are absolutely detectable while the other neural activity is present as small amplitude LFP. In Figure 26b cells are clustered in groups of two, in this case as many as two different amplitudes of spikes can be detected which correspond to two different neurons. As the clustering coefficient increases, shown in Figure 26c and d, the signal shape would change into a cluster of undetectable spikes and a LFP shaped signal emerges in the case when the clustering is 16 cells.

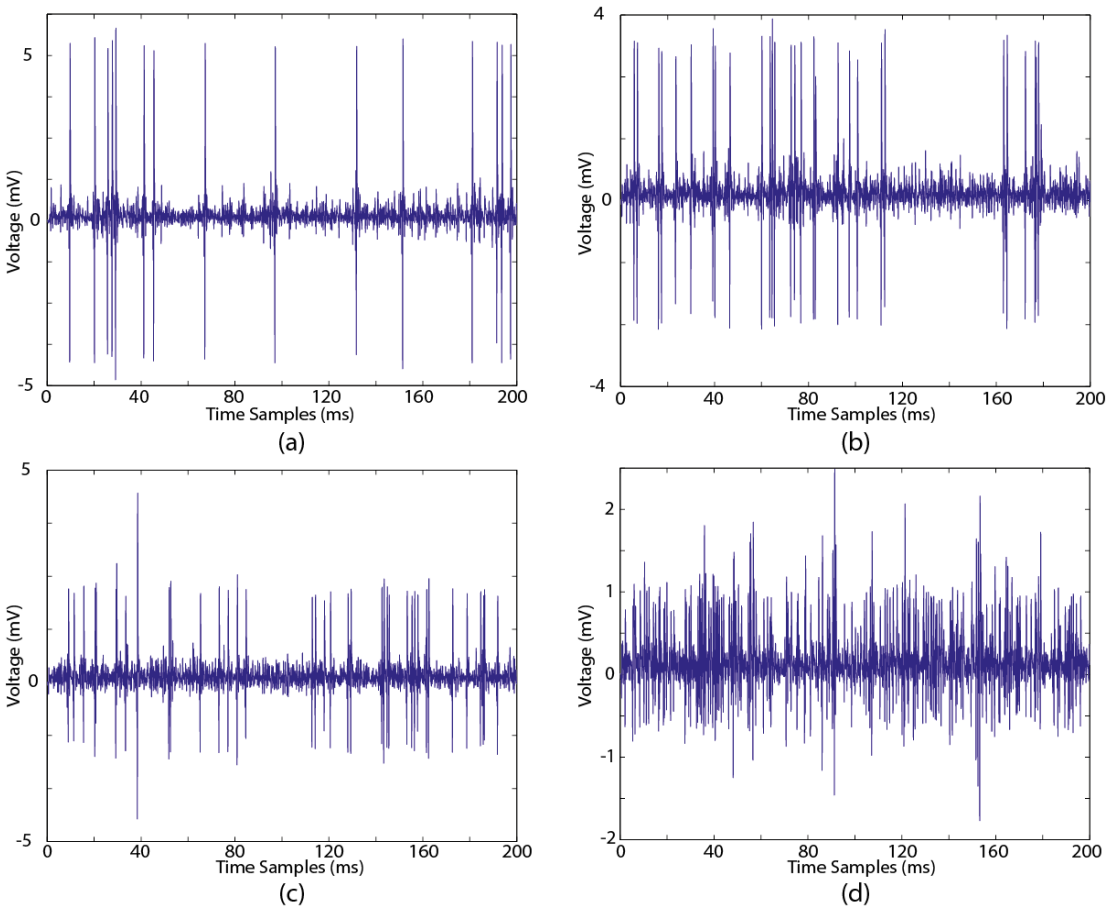



**Figure 26. Modeled neural recording signal for different number of clustering of neurons. (a)Recording when neurons are clustered in groups of one (b) Recording when neurons are clustered in groups of two (c) Recording when neurons are clustered in groups of four (d) Recording when neurons are clustered in groups of sixteen**

## B. Probe adjacency

As it was verified in last section the number of close range neurons and the relative distance of them from recording section of the probe have an important effect on the recorded signal. In order to observe the real cortex structure and arrangement of the neurons, we modeled fifth layer of cerebral cortex and then the effect of inserting a probe in that arrangement was studied.

The volume and density of neurons in motor cortex was derived from ( Rivara). In his thesis the author uses 6 post mortem samples for finding the distribution and density of Betz cell compared to other pyramidal cells in primary motor cortex. The results vary significantly among samples, so in our simulation we used the average and standard deviation of the 6 samples. In his thesis author showed how the density of Betz cell increases in the hand knob section of the motor cortex, which is associated to movement of the hand. Additionally, the author presents a size-based differentiation between Betz cells and other pyramidal cells in fifth layer of motor cortex.

Figure 27 shows the modeled structure of fifth layer of cortex, which consists of the large dark red spheres representing Betz cells around 10%, large pyramidal cells shown as blue spheres 65% and small spiny and smooth stellate cells. On average 34.23% of the cube is filled with neuron cells.

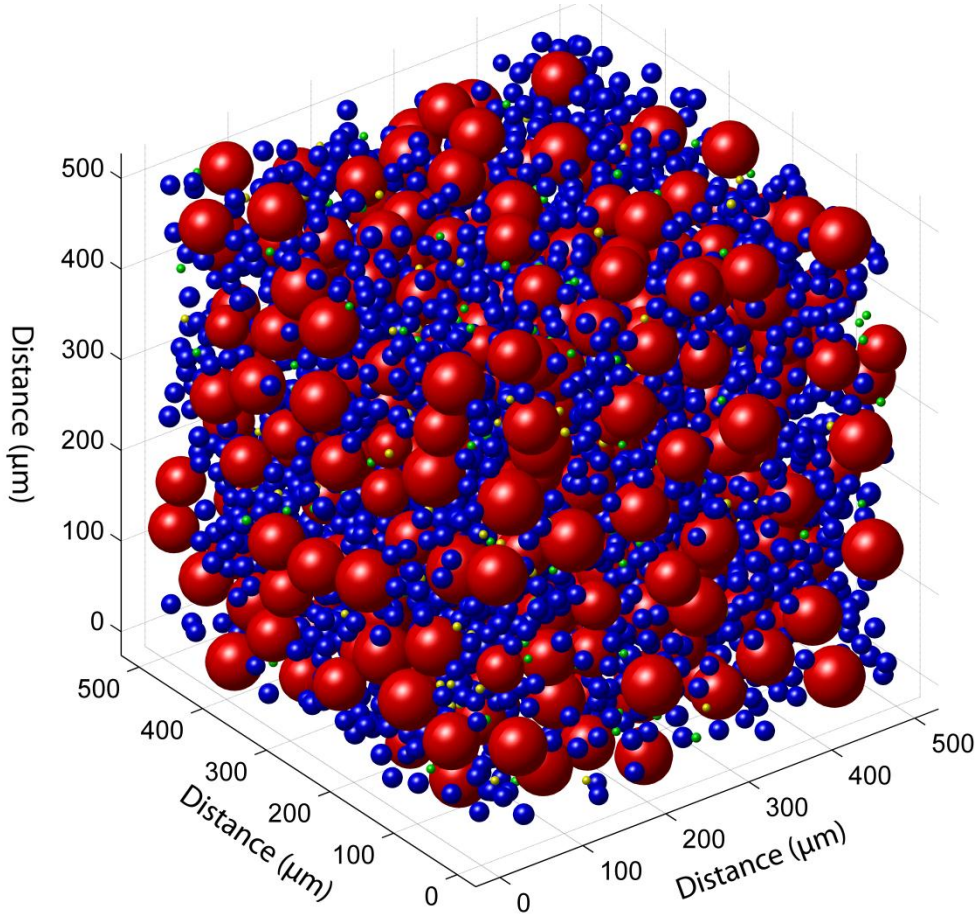



**Figure 27. Model of a section of layer V of Cerebral Cortex of brain created by using the density and volume provided in ( Rivara)**

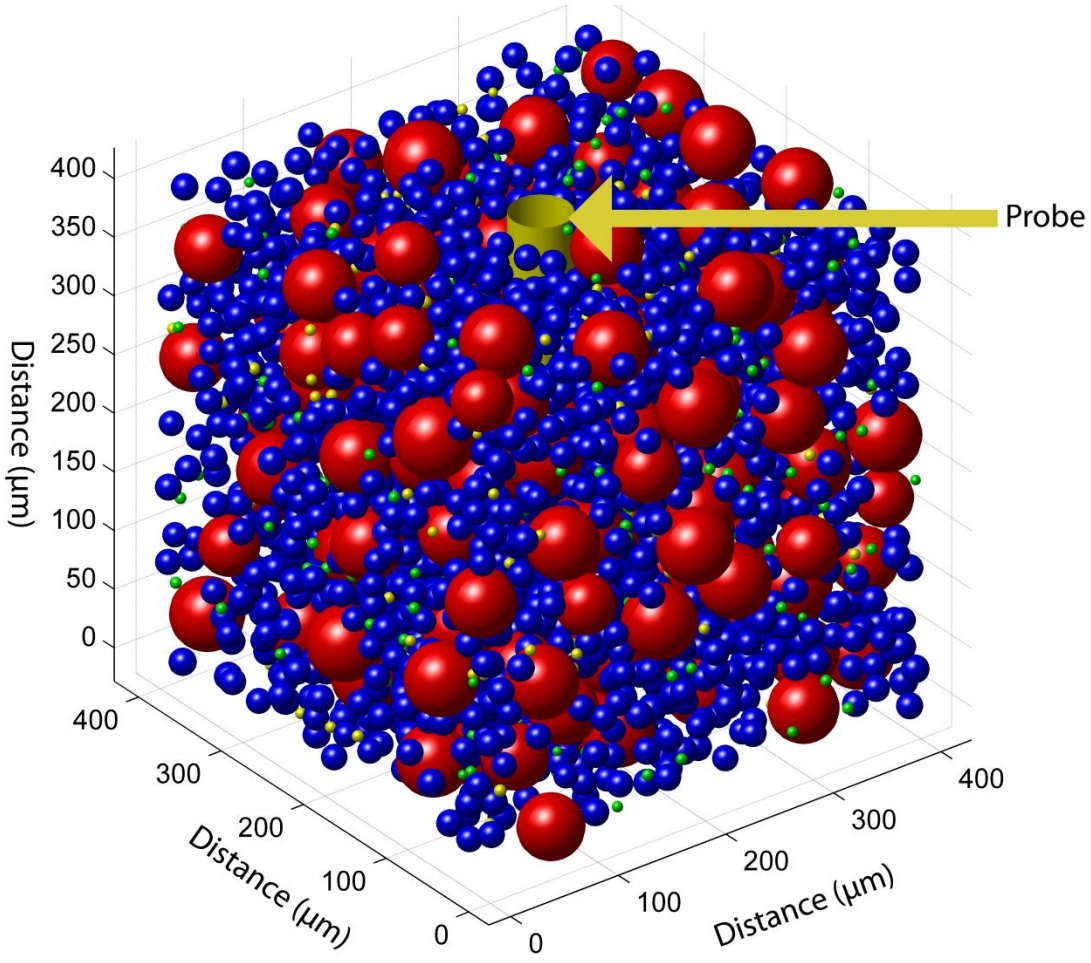



**Figure 28. Model of a section of layer V of Cerebral Cortex of brain with an electrode inside**

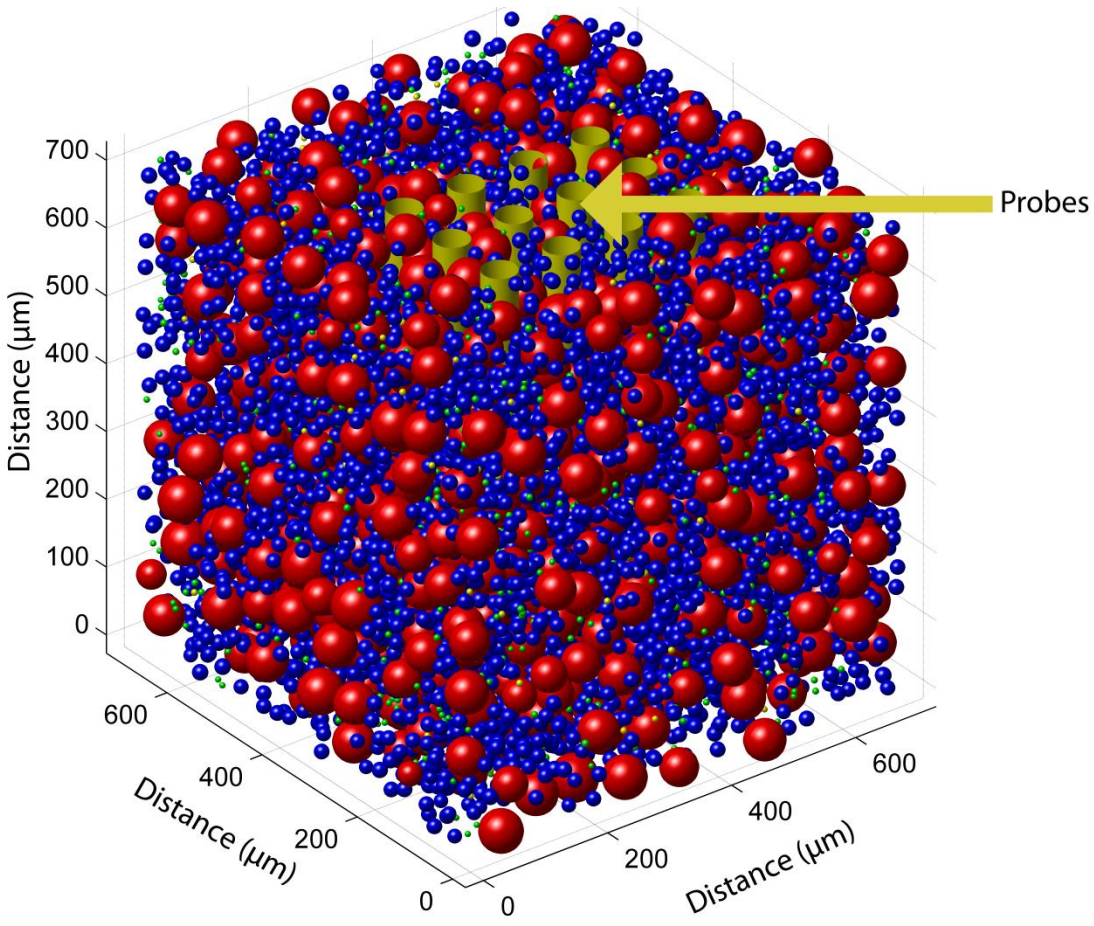



**Figure 29. Layer V of brain with 12 probes inserted**

. In Figure 28 a 50μm thick probe is inserted into the cells, as a result the density of neurons around the site of penetration is higher and it decreases further from the probe. Finally Figure 29 shows a model for layer V of cerebral cortex with 12 probes inserted. This model will be used throughout this dissertation in order to derive reliable spiking signals.

### C. Shielding of the probes

The model presented in last section is used to study the possible neural recording of the adjacent neurons on the inserted probe for different sizes of conductive tips.

In this step, four different cases are studied. In the first case, (represented in Figure 30-a) the probe is fully unshielded and had the length of 400 μm. As it can be seen the reading has many different spikes. Figure 30-b shows the modeled recording in case of a shorter electrode (200μm) in this case the spikes are more visible but the timing between them is very small which eliminates any chance of spike detection. Figure 30c has a 50μm unshielded region, which is equivalent to Utah array. As can be seen, three different spikes are detectable but the SNR is lower. Figure 30d represents the case when the probe is completely shielded and only the very tip of probe can receive signal, this case is really close to the case of Michigan array. In this case, the three spikes are completely visible and the LFP level is really low indicating a really high SNR. As discussed in (I. Obeid, P.D. Wolf) lowering SNR affects the spike detection accuracy more than different algorithms proposed for this purpose.

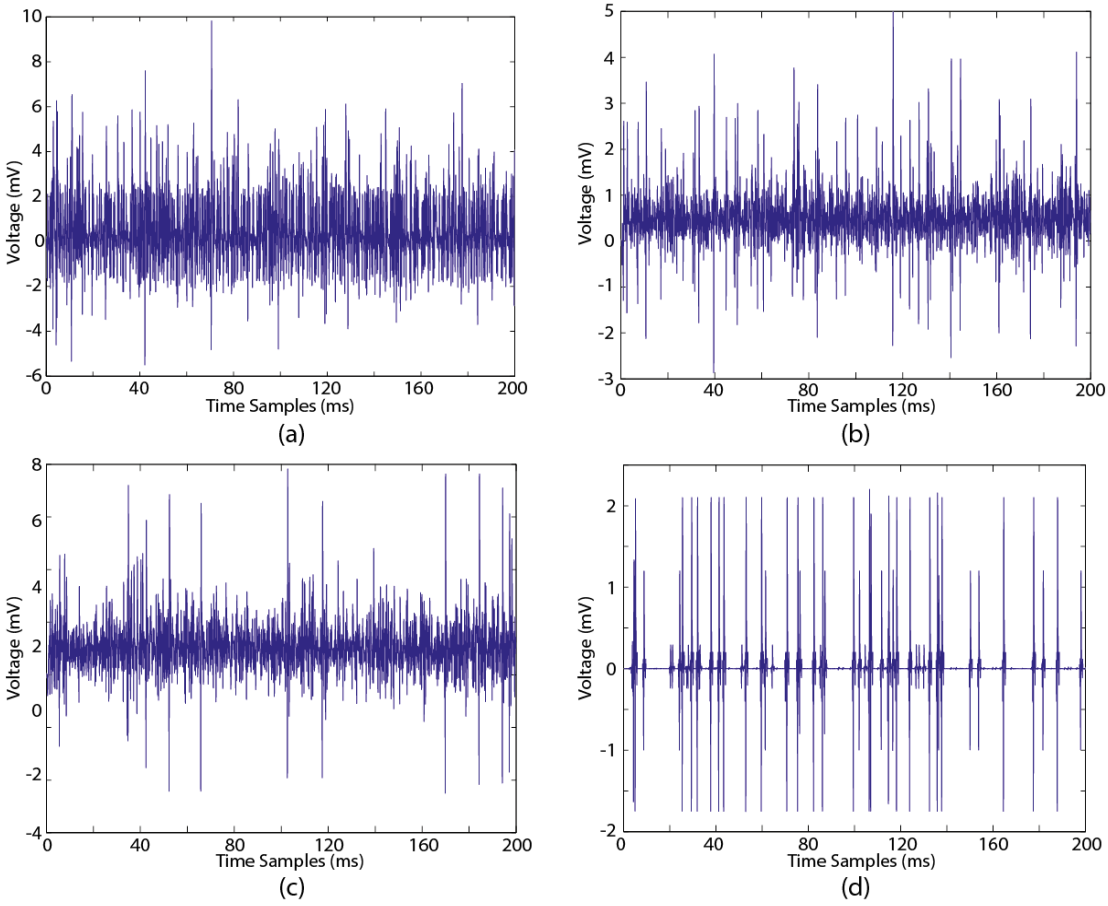



**Figure 30. Modeled neural recording signal for different electrode size and shielding (a)Recording for unshielded 400um long electrode (SNR = 3.67) (b)Recording for unshielded 200um long electrode (SNR = 3.86) (c)Recording for 150um shielded 200um long electrode (SNR = 4.07) (d)Recording for fully shielded 200um long electrode (SNR = 8.17)**

**Table 1 AVERAGE TIME BETWEEN SPIKES FOR DIFFERENT PROBE SIZE/SHIELDING**

| Size of probe / Size of conductive tip | Average time between spikes detected |
|---|---|
| 400um/400um | 1.3469ms |
| 200um/200um | 1.5007 ms |
| 200um/50um | 4.0716 ms |
| 200um/0um | 8.1676 ms |

As a side effect we can observe that as the number of neurons detectable by a probe increases the possibility of detecting more and more spike increases accordingly. Table 1 shows the average of times between spikes for each of the cases disused previously. This decrease in time limit for processing of the signal in spike detection method results in increase of need for more and faster signal processing units and higher clock rate, while it removes the advantage of spike detection by higher bandwidth needed to send even a small amount of information for each spike detected. This can be another reason to push the signal processing algorithm from computational extensive and power consuming methods to less computational methods with higher bandwidths, such as crosstalk removal methods.

## 5. Conclusion and Motivation

As we mentioned earlier improving the SNR is an important issue for effectiveness of different spike detection algorithms. However, as it was found in this experiment, shape and size of the probe has a drastic influence on the signal SNR

level. Additionally, we discussed the reasons behind selection of different probes and how limitation in fabrication and cortical environment such as different depth of target layer of cerebral cortex may influence the choice of probes. Some probes such as unshielded general purpose result in cases that spike detection would be obsolete and compression based signal processing or feature extraction based on different neuron's specification would take over. Furthermore, best cases of shielded probes result in at most 50%-60% of electrodes having automatically detectable spikes, the rest of the electrodes are neglected completely. Devising a low power analog computation method may result in development of hybrid chips that can divide the electrodes into two groups of spike-detectable and spike-undetectable electrodes. The first type of electrodes use common spike detection algorithms and the recording on the rest of the electrodes are transmitted, after low-power area-efficient analog compression, for human or a more advanced signal processing unit analysis. Furthermore the ratio of spike-detectable electrode will decrease drastically as we move from layer V of cerebral cortex, which has the biggest neurons, to the other parts of the brain. Thus, the need for analog compression increases. Finally, if we cannot devise a method for aligning the neurons with unshielded part of the probe, there is no choice other than using unshielded electrodes with analog compression. In next chapter we design low-power area-efficient analog compression algorithm for use in a fully implantable neural recording system.

# III. Removing Crosstalk from Multi-Probe Neural Activity Recorder

Power consumption, noisy environment and limited area leave in-vivo neural activity recording a challenge in the field of human machine interaction. Efforts have resulted in some promising progresses in more accessible area of nervous system such as cochlea and eye. But it leaves the main challenge to interacting with the cerebral cortex of human. In an effort to access more valid and detailed information from brain activities several groups try to advance a device which can be implanted inside the brain tissue to have more connection with specific neurons to observe and gather more accurate information. As it was discussed earlier insertion of the probe inside a neuron which has the maximum electrical activity will result in the death of the cell. So these implantable devices should only rely on recording hugely distorted and degraded extra-cellular electrical activity. These activities have been degraded because of the neuron shield and distorted because they are combination of many neighboring neurons firing with different rate and amplitude. Many of the research groups directed their research based on the assumption that "each probe can only read limited number of neurons activity," but as it was shown in previous chapter technological and environmental limitations prevents current probes to achieve that ideal environment. Consequently, the implantable device either should discard the information recorded from probes with more than three readable neurons or we should devise a method to maximize the useful information and remove the redundant

noise in order to send minimum amount of information out to computational environment.

## 1. Introduction

The correlation between signals is the source of redundant information between the signals, thus eliminating redundancy results in less waste. Consequently, fewer bits are needed to describe each sample of the signals. Mainly there are two different methods for eliminating the redundancy:

First method or K-transform, the signals are moved from one space to another space with less correlation. The common example of this method is image compression using DCT (Discrete Cosine Transform). In an image the pixels close to each other have high correlation. In DCT algorithm the array of signal is transformed using DCT, it is proven that an image in Discrete Cosine domain has lower correlation rather than in RGB domain. This results in coefficient of the new space having higher amount of information embedded in them. Consequently, transmitting coefficients with larger value and discarding smaller coefficients would result is smaller loss of information. This method is used in most of image compression algorithm such as JPEG. But this method needs a lot of processing power, and as a result high power consumption. This disadvantage prevents the applicability of this method in limited power budget in implantable neural activity recorders.

Second method is Prediction methods. Predictively of a signal is tightly related to the idea of redundancy the signal. This means that, the better we can predict

a signal from others the more redundancy the signal contains and less new information is contributed by other observations. Prediction can be used based on temporal or spatial redundancy, in temporal method the previous values of a signal is used to predict the current value of the signal. The correlation in signal value results in a decent prediction that can be used as a base for calculating the real value of the signal using the error of estimation. Since this error has discarded the temporal redundancy, it would be transmitted using lower number of bits. The spatial redundancy existence is the result of correlation between adjacent signals. For example in brain because a neuron can couple to more than one probe in its vicinity, the signal read from probes would have correlation. This correlation is the source of spatial redundancy. Using prediction method, we can estimate the value of the signal on one probe using adjacent signals and then the error in predication would have less redundant information and result in fewer number of bits needed for transmission.

## A. Information Theory

In information theory Entropy is defined as a measure of the uncertainty of a random variable. If the probability mass function of a discrete random variable (X) with alphabet $\chi$ is defined as $p(x) = Prob\{X = x\}, x \in \chi$ . The entropy of this discrete random variable is defined as:

$$H(X) = -\sum_{x \in \chi} p(x)\log(p(x))$$

There is an intuition behind this equation. Assuming $x_i$ is a member of alphabet which is least probability to occurrence, there is more information embedded in if it happens. For example if an unfair coin has the probability of 99% heads and 1% tail, during an experiment of many trials there is more information in occurrence of a tail than occurrence of a head. Function $\log\left(\frac{1}{p(x)}\right)$ has the property of giving higher value to the members of alphabet that has lower probability and as it can be observed the entropy is the expectation of this random variable. There is a corresponding definition for differential entropy. Differential entropy is defined for continuous random variable with probability density function f(x), and it os calculated using the following equation:

$$h(X) = \int_S f(x)\log(f(x))dx$$

In which S is the support set of random variable. If the random variable has normal distribution, it can be proven than the entropy is proportional to log of variance of the normal distribution. That is: if $X \sim \Phi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x^2}{2\sigma^2}}$)

$$h(\Phi) = \frac{1}{2} log 2\pi e\sigma^2$$

Joint entropy is defined as minimum amount of information embedded in two random variables, and it is defined as:

$$H(X,Y) = -\sum_{x\in\chi}\sum_{y\in\gamma} p(x,y)\log(p(x,y))$$

Or in continuous domain:

$$H(X,Y) = -\int_{S_X}\int_{S_Y} f(x,y)\log(f(x,y))$$

Another useful definition is conditional entropy which is defined as expected value of the entropies of the conditional distributions, averaged over the conditioning random variable, and it is formulated as:

$$\begin{aligned} H(X|Y) &= -\sum_{x\in\chi} p(x,y)\mathrm{H}(Y|X=x)) \\ &= -\sum_{x\in\chi}\sum_{y\in\chi} p(x,y)log(p(x|y)) \\ &= E\log\big(p(X|Y)\big) \end{aligned}$$

Or in continuous distribution:

$$h(X|Y) = -\iint f(x,y)\log\left(\frac{f(x,y)}{f(y)}\right)dxdy$$

The intuition behind these definitions is that if two random variables are known, the information in them is equal to jointed entropy of two variables. Which is equal to entropy of one random variable plus the conditional entropy of the second one. Or:

$$H(X,Y) = H(X) + H(Y|X)$$

One important theorem derived from conditional entropy is that information doesn't hurt or in other words:

$$H(X|Y) \leq H(X)$$

The relative entropy is a measure of the distance between two distributions. It shows the inefficiency of considering two distributions are equivalent. For example if we estimate one random variable with the other random variable, we need as much as relative entropy more information to recover original random variable. Relative entropy of two random variables is defined as:

$$D(p||q) = \sum_{x \in \chi} p(x) \log\left(\frac{p(x)}{q(x)}\right)$$

$$D(f||g) = \int f \log\left(\frac{f}{g}\right)$$

Notice if two random variable are equal the relative entropy of them is equal to zero, in other words we can estimate one from the other without any extra information.

Mutual information of two random variables is defined as a measure of the amount of information that one random variable contains about another random variable. It is the reduction in the uncertainty of one random variable due to the knowledge of the other.

$$I(X;Y) = \sum_{x\in\chi}\sum_{y\in\gamma} p(x,y)\log(\frac{p(x,y)}{p(x)p(y)})$$

$$I(X;Y) = \int_{S_X}\int_{S_Y} f(x,y)\log(\frac{f(x,y)}{f(x)f(y)})$$

The joint entropy of two random variables can be expressed by entropy of each of them with the following equation:

$$H(X,Y) = H(X) + H(Y) - I(X;Y)$$

So increase in mutual entropy of two random variables would result in decrease of extra information needed to estimate one from another one. Using Jensen inequality it is proven that mutual information between two random variables are greater or equal to zero, or:

$$I(X,Y) \geq 0$$

To give more insight, another theorem defines the bounding of the entropy of a random variable:

$$0 \leq H(x) \leq \log|\chi|$$

This proves that the maximum entropy of a random variable is the log of the cardinality of its alphabet. One should notice that the equality situation happens when the distribution is uniform. In addition we can prove:

$$H(x) = \log|\chi| - D(x||u)$$

In which u is uniform distribution over alphabet $\chi$. Intuitively, if the distribution for a random variable is close to uniform the amount of information included in that random variable is higher.

These definitions can be extended to more than two random variables. For example, the conditional differential entropy of more than two random variables is defined as:

$$h(X_1, X_2, \dots, X_n) = -\int f(x^n)\log(f(x^n))dx^n$$

In order to relate the entropy to communication, the following theorem and definitions should be considered.

Data compression normally is achieved by assigning short codes to the most frequent outcome of the random variable and longer codes to the less frequent ones. So the expected length of L(C) of a source code C(x) for random variable X with probability mass function P(x) can be derived as

$$L(C) = \sum_{x\in\chi} p(x)l(x)$$

In which l(x) is the length of the code associated with x. In every data compression the main effort is to minimize this expected value.

In addition it has been proven that if we want to code a large sequence of random variables ($X^n$) with probability function of p(x), we can find a binary invertible coding which for $\varepsilon \geq 0$;

$$E\left[\frac{1}{n} l(X^n)\right] \leq H(X) + \varepsilon$$

For a coding which is reversible it can be proven that the expected length of the code would be greater or equal to entropy; Based on last two theorem one can deduce that the optimal length of a reversible coding for a random variable is

$$H(X) \leq L < H(X) + 1$$

In other words, the expectation of the length of coding would be would be equal to H(X) for each random variable. Or on average, the large sequence $X^n$ can be coded by n.H(X) bits.

As we mentioned earlier the entropy of a Gaussian random variable is proportional to the variance of the probability mass function of the random variable. Thus in order to minimize transmission bit-rate of a Gaussian signals it is required to minimize the variance of its distribution.

### B. Linear approximation

Prediction is defined as a statistical estimation method where one or more random variables are estimated from observation of other random variables. For example prediction of one vector $Y = (Y_1, Y_2, \dots, Y_k)^t$ using another observed

vector $X = (X_1, X_2, \dots, X_n)^t$. Notice that the dimension of the vectors can be different, for example if n = k = 1, prediction is a simple derivation of one random variable from another one. Or if only k equals to one, we predict one random variable from a set of other random variables, as we discussed above in our spatial prediction we can predict one signal value from the neighboring signals' values. If prediction is defined as a function $\widehat{Y} = \hat{Y}(X)$, the best prediction should be defined and derived. The most common measure of performance of a predictor is the mean square error, defined as:

$$\begin{aligned} \epsilon^2 &= E(||Y - \hat{Y}||^2) \\ &= \sum_{i=1}^{k} E[(Y_i - \hat{Y}_i)^2] \end{aligned}$$

This measure can be viewed as how much energy of the signal reduced by removing the predictable information from it. Consequently, the better predictor is the one which has the lowest MSE. There are two main class of predictors; linear predictors and unconstrained predictors. The first class constrains the predictor to be a linear operation on observed vector. And second class is unconstrained. Thus the linear predictor is just a subset of unconstrained predictors. A predictor is called an optimum predictor within a class of predictors if it minimizes the MSE over all predictions in the given class. As it was mentioned any predictor can be spatial or temporal based on the definition of X and Y. It can be proven than in non-Gaussian processes linear predictors cannot fully extract all the possible correlation for prediction, but it can be proven in Gaussian processes linear predictors are the

optimum predictors. In other words in the special case of Gaussian process the optimal estimator reduces to linear combination of the observed values.

A linear predictor is defined as $\hat{Y} = AX$ in which $A = (a_1, a_2, \dots, a_k)^t$ and $\widehat{Y_k} = a_k^t X$, so the performance of this predictor can be calculated as:

$$\epsilon^2(\hat{Y}) = \sum_{k=1}^{N} E[(Y_k - a_k^t X_i)^2]$$

In order to find the optimal linear predictor, matrix A should be defined in a way that minimizes the $\epsilon^2(\hat{Y})$. Since there is no interaction among the terms in the sum, this can be achieved by minimizing each term $E[(Y_k - a_k^t X_i)^2]$ over $a_k$ . So we define performance for each term separately.

$$\epsilon^2(a) = E[(Y - a^t\ X\ \ )^2]$$

So mathematically we can calculate the derivative of $\epsilon^2(a)$ and assign it to zero, and obtain the global minimum.

$$2\sum\left(Y - a^t\ X\ \ \right)(X) = 0$$

$$a^t\ = \frac{\sum YX^t}{\sum X^2}$$

Or the optimal predictor coefficients is the solution of equation

$$a^t R_x = E(YX^t)$$

In which $R_x = E(XX^t)$.

A finite memory linear prediction is defined as a predictor that the future value of a stationary random variable is predicted based on the limited number of its past values. Spatial linear prediction can be considered a finite memory linear prediction, in which the previous values are the neighboring random variables. To define finite memory predictor based on m last values, we need a predictor function

$$\widehat{X_n} = f(X_{n-1}, X_{n-2}, \dots, X_{n-m})$$

In which the prediction minimizes the MSE

$$E[(X_n - \widehat{X_n})^2]$$

We can prove that the optimal function would be

$$\widehat{X_n} = E(X_n | X_{n-1}, X_{n-2}, \dots, X_{n-m})$$

This can be nonlinear. The disadvantage of this solution is that it might not have a closed form expression. Additionally, computation of a conditional expectation would require an explicit knowledge of joined probability distribution of m+1 consecutive samples. But it can be proven in special case of Gaussian random variable the optimal estimate can be reduced to a linear combination of past values or in our case the neighboring signals.

$$\widehat{X_n} = \sum_{i=1}^{m} a_i X_{n-i}$$

## 2. Analysis of Synthesized Neural Activity

In the previous chapter we created a model for human brain and synthesized a set of neural activity recording based on different criteria of probes. In this section we look at these signals as random variables and try to apply the concepts defined in last section on these signals. This motivates us to implement a new algorithm which can remove spatial redundancy of signals and minimize the embedded information, consequently prepares the signals for a lower bitrate transmission module.

### A. Nature of neural signal

As it was discussed in previous section discovering the probability density function of a random variable is an important step in analyzing the random variable. For this purpose the 12 different constructed 10 sec simulation signal was used to derive the probability density function of neural activity. Figure 31, Figure 32, and Figure 33 show the PDF and signal for twelve probes shown in Figure 29. As it can be seen, the probability mass function can be estimated perfectly by a zero mean Gaussian function. Table 2 shows the average and variance of the mapped Gaussian probability density function. In addition Table 2 presents the entropy of each of these signals. One observation in Figure 31, Figure 32, and Figure 33 is larger variance corresponds to increased number of distinguishable spikes. Additionally, it can be observed in Table 2 increase in the variance of the random variable, results in increase in entropy of that random variable.

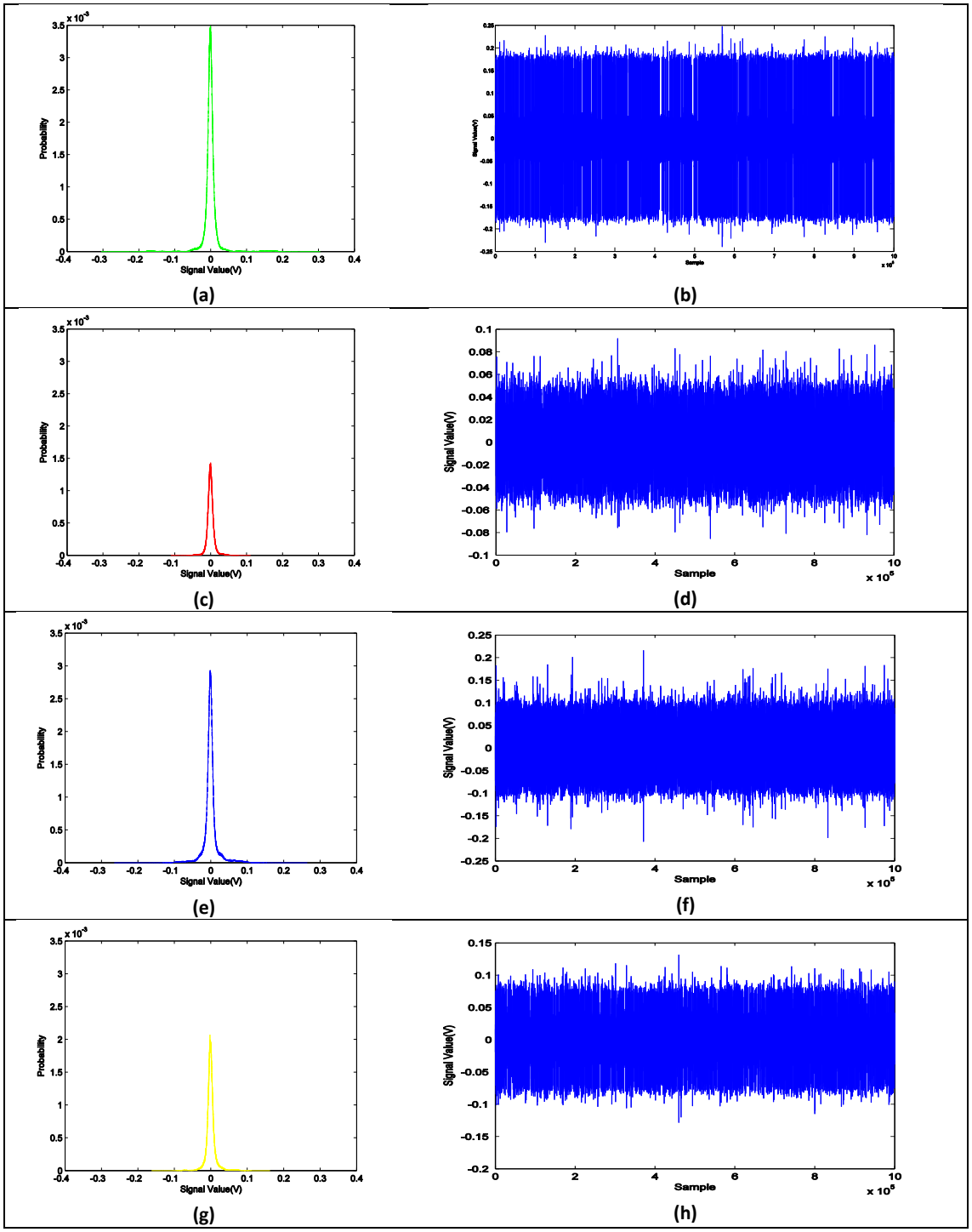



**Figure 31. Right the synthesized signal Left the PDF of the signal Part1**

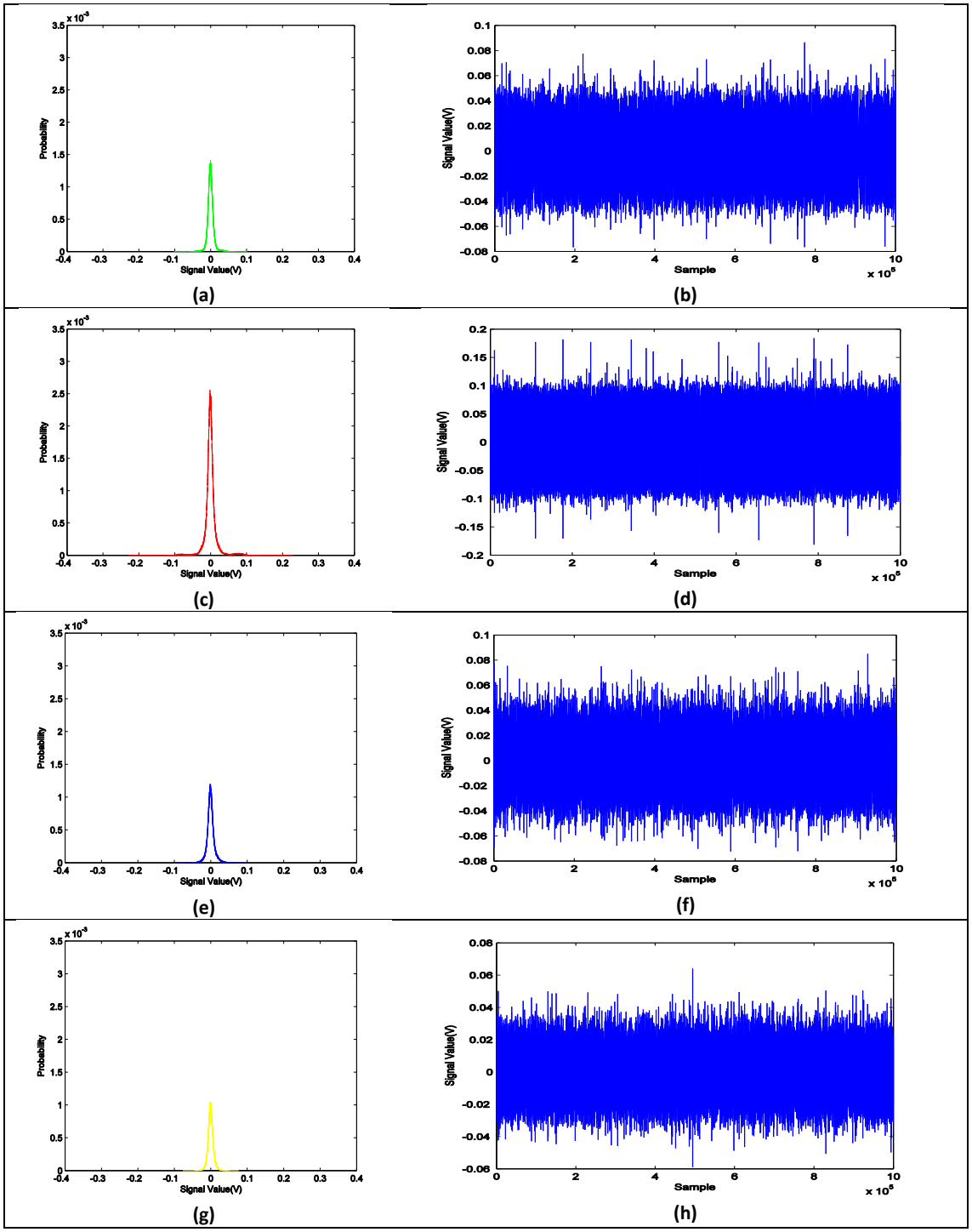



**Figure 32. Right the synthesized signal Left the PDF of the signal Part2**

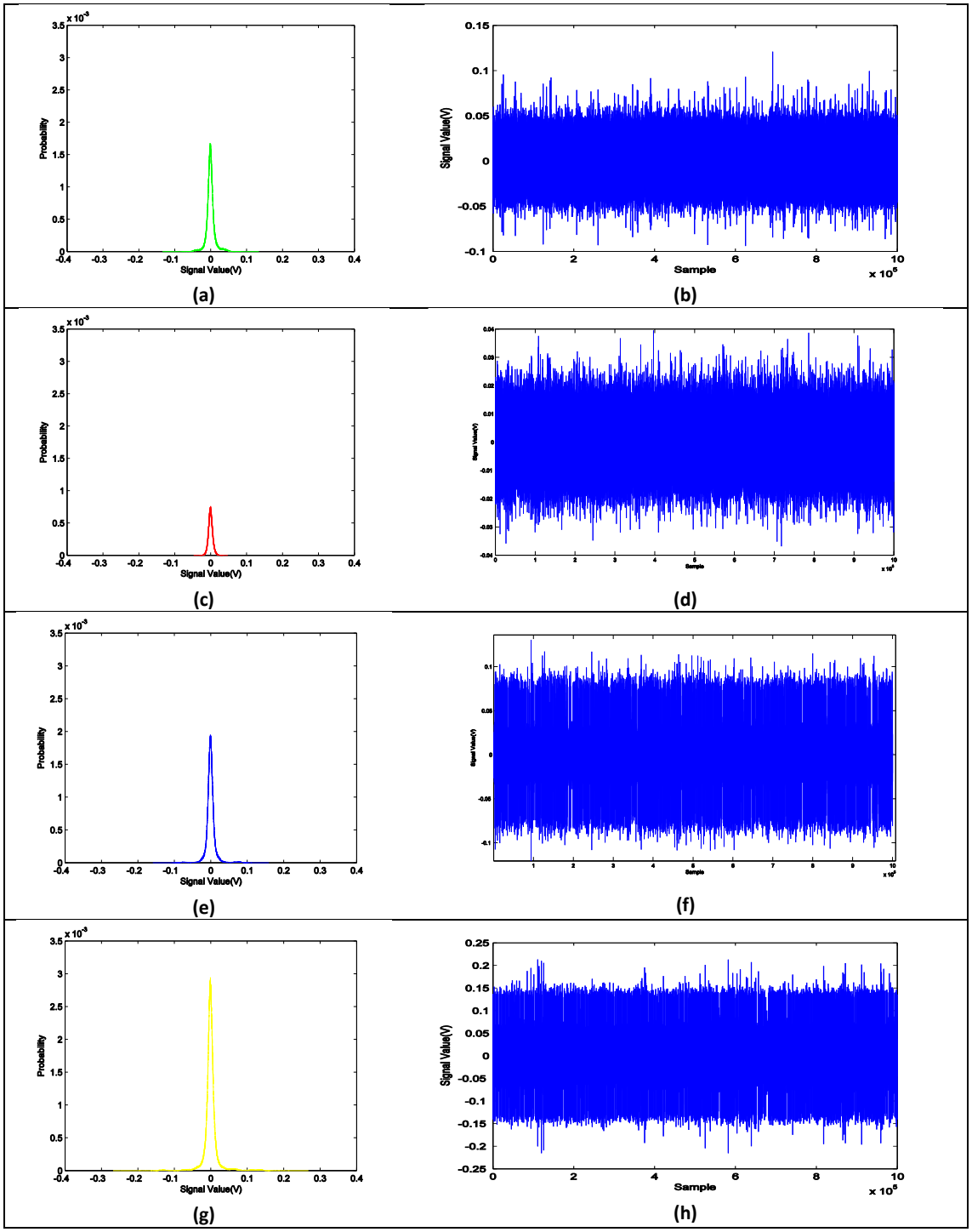



**Figure 33. Right the synthesized signal Left the PDF of the signal Part3**

**Table 2. The Average, Variance and Entropy in each of signal observed from twelve probes in Figure 29**

| Probe | Mean | Variance | Entropy |
|---|---|---|---|
| 1 | **3.2 µ** | **548µ** | **9.55** |
| 2 | **3.1 µ** | **96µ** | **9.15** |
| 3 | **3.4 µ** | **396µ** | **9.78** |
| 4 | **3.0 µ** | **167µ** | **9.36** |
| 5 | **3.0 µ** | **79µ** | **9.03** |
| 6 | **3.4 µ** | **308µ** | **9.64** |
| 7 | **3.3 µ** | **102µ** | **9.29** |
| 8 | **3.0 µ** | **60µ** | **8.98** |
| 9 | **3.1 µ** | **134µ** | **9.34** |
| 10 | **3.0 µ** | **38µ** | **8.69** |
| 11 | **3.3 µ** | **164µ** | **9.29** |
| 12 | **3.2 µ** | **427µ** | **9.66** |

As we mentioned earlier, correlation is the source of redundancy in a random variable. The main source of correlation in our probes is the neurons which couple on more than one adjacent probe. This coupling shows itself as crosstalk noise between two probes' recording. In order to scientifically observe the effect of this correlation in our synthesized model, we try to derive the joint probability mass function of two probes. Figure 34 represents the 3D representations of probability density function of probe 7 and probe 6. This probability density function is used to derive the joint entropy of the two probes.

**Table 3. Joint entropy and mutual entropy of two probes**

| Property | Value |
|---|---|
| H(X,Y) | **14.40** |
| I(X;Y) | **4.5** |
| D(Y\|\|X) | **4.76** |

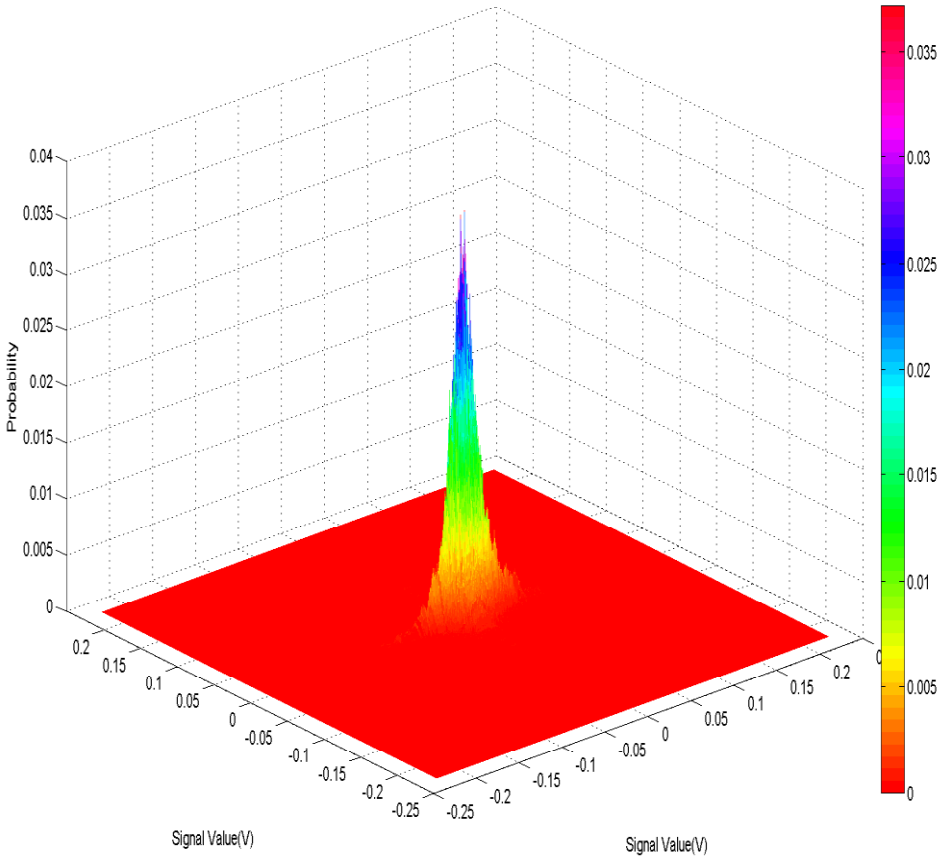



**Figure 34. Joint probability density function of two adjacent probes.**

Table 3 shows the joint entropy and mutual entropy and conditional entropy of two observed probes. As you can see out of 20 bits needed to transmit each of the probes separately, 5 bits are redundant and can be removed by an algorithm that can remove the correlation of these two signals. In the rest of this thesis we concentrate on developing and implementing a hardware friendly and low power design of the algorithm that can eliminate the crosstalk between neighboring signals.

## 3. Time-based Cross-talk Removal Algorithm

As it was mentioned before transform based methods of redundancy removal rely heavily on the frequency domain processing of the signals. But in a limited area and power environment such as implantable neural recording device these methods lose their appeal due to the high computation needed in transformation of time domain signal to frequency domain. On the other hand if we are able to undergo the noise removal process in time domain using spatial prediction, we can avoid transformation step. One important assumption required for using this method is the effect of phase of signals should be discarded. This is achieved by the fact that signals are derived from close enough probes this effect can be ignored. In time based cross-talk removal we try to extract use linear estimation of neighboring signals to construct each signal. As it was discussed before the predictor coefficients represent, the ratio of the one signal that can be estimated by the other one. As an example if second signal is only a deteriorated version of the first signal the predictor coefficient will be the ratio of degradation. In a noisy environment of brain this ratio is not easy to calculate, so a new method for deriving this ratio is minimizing MSE or in other

words to minimize the error between the first signal amplified by the coefficient and the second signal. In other word find k that minimizes:

$$MSE = \int (x_2(t) - kx_1(t))^2 dx$$

Mathematical method for solving this equation is to take the derivative of this equation and assign it to zero to calculate the k which gives us minimum value, or:

$$-2 \int \left(x_2(t) - kx_1(t)\right) x_1(t) dt = 0$$

$$k = \frac{\int x_1(t) x_2(t) dt}{\int (x_1(t))^2 dt}$$

In our algorithm we split the time into two phases; in the first phase or learning phase the prediction coefficient of each pair of signals are calculated, in the second phase these coefficients are used to estimate the second signal from first signal.

The defined mathematical method for calculating coefficients has its draw backs in real world; first this method only works if we have access to the whole range of signal. Second, limitation in fabrication process prevents this design to be fully implementable in circuit level. For example, integration in long period of time is impossible due to limit on the size of capacitors that can be developed in current chips. Consequently, we modified the algorithm to be more hardware friendly. This new algorithm uses gradient descent for calculating the value of k, this result in

shorter period of integration and need for smaller integration capacitors. In the next two subsections we introduce the modification on the algorithm in order to overcome these two limitations.

### A. Periodic learning

As we mentioned before, we the mathematical method of deriving optimal prediction coefficients, requires the learning unit to have access to the whole signal. But in real world this is almost impossible. This the reason behind splitting the time into learning and transmitting phases. It can be assumed that one learning phase is enough for transmitting the signal for the rest of the time. But in this chapter we use the synthesized signal to prove periodic derivation of the coefficient would result in decrease of overall information of the signal. At first we find the optimum coefficient for the long period of signal, based on that we calculate the entropy of the residue signal. In the second attempt we split the time into 10 sections and in each section we calculate the coefficient separately. In this case we again calculate the entropy of residue for each section and for the whole signal. Comparison between these two entropies shows that periodic learning results in smaller amount of information and consequently lower bitrate for transmitting the signal.

In this experiment we try to estimate signal on probe 6 using signal on probe 7. The entropy of the signal on probe 6 without any estimation is equal to 10.92. After learning phase the global prediction coefficient is derived to be equal to 2.5. The estimation result from this prediction coefficient result in signal with entropy of 9.92.

In the second step of this experiment we split the time into 10 equal steps and learning phase is done in each of these phases separately. Table 4 summarizes the value of local prediction coefficient and their effect on the local entropy. Now we can use these new coefficients to derive the global entropy of the residue, which is equal to 9.7. As it can be seen the entropy is lower when we use periodic learning instead of one step of learning. In addition, periodic learning would require smaller learning phase, because, if in one of the learning phases we could not drive the optimal prediction coefficient, it would only effect a small period of our transmission phase. Subsequently, in the following learning phase a new prediction coefficient would be calculated.

**Table 4. The local predication coefficients and their effect on entropy**

| phase | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Initial Entropy | 10.96 | 11.15 | 11.07 | 11.02 | 11.19 | 11.08 | 11.13 | 10.99 | 11.18 | 10.97 |
| Prediction coefficient | 3.25 | 3.25 | 2.5 | 3.5 | 3.25 | 3.5 | 4 | 3 | 3.75 | 2.5 |
| Residue Entropy | 10.28 | 9.90 | 10.13 | 9.99 | 10.34 | 10 | 9.98 | 9.87 | 10.04 | 10.40 |

## B. Hardware friendly algorithm

In order to overcome the implementation limitation of our design, we modified the algorithm to include subtle changes which makes it more implementation friendly. In the modified algorithm we integrated the error in two small consecutive time period, in each of these periods k differs slightly, the comparison between these two errors shows if the change in k increase or decreases

the error. Consequently, the algorithm can decide whether the change, resulted in k to get closer to minimize the error or it was a wrong step. Based on this observation k is changed to a new value. This process continues until k reaches a minimum and start oscillating around one value, or the time limit for calculating k is over. This method can be summarized in following equation:

$$\int_{t_0}^{t_1} (x_2\text{-}kx_1) - \int_{t_1}^{t_2} (x_2\text{-}k'x_1)$$

If the result of second integration is closer to zero than the first integration, it shows that k' is a better approximation of optimal predictor coefficient and in the next cycle, the same trend should be followed in calculation of next k. On the contrary, if second error integration is larger than the first integration, the trend was wrong and in the next step k should be updated to the initial value. Figure 35 shows the basic idea of this method.

Due to the limitations in environment and implementation, small modification is implemented in the algorithm. First modification in algorithm is to divide two signals from adjacent probes by each other, the reason for this division is to normalized the error for integration level, second modification is a result of the problem with device matching in analog devices, which prevents two integrator operate similarly. This limitation pushes us to use same devices in both time periods of error calculation, including same integrator for both periods.

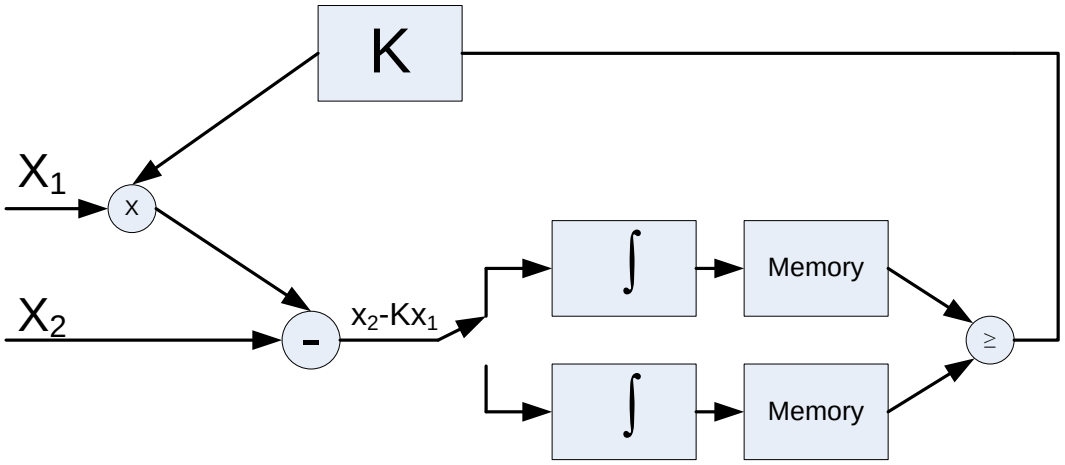



**Figure 35. Basic block diagram of gradient descent algorithm**

Additionally we have limited capability of storing analog values in the analog systems. This limitation prevents us from storing the value of first period while the value of error in second period is calculated. In order to overcome these two limitation we use the same integrator with different polarity in each period, which means if in first period the integrator input is a-b in the second period the input is b-a. So, the value of error in first period doesn't have to be stored anywhere and the same integrator is used for calculation of both time periods. The final result stored in the integrator would be the difference of error integration between two phases. This new design will result in a new challenge which the value of the second phase cannot be stored and used as a first phase error integration for the next phase. In order to overlap these two phases, we introduce a second integrator which has the opposite phase of the first integrator; consequently although these two integrators might suffer from mismatch because each value of integration is compared within same integrator

the final result is independent of processes matching. At the same time two phases overlap which results in faster deriving of the prediction coefficient.

Other main concern is the constant which should be stored and finally transmitted to the receiver to reconstruct the original signal. For this section of our design we moved to digital design. This is due to the fact that digital design lends itself to easily implementing a counter and DA convertor. Additionally, due to the fact that the final result should be stored and transmitted digitally. Another section in our design which is digital is the control unit which harmonize different analog subsection. Figure 36 shows the overview of block diagram of the modified algorithm. In the next sections we present the model and then introduce two versions of prototype of this design that is used for the purpose of proof of concept.

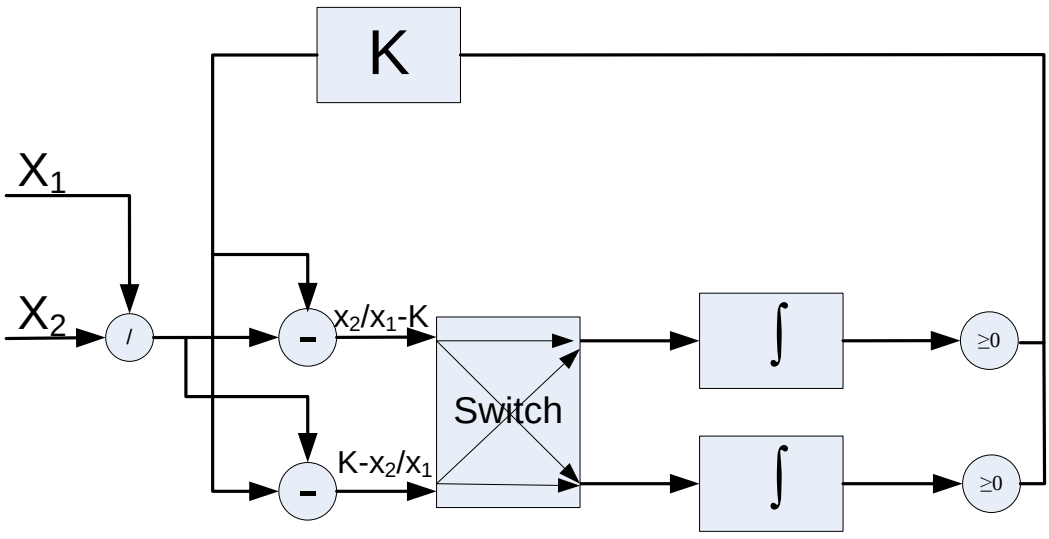



**Figure 36. Block diagram of modified gradient descent algorithm**

# 4. Model of Time-based cross-talk removing algorithm

In this section we briefly present and overview of the components of the model of the algorithm in Simulink. We will start with small basic block diagram of the design and then continue to the bigger picture.

## A. JK-Flip-flop

JK-FF with asynchronous set and reset, is one of the main digital elements used in our design. This block is used for storage, synchronization, and counter design. The reason behind selecting this storage unit is its capability to act as a Toggle-Flip-flop or D- Flip-flop. Additionally asynchronous set and reset provides us with extra flexibility in our design. Figure 37 shows the block diagram of the Flip-flop modeled in Simulink. As it can be seen it is modeled as a combination of a generic JK-Flip-flop and a SR-Flip-flop. The asynchronous part of the Flip-flop is modeled by the fact that SR-Flip-flop overrides the JK-Flip-flop in situations where Set or Reset signal is activated.

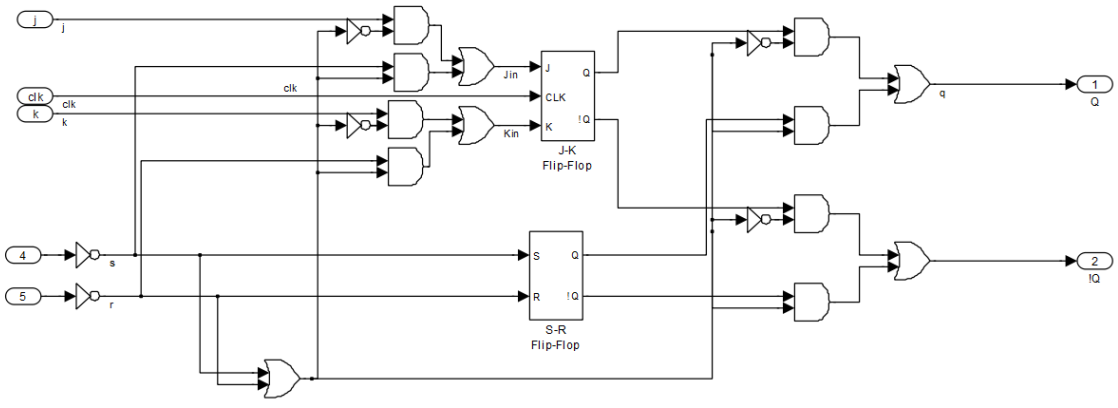



**Figure 37. Block Diagram of JK-Flip-flop with Asynchronous Set and Reset in Simulink**

## B. Up-Down Counter with Saturation

The Flip-flop introduced in the last section is used to create an up-down counter which stores and calculates the value of prediction coefficient based on the results from analog computation. Since the counter has limited number of bits it should saturate to a minimum or maximum value, this was implemented by using the asynchronous set & reset.

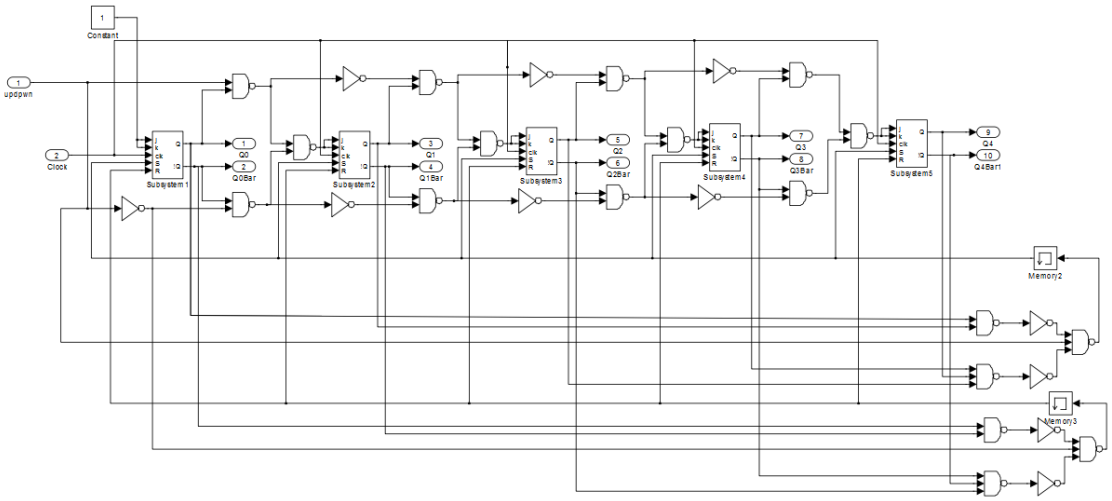



**Figure 38. Block Diagram of Up-down Counter with Saturation**

Figure 38 shows the block diagram of the up-down counter as you might notice because Simulink lack of delay in its logic blocks, delay modules are added in the feedback loops. In our design we consider 2 least significant bits as fraction part, and the saturation happens on 7.75 and 0. So the range of prediction coefficient is between 0 and 7.75 with resolution of 0.25.

## C. Digital Control Unit

The heart of control unit a 5-bit counter which synchronizes the control unit and is used to create control signals for computational modules, Figure 39 shows the model of this counter, it will count from 0 to 31 this period is split into different periods which is used to activate or deactivate different subsection of the system.

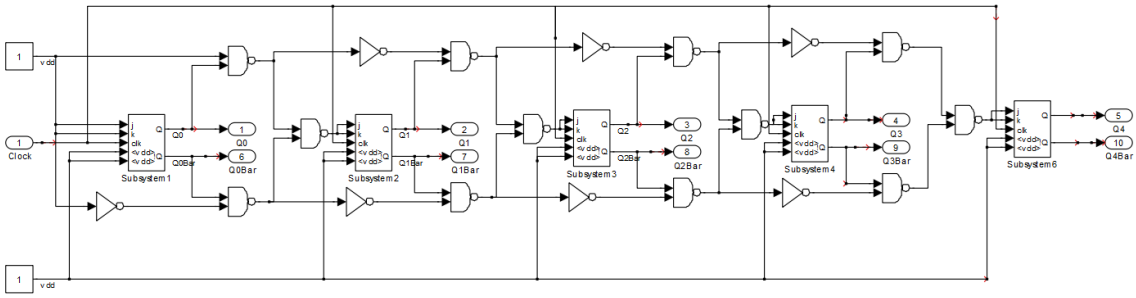



**Figure 39. Control Unit Counter**

The first control signals are two non-overlapping clocks. The purpose of these two signals is to direct correct input to the integrators. As we mentioned in the last section in first phase of integration $\frac{x_2}{x_1} - K$ is fed to integrators and in second phase $K - \frac{x_2}{x_1}$. These two non-overlapping signals are used to direct correct input to the integrators. Figure 40 shows the block diagram of non-overlapping signal generator. As you can see two JK-Flip-flops are placed in the output of the signals. The purpose behind these Flip-flops is to prevent any jitter to accidentally propagate through the control unit. This will delay the signals by one clock cycle that is considered in our calculations.

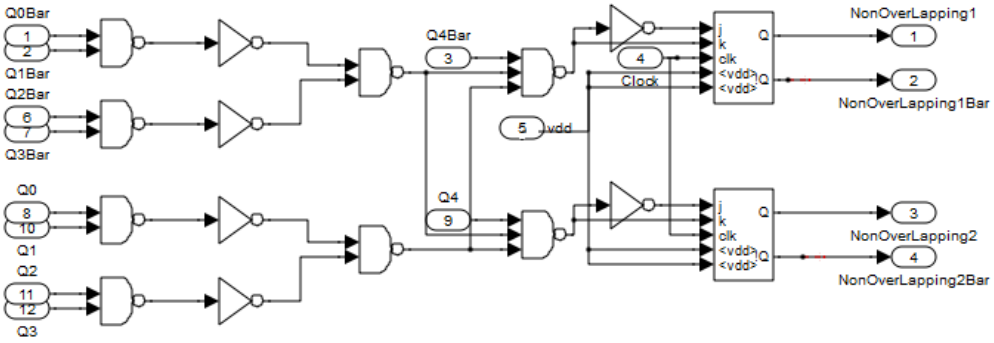



**Figure 40. Block Diagram of non-overlapping phase generator**

The next control signals are the reset signals for the integrator, after each two phase of integration the integrators should be reset to zero. For this purpose, the time period which is not covered in any of the non-overlapping phase control signal is harvested by digital circuit shown in Figure 41, and based on the value of "skipper" one of the reset signals would be activated.

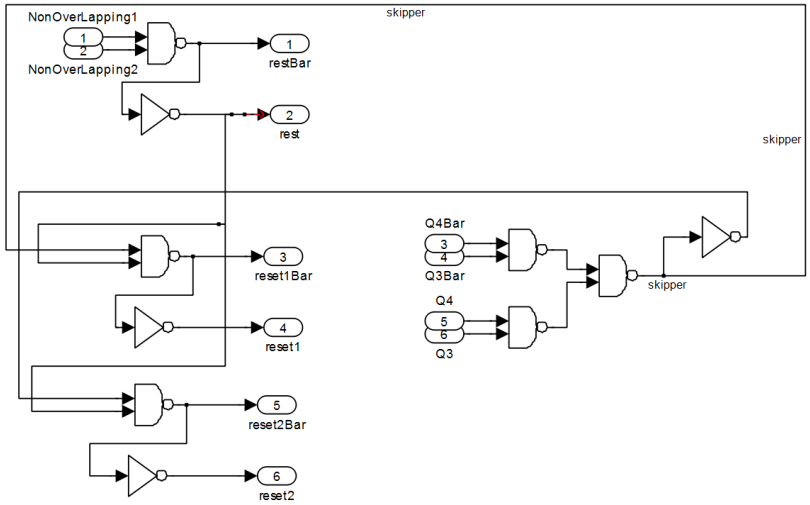



**Figure 41. The Block Diagram of the Reset Signal Generation Circuit**

In order to have most precise comparators, they need to be calibrated before each comparison phase. A digital control circuit is responsible to activate the comparison phase of the comparator. Since the comparison is needed only in the very last phase of integration the circuit will activate comparators in that specific time.

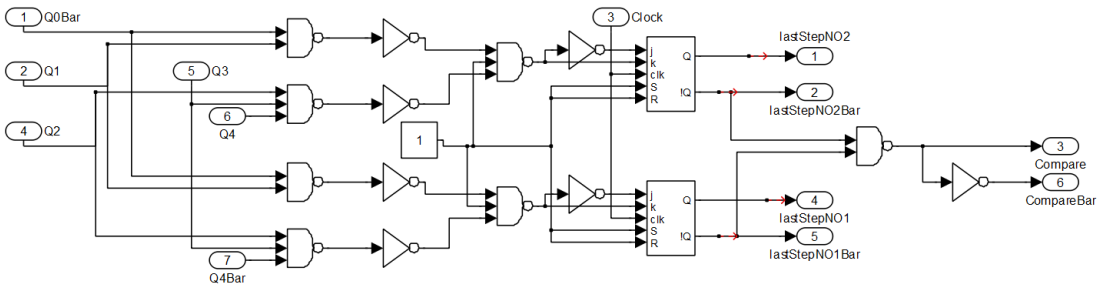



**Figure 42. Block Diagram of Comparator Activation Signal Generator**

As it is shown in Figure 42, two signal lastStepNO1 and lastStepNO2 calculate the very last clock cycle of each of the non-overlapping control signals, the conjunction of these two signals are used to activate the comparators.

Shown in Figure 43, last section of the control unit is a sequential circuit which decides whether the last step of change of prediction coefficient was beneficiary or it was a wrong decision, which is calculated in signal *Wrongright*. This signal is then used to calculate the *UPdown* signal fed to the UP-down counter. The *cmp1* and *cmp2* signals are provided by the two comparators. The current results and previous phase results of comparators are used to derive the direction in which the counter should change the prediction coefficient.

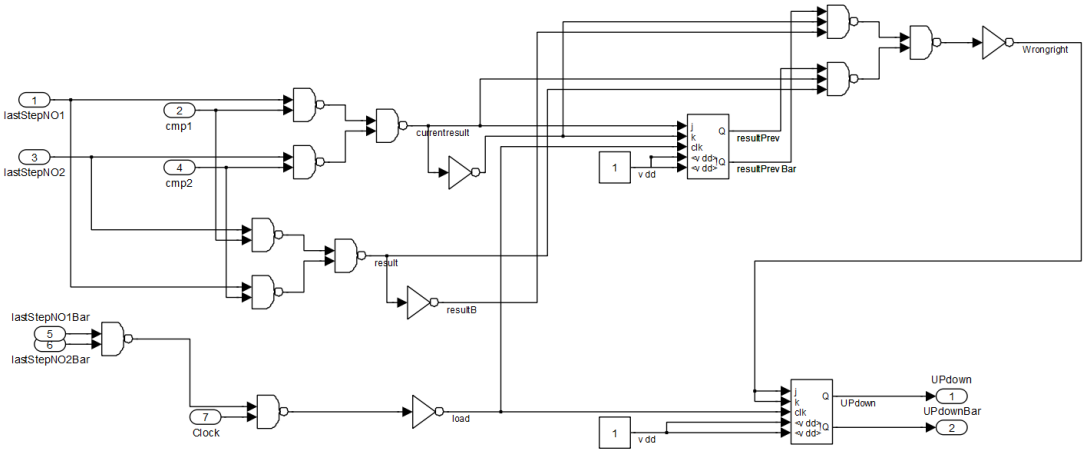



**Figure 43. The Block Diagram of Counter Direction derivation Unit**

On a final note about the control unit, since this design would not consider the case where two input signals have different polarities, the sign of signals are used to activate or deactivate some parts of the design.

## D. Computational Section of the Model

In this step the computational section of the design is added to the control unit, Figure 36 shows the general guideline for this step which is implemented as it is shown in Figure 44. The two signals *A* and *B* are fed to the model and the prediction coefficient is calculated in limited number of clock cycles. The DAC block is a normal Digital to Analog Converter the output of this block is divided by 4 to derive the correct prediction coefficient. The selection of correct input for integrator based on the phase is done in *Switch1* and *Switch2*.

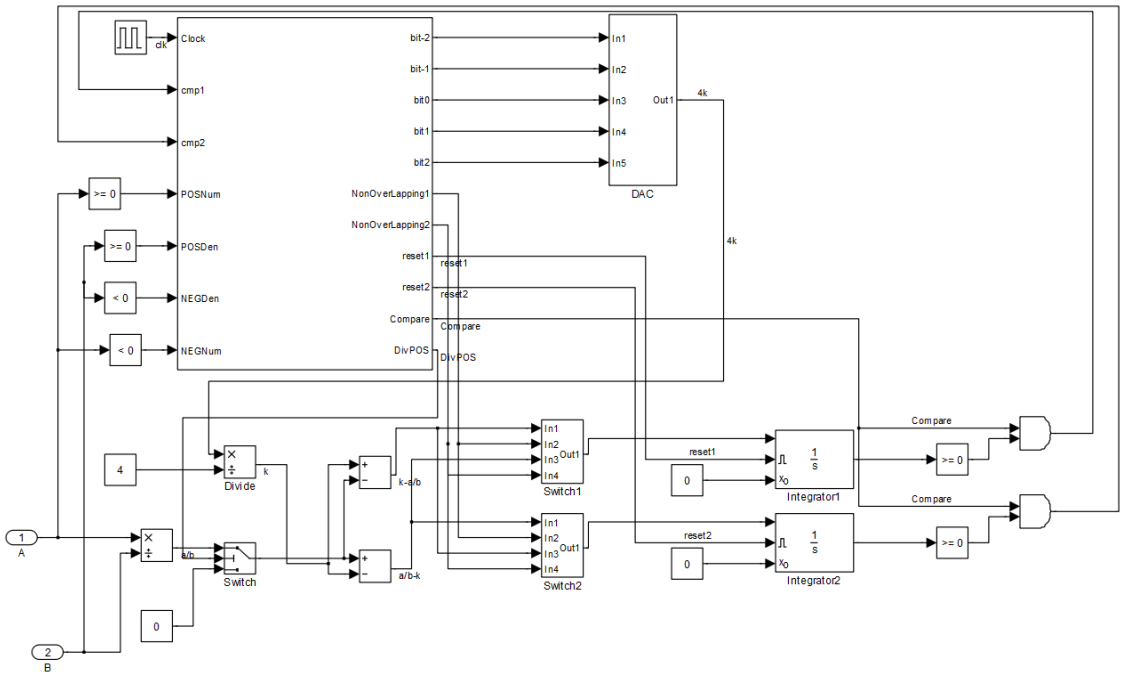



**Figure 44. The Block Diagram of the Design Modeled in Simulink**

## E. Simulation with Synthesized Signals

In order to verify the functionality of the model, we apply five different sets of input signals. In the rest of this section we analyze the result of these 5 sets of inputs.

The first set is two DC voltages which are supplied to the inputs. The Amplitude of one of these signals is five times the amplitude of the other one. Consequently, the predication coefficient should converge to constant value 5. Figure 45 presents results of this simulation. In the first two graphs we present the two signals which are DC values of 5 and 1. Consequently, the instantaneous ratio of the signals is always 5. The fourth graph presents how the prediction coefficient converges to the ratio of the signal, as it can be seen the value stored on integrator

decreases which represents the error of the prediction. Finally the transmitting signal is shown which after a period it is equal to zero.

The next simulation is done using the same concept of constant ratio but with sinusoidal inputs. This time the ratio is equal to 0.5. And similarly it can be seen in Figure 46 that the learning unit converges the prediction coefficient to the correct value to minimize the error on the integrator and consequently information in residue which can be seen in last two graphs of Figure 46.

Following simulation shown in Figure 47 is interesting example. In this case one of the signals is a sinusoidal, but for the other input signal we apply a pulse which represents the polarity of the initial sine wave. This combination of input should result in a value of the prediction coefficient which minimizes the energy, which interestingly is equal to average ratio of a sinusoidal wave, or in other words $(2/\pi)$ or 0.636. Since the ratio of the amplitude of these two signals is equal to 10 the prediction coefficient oscillates around the constant value of 6.36.

Figure 48 shows a real crosstalk example. In this case, the second input is the signal applied in the first input plus another signal, in other words signal one couples a crosstalk noise on the second signal. The learning unit calculates the prediction coefficient to be equal to the value 1. This results in complete removal of crosstalk noise. It can be seen in last graph of Figure 48 that the signal required to be transmitted is a clean sinusoidal wave.

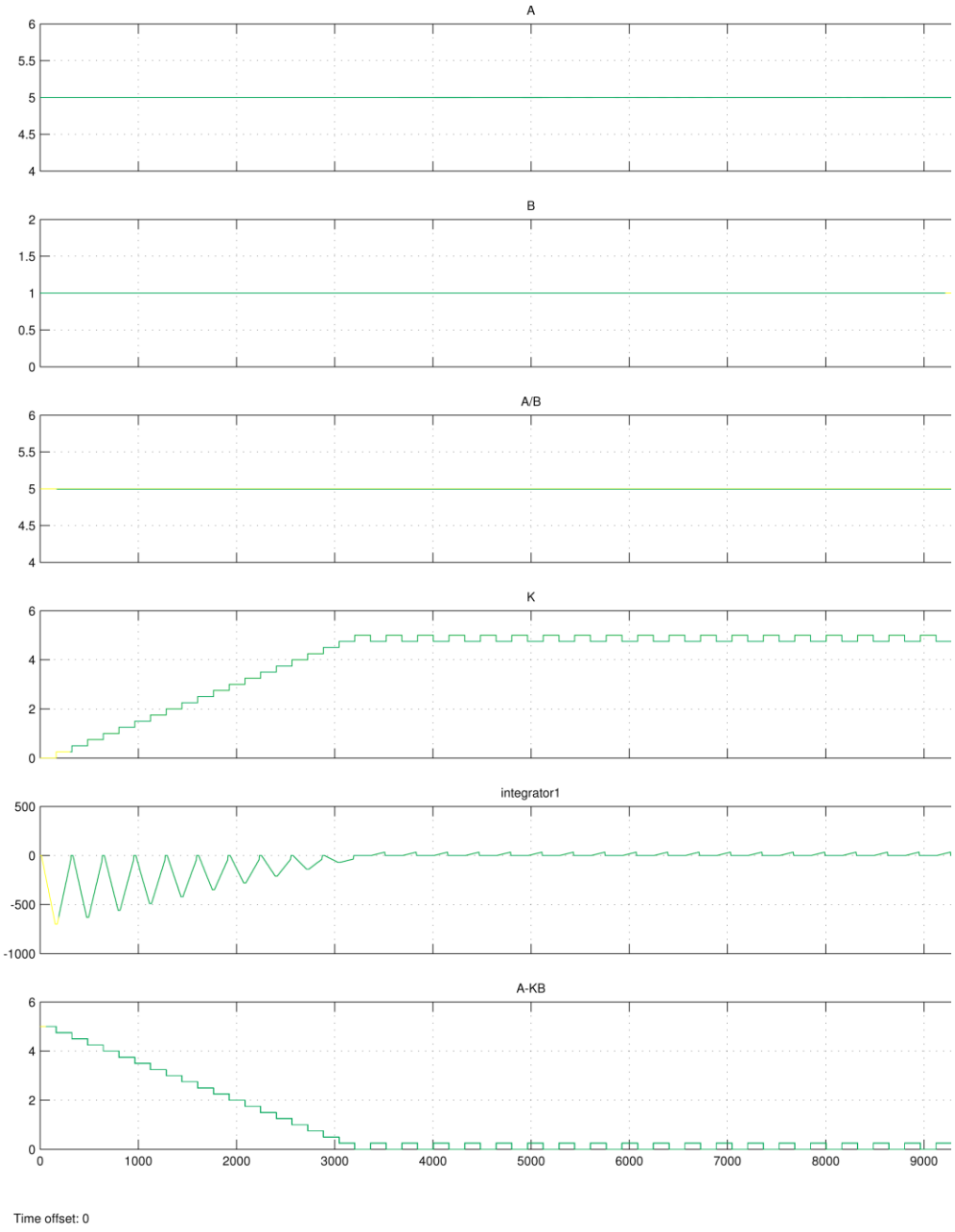



**Figure 45. The result of applying two DC values to the inputs of the Simulink model**
**The top two graphs present the two inputs. The third graph is the instantaneous ratio of the two signals the fourth graph shows how the prediction coefficient converges to five. The sixth graph represents the value stored in one of the integrator which is equivalent to error of estimation; finally the last graph is the residue signal that should be transmitted**

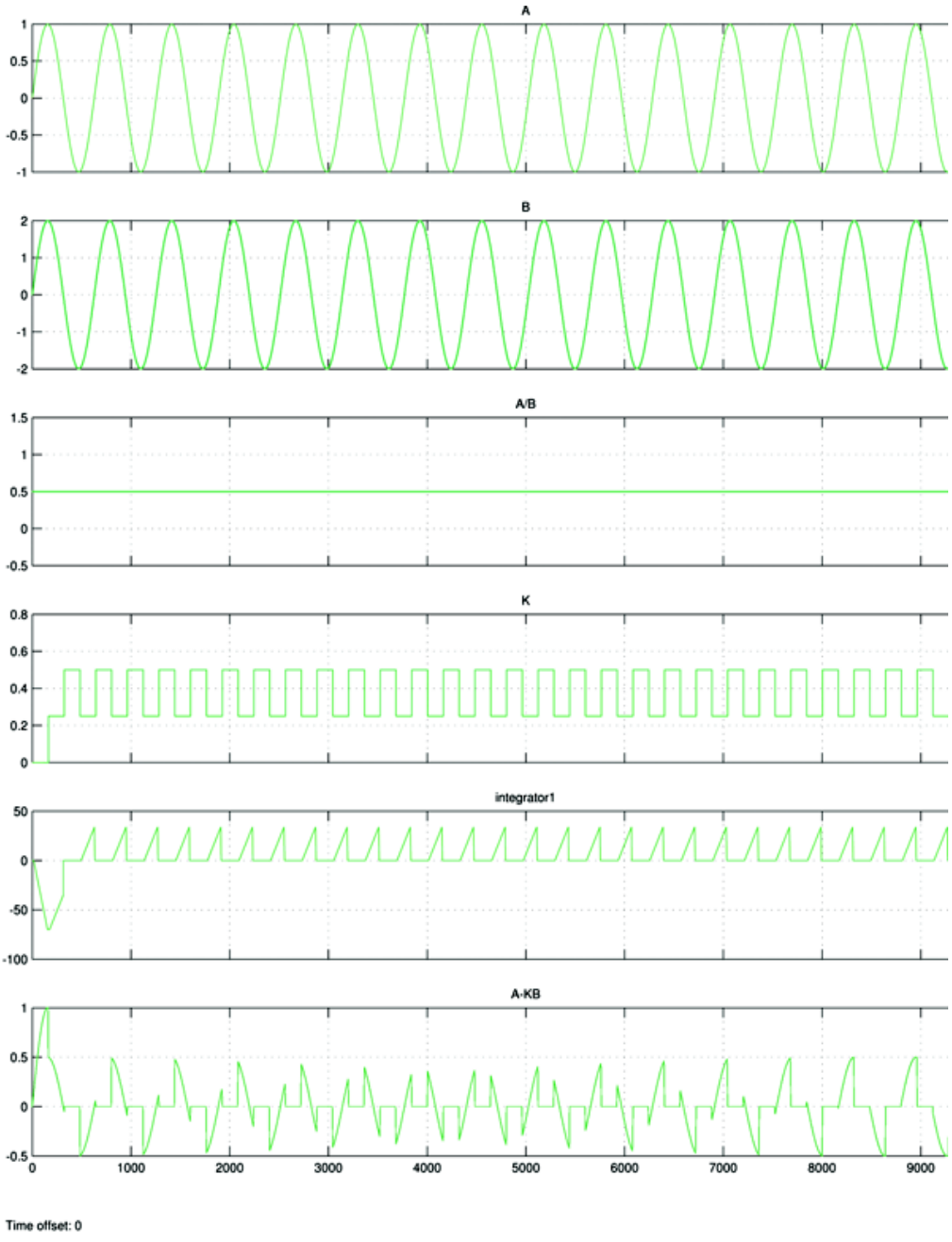



**Figure 46. The result of applying two same phase Sin wave to the inputs of the Simulink model**
**The top two graphs present the two inputs. The third graph is the instantaneous ratio of the two signals the fourth graph shows how the prediction coefficient converges to 0.5. The sixth graph represents the value stored in one of the integrator which is equivalent to error of estimation; finally the last graph is the residue signal that should be transmitted**

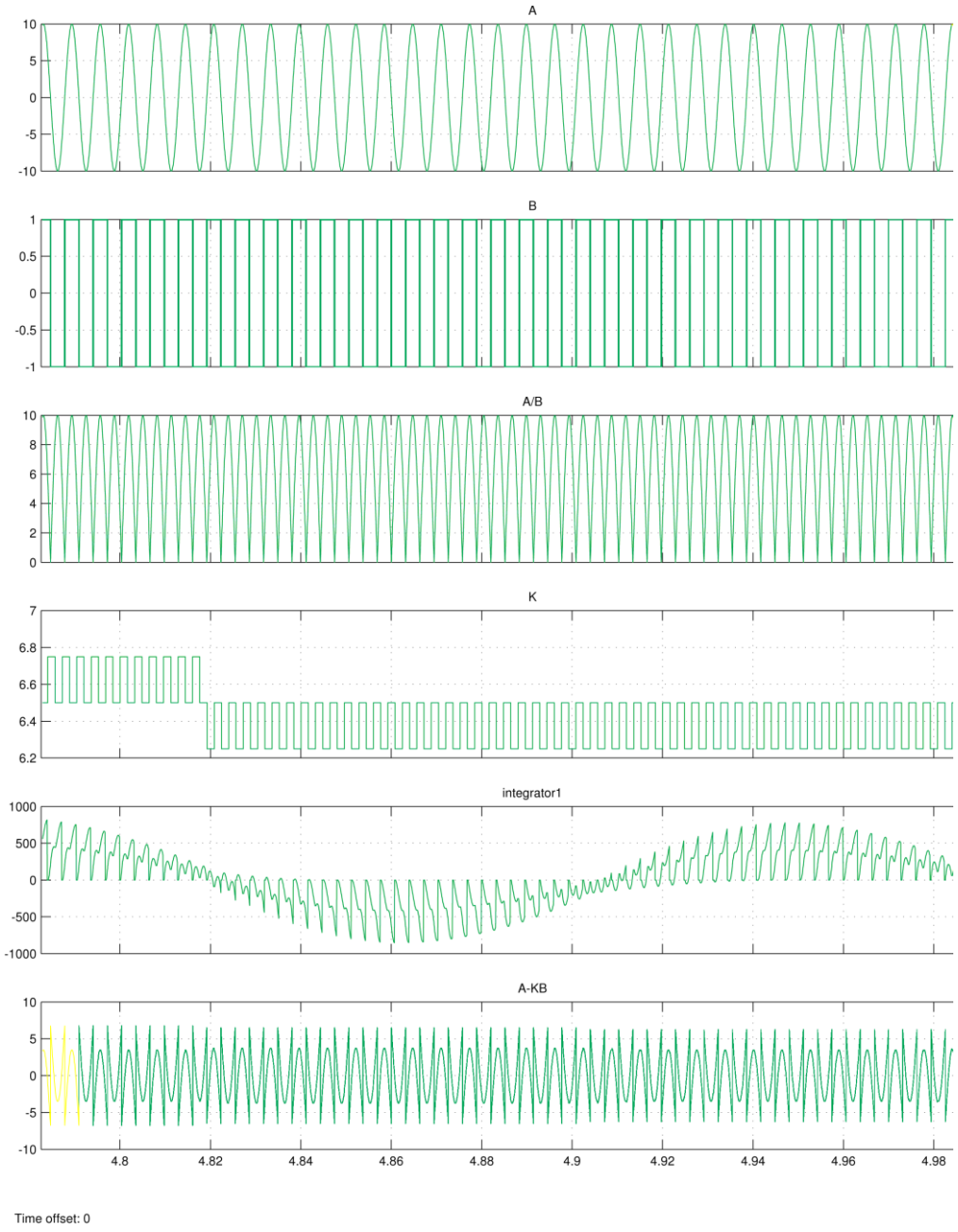



**Figure 47. The result of applying one sine wave and the a signal representing the sign of it to the Simulink model. The top two graphs present the two inputs. The third graph is the instantaneous ratio of the two signals The fourth graph shows how the prediction coefficient converges to Average value. The sixth graph represents the value stored in one of the integrator which is equivalent to error of estimation. Finally the last graph is the residue signal that should be transmitted**

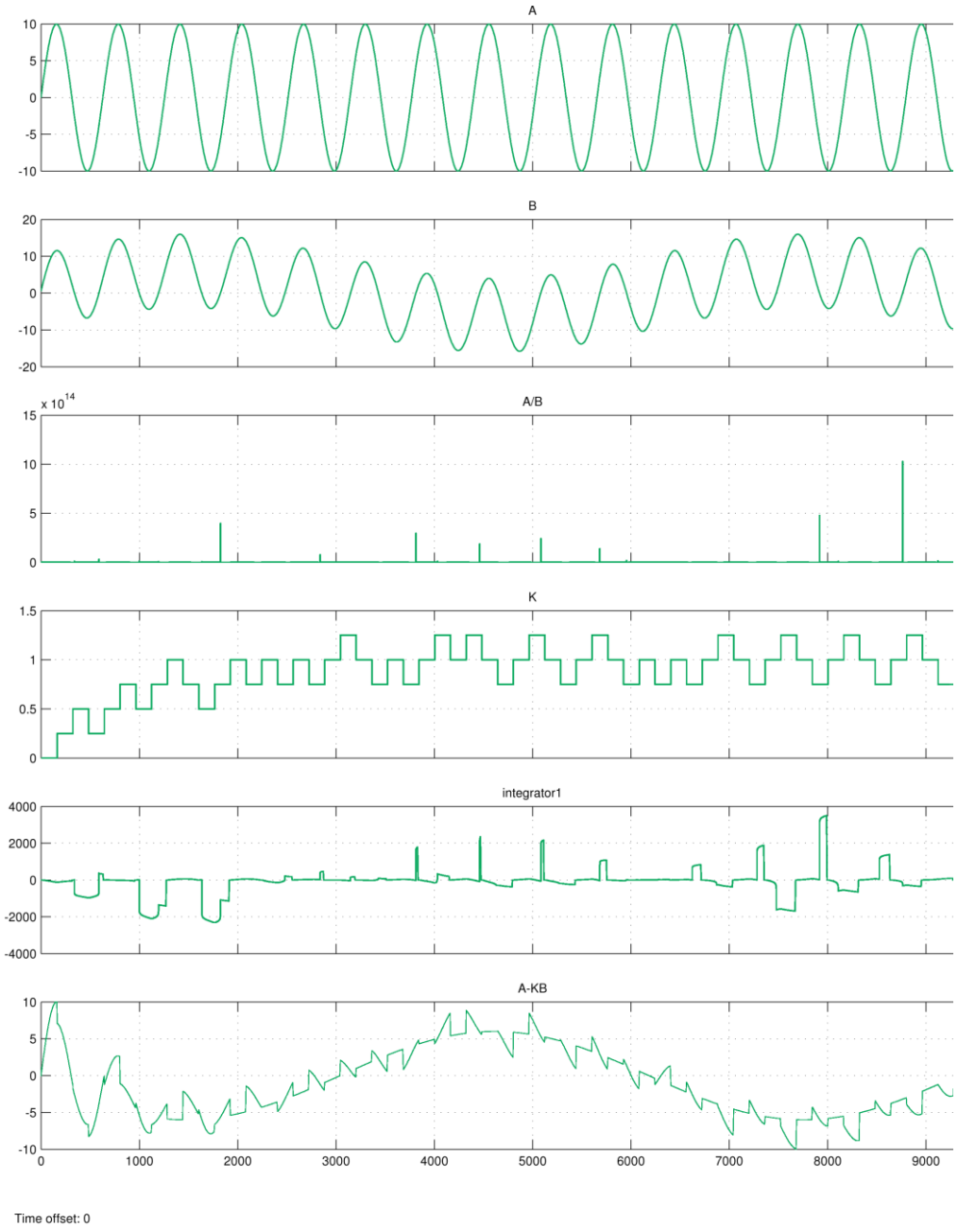



**Figure 48. The results of applying two cross talked input waveforms to the Simulink model. The second waveform is addition of first signal which is a sine wave and another sine wave. The top two graphs present the two inputs. The third graph is the instantaneous ratio of the two signals. The fourth graph shows how the prediction coefficient converges to remove the high frequency sine wave. The sixth graph represents the value stored in one of the integrator which is equivalent to error of estimation. Finally the last graph is the residue signal that should be transmitted which is the low frequency sine wave**

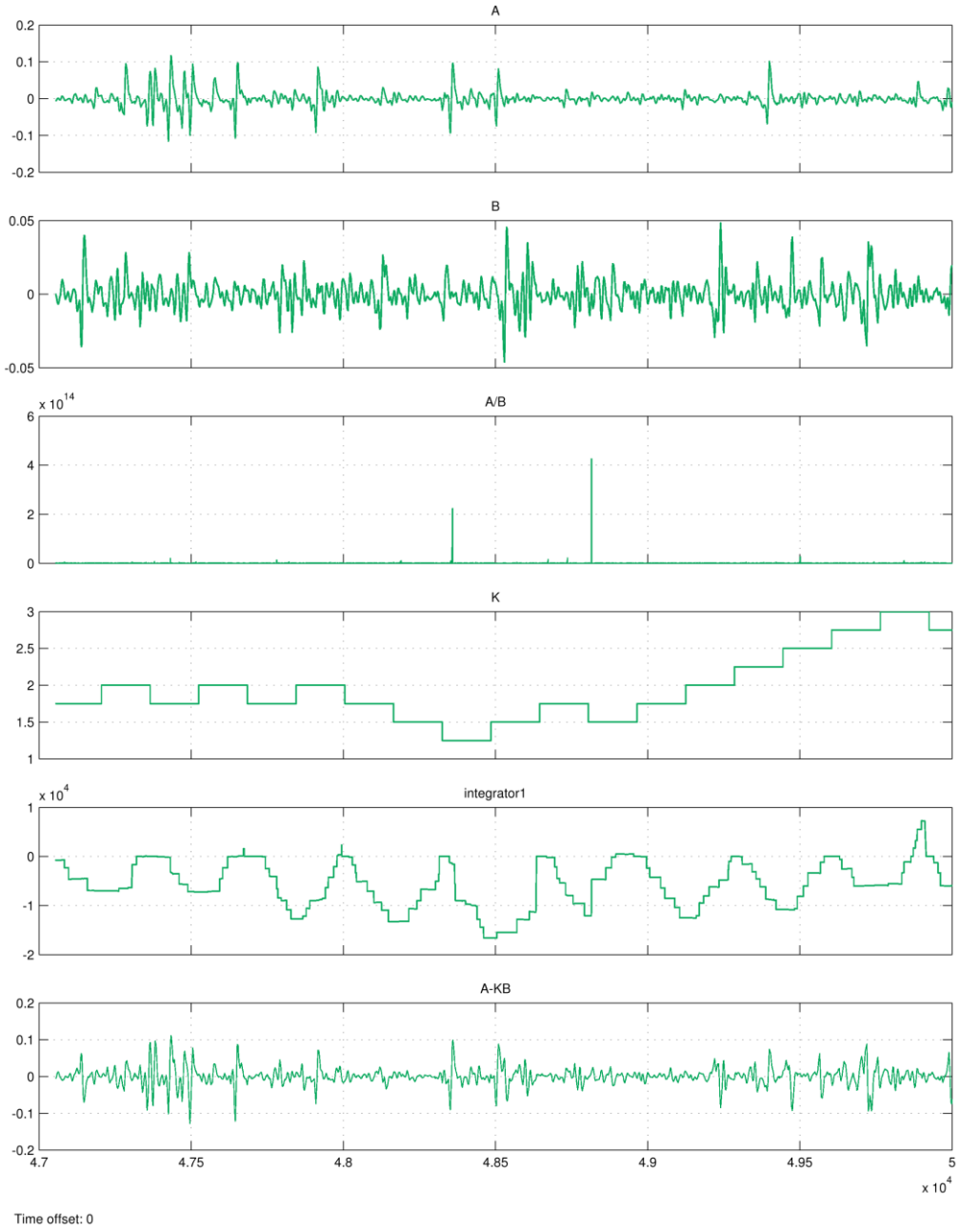



**Figure 49. The result of applying two Synthesized Neural signals to the inputs of the Simulink model The top two graphs present the two inputs. The third graph is the instantaneous ratio of the two signals The fourth graph shows how the prediction coefficient converges to two. The fifth graph represents the value stored in one of the integrator which is equivalent to error of estimation, Final graph is the residue of signal A**

Finally in the Figure 49 we simulated the model with two synthesized neural waveform which was created using the model presented in chapter 2. The learning unit tries to minimize the error by assigning the prediction coefficient to two.

# 5. Prototypes of time-based cross-talk removing algorithm

## A. Breadboard Prototype

For the proof of concept purpose we designed a discrete component version of our design on a breadboard. It was design with a one-to-one correspondence to the Simulink model. For digital section of the design we used TTL chips, for example TTL7400 was used for two input *Nand* gates, 7404 for *Not* gates, 7476 for JK-Flip-flops, and 7410 was used for three input *Nand* gates. Figure 50 shows the complete design. Two sets of two 7segments show the state of the control unit and current state of prediction coefficient. One 08DAC02 is used to convert the coefficient to an analog equivalent signal. To implement each computational part different analog components were used which we will discuss in the rest of this section.

First divider was implemented with an analog multiple/divider from analog devices part number AD534. The configuration which is used in order for the chip works as a divider is shown in Figure 51.

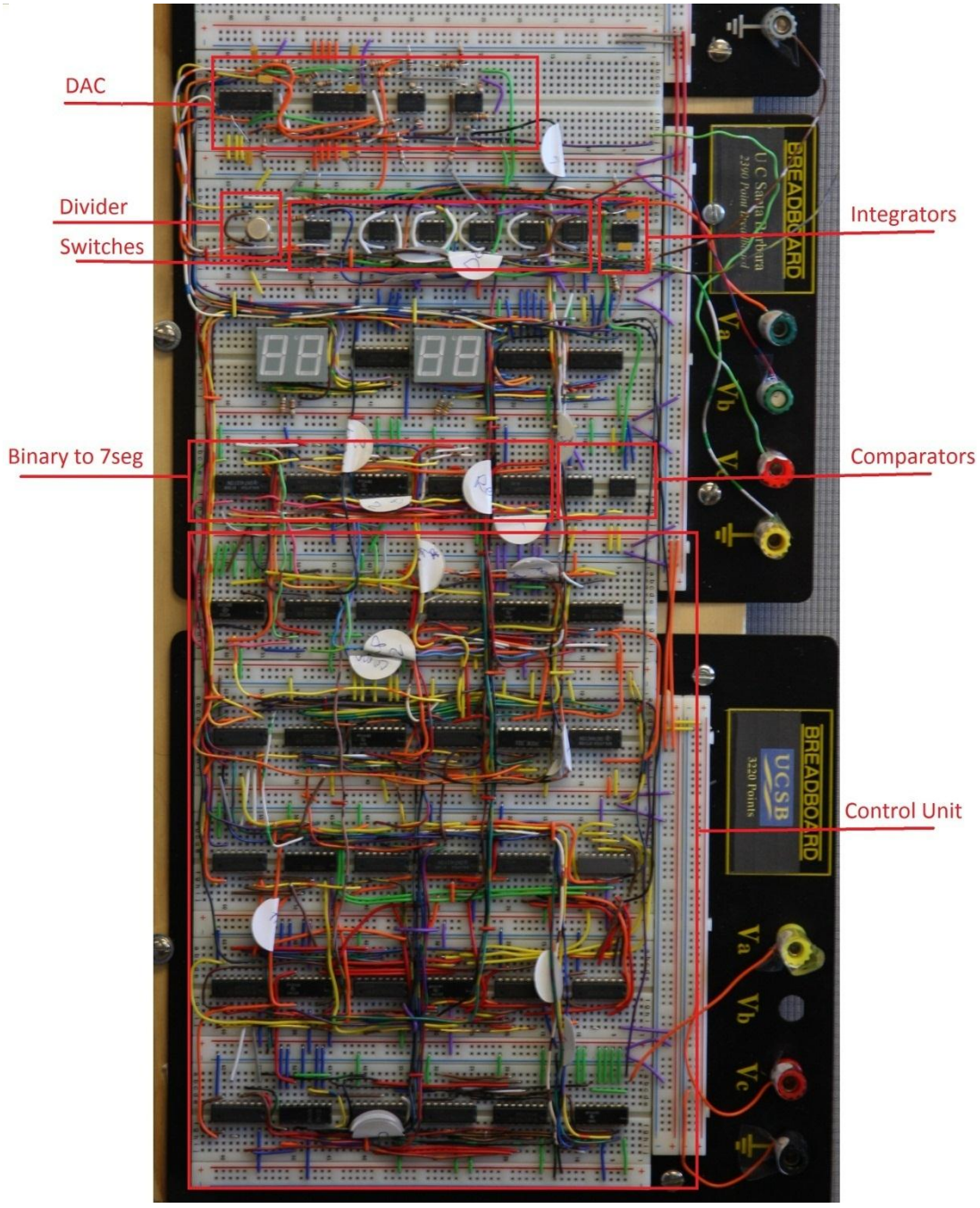



**Figure 50. The image of the Breadboard prototype of the algorithm with each subsection specified**

MAX323CPA is an analog switch used to route the signal through the breadboard. This chip consists of two analog switches which work in the range of -5

to 5 volts. The switch enable is TTL compatible, so using signals created from control unit the switches select correct input for the integrators. AD8561 is the analog comparator used in our implementation.

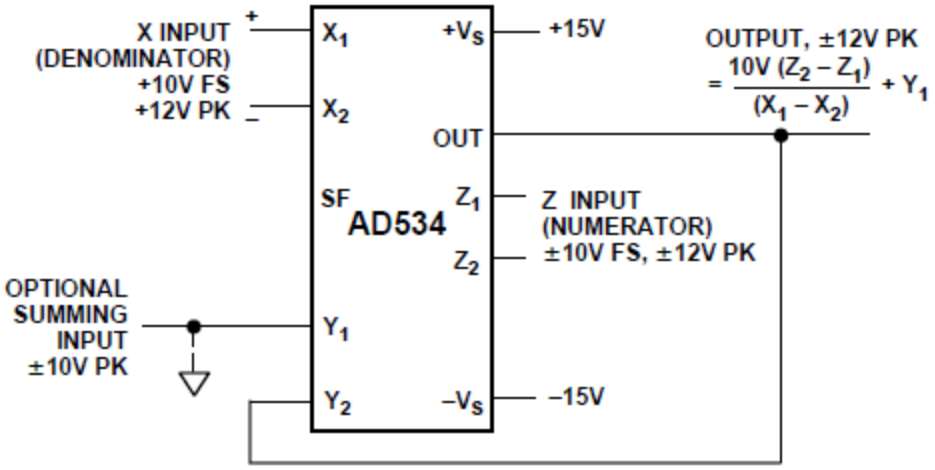



**Figure 51. Configuration for AD534 to work as a divider source: http://www.analog.com**

LM1458 is the operational amplifier which used for different purposes, first it was used as an adder/subtractor. The block diagram of implementation of adder with Op-amp is shown in Figure 52. The output signal equals to $-R4(\frac{V_1}{R1}+\frac{V_2}{R2}+\frac{V_3}{R3})$

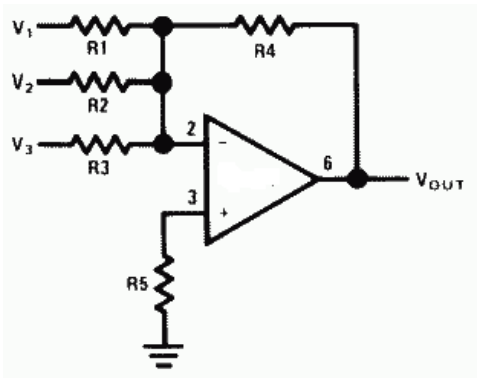



**Figure 52. The block diagram of implementing weighted adder using an op-amp**

Another use of the LM1458 is in implementing the integrator of the system, Figure 53 shows the block diagram of integrator implemented using Op-amp.

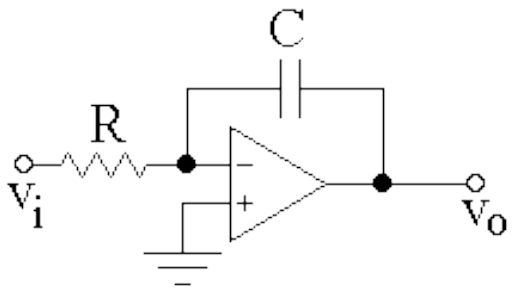
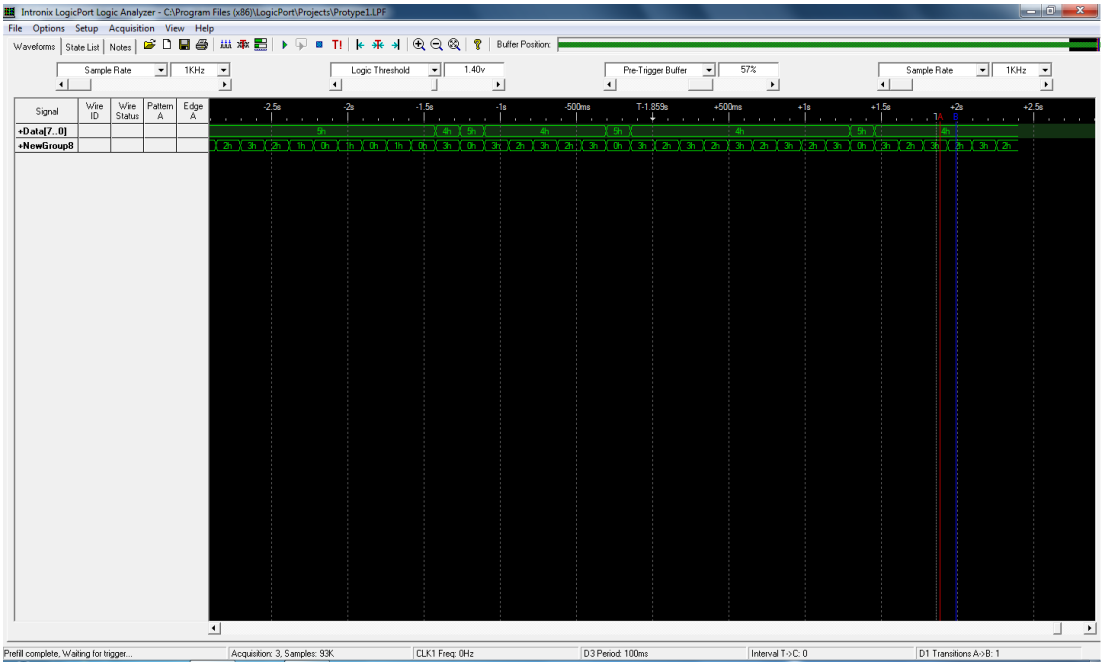



**Figure 53. The block diagram of implementing integrator using an Op-amp**

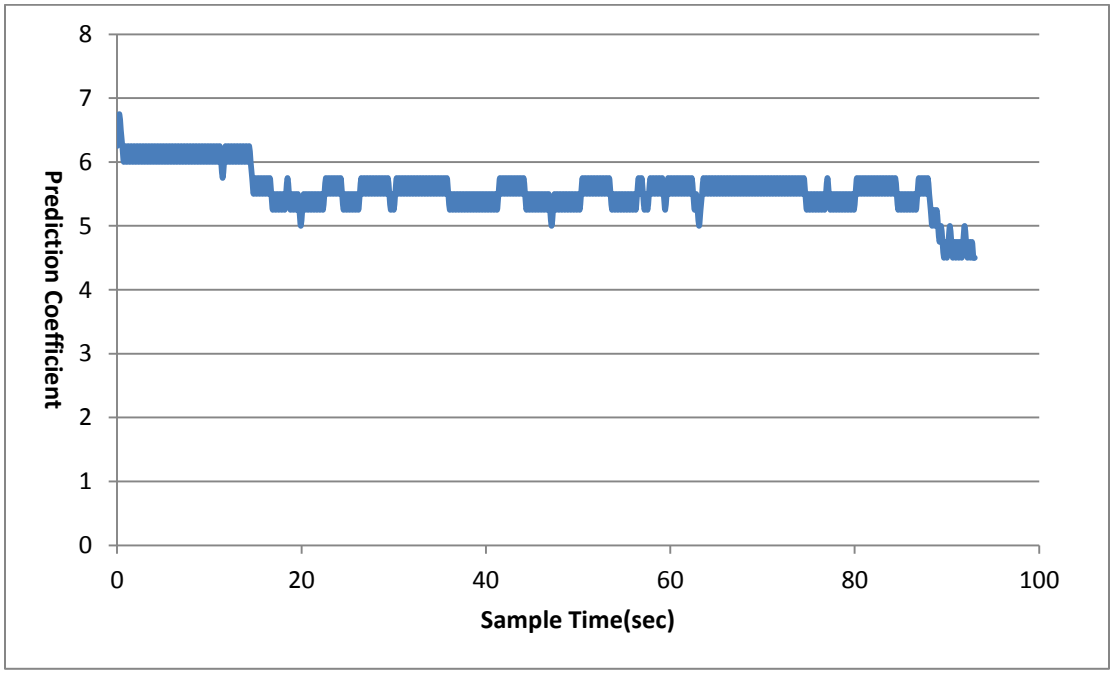
The prototype was tested with DC and sinusoidal input tests shown in previous section and passed the test, the high level on noise in breadboard prevents the design to be tested with very small amplitude signals. So in the next section we implemented the design using low noise PCB board.

## B. PCB Board Prototype

After testing the breadboard prototype a PCB version of the design is developed and tested. Figure 55 shows the image of the PCB board implementation. First we tested the board with DC and sinusoidal inputs as it is shown in Figure 54. Then, we developed a testbench to verify the design using the synthesized neural signals. For this purpose we used a SCB-68 LabVIEW board to convert our MATLAB signals to equivalent analog output. Figure 56 shows the block diagram of the program used to feed the synthesized neural signal to the PCB Board prototype.




**(a)**




**(b)**

**Figure 54. Result of applying the Sinosoidal inputs to the PCB board prototype. The ratio of amplitude of waveform changed in t = 15s and t = 90 from 6.4 to 5.5 and from 5.5 to 4.8 correspondingly. Part a shows an snapshot of the result gathered through logic analyzer. And Part b shows the final diagram shown in Excel diagram**

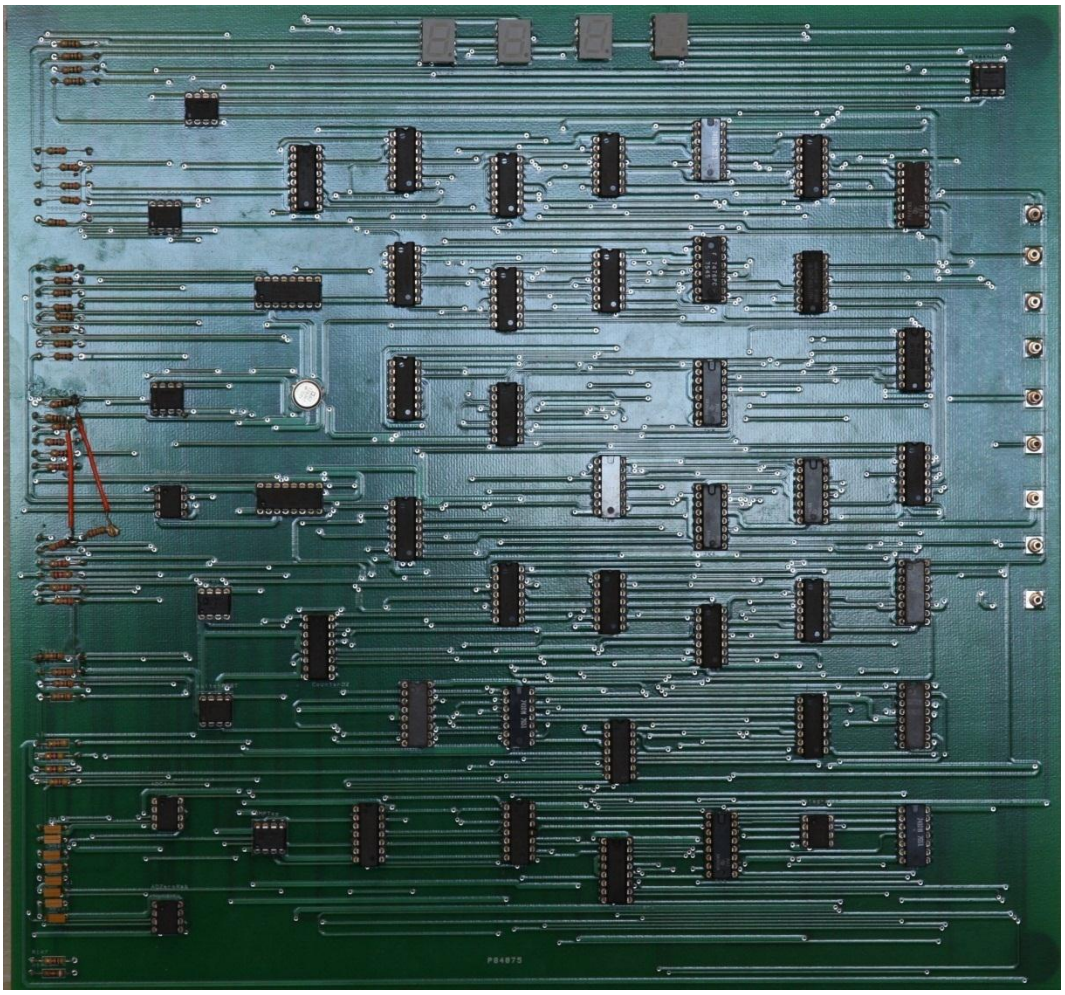


**Figure 55. The PCB Board prototype of the algorithm**

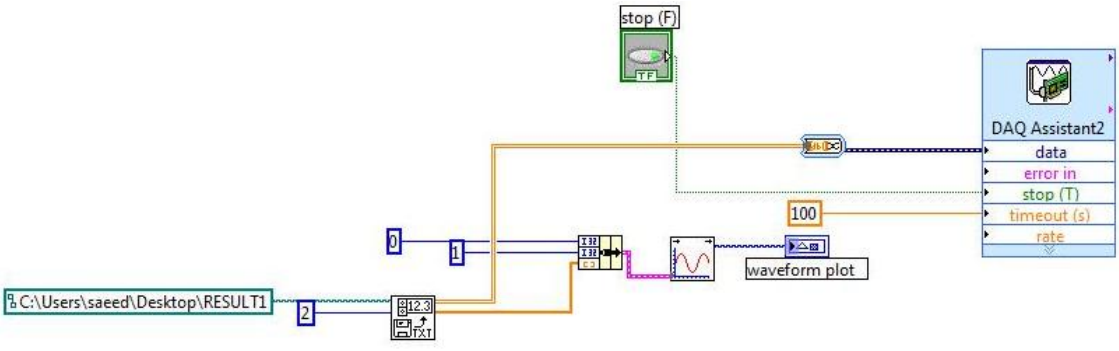



**Figure 56. Block Diagram of the program for feeding the synthesized neural signals to the PCB prototype**

The neural activity is fed through LabVIEW board to the PCB prototype, the limited bandwidth of the LabVIEW board, pushes us to decrease the sample rate and frequency of the clock accordingly. So, the ratio of two signals stays unchanged. An image of the whole testbench is shown in Figure 57.

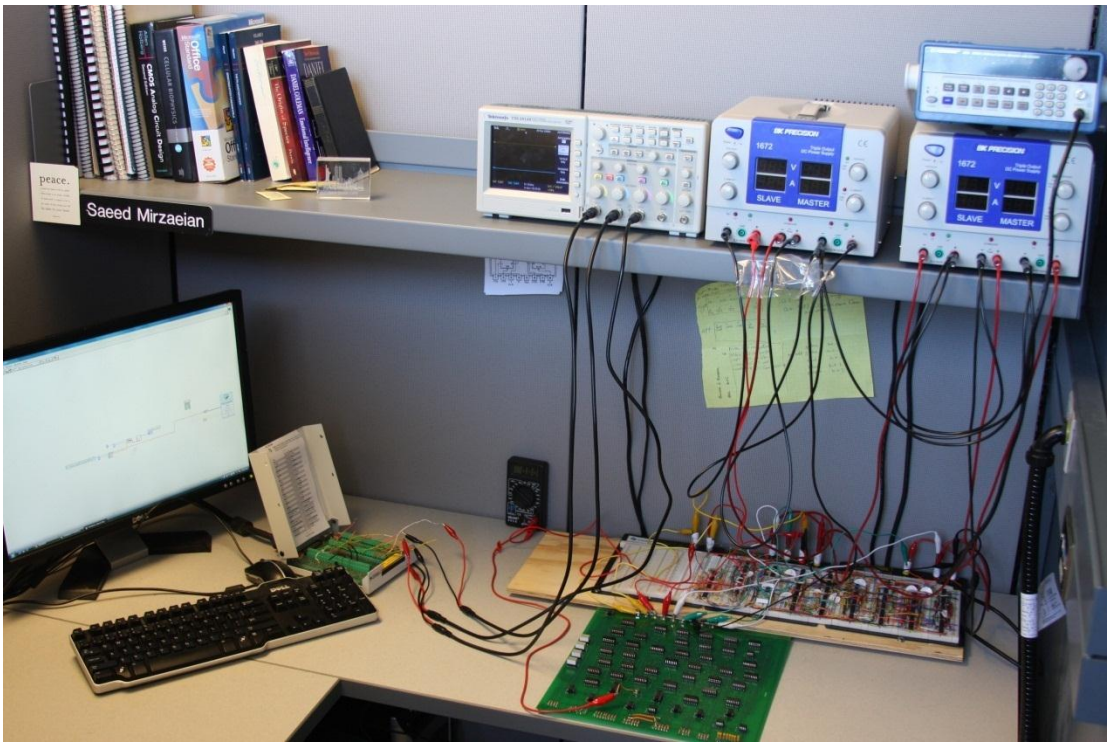



**Figure 57. The test setup for testing the PCB Board prototype with synthesized neural signals**

The test signal was divided into 10 equal time period after each time period the learned prediction coefficient is read and used to compress the next time period of the test. Table 5 shows the resulting values. The entropy of the signal is reduced from 10.92 to 8.49 which is equivalent of 2.5 bits of saving in the system.

**Table 5. The resulting prediction coefficient when neural signals are fed to the PCB board prototype**

| Time Period | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Prediction Coefficient | 0 | 6.5 | 1.75 | 0 | 0 | 5.25 | 4 | 0.5 | 1 | 3.75 |

## 6. Four neighbors module

This chapter was based on the crosstalk of only two signals. But the same concept can be extended to more than two signals. In our design we estimate each signal with the three neighboring signals on the left and top of the signal. To clarify this concept, we consider the sixteen neighboring probes in Figure 58. And the signal transmitted for signal A is shown with the symbol A'. Now the transmitting signal from probe F (F') can be showed as: $F' = F - K_1A' - K_2B' - K_3E'$.

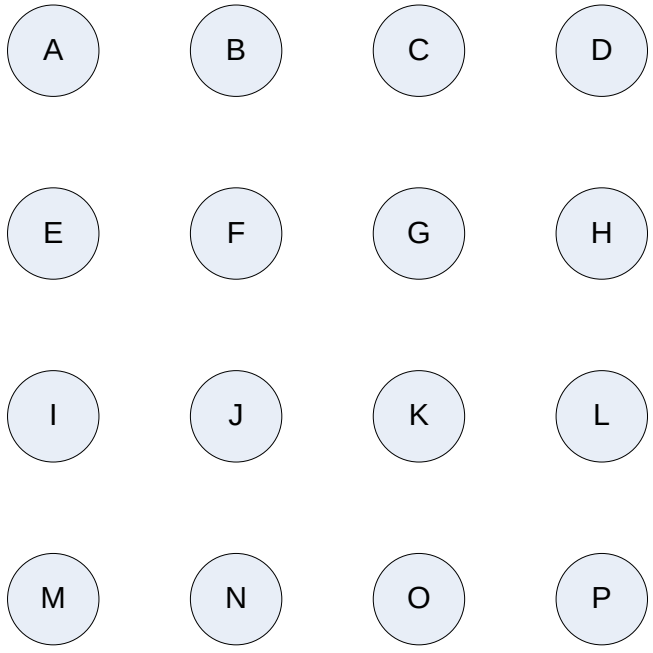



**Figure 58. Block Diagram of sixteen neighboring probes**

Similarly for K (K') we have following equation:

$$K' = K - K_4F' - K_5G' - K_6J'$$

For the corner cases, we use the following concept; the probe A is transmitted by its own, B is only estimated by A'. E is estimated by A'. And finally probe F and every fully surrounded probe. The concept is shown in Figure 59.

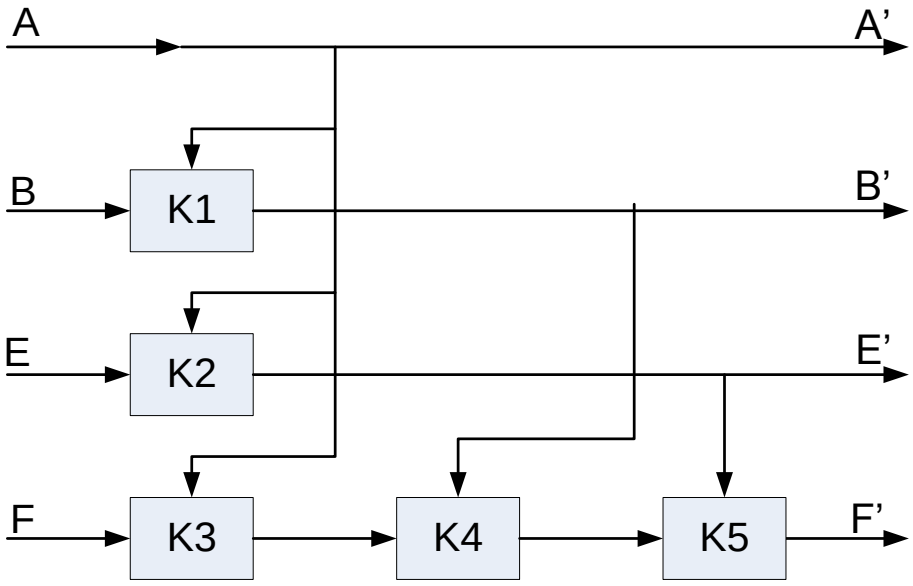



**Figure 59. The order which the corner blocks are calculated**

## 7. Conclusion

In this chapter we first gave an overview of information theory and signal compression methods. Then, we examine our synthesized neural signal with the concepts defined on the first section. As a result we found a high source of redundancy in the neighboring signals. Consequently, we introduced a hardware

friendly novel algorithm to remove the redundancy in neighboring probes which is the result of coupling of the same neurons on them. Then the design was modeled in Simulink and tested. Thus, two different prototype of the algorithm was implemented using discrete electrical elements. Finally, we extended the algorithm to a broader perspective.

# IV. Compact, Low Power and Low frequency Analog Divider, Comparator, and Integrator for Neural Signal

This Chapter presents a series of computational circuits that can be used to sense low frequency, and noisy neural signals. The computational blocks are designed using current-mode approach based analog VLSI CMOS technology operating in the subthreshold regime. These computational blocks were implemented and tested in IBM 0.13um technology and will be used as building blocks for implantable neural signal recorder.

## 1. Introduction

In-vivo devices have stringent requirements which prevent the use of computationally intensive signal processing algorithms in these environments. The first limitation is the power, commonly used signal processing methods in computer network or image processing deals with a huge amount of data, but can only be implemented using power consuming processors. Data in these domains are usually are of high fidelity. The information extracted from bio-environments, on the other hand are primarily in the low frequency domain and more noisy. This aspect of bio-signals lends itself to more lossy processing units compared to critical information involved in computer networks and communications. Another limitation in biosignal is the area of the processing units; in normal computational environment the systems

are immobile and can occupy almost a whole room, but in bio-signal cases, the mobility of subjects limits the size of devices that can be used for signal extraction, processing and transmission. Due to these special criteria of bio-signals, there is a trend towards using low power lossy analog computational methods. Analog computation, however, lacks some of the requirements needed in normal biosignal processing units. First, the low frequency of biosignals results in the need of rather large capacitors in order to create the large time constants needed for low frequency computational circuits. Other deficiencies in analog circuits are lack of reliable memory elements and dependency on fabrication process variation.

In this chapter we introduce many novel computational elements useful for in-vivo biosignal processing. Main computational elements we introduce are the integrator, comparator, multiplier, divider, and V-I converter. In next section of this chapter we introduce the design of these blocks and their analysis. In section 3 we present our implementation of these computational blocks in IBM 0.13µm and conclude our paper in section 4.

## 2. Blocks

In this section we present each of the blocks that were designed. First using two small Miller-capacitors we introduce a novel compact and low frequency integrator. The next component is a process-robust comparator; this component uses auto-zeroing for compensation of any process related mismatch during fabrication, the result of this compensation is stored as a differential value in two small

capacitors. In the next section we introduce our divider which comprises of two common analog dividers. Since this divider works only in first quadrant, we implement a rectifier which modifies the inputs in order for the divider to be used in all four quadrants. We start with a simple Winner-Takes-All and use it as a comparator to design our very accurate rectifier. In the next subsection a very minimalistic multiplier is introduced. Finally we present our linear differential current to voltage converter.

### A. Integrator:

Common analog integrators are designs for high frequency data, but in the case of neural recording the range of frequency is from 5 Hz to 10 kHz this low frequency range results in a need of integrators with huge capacitors (Shutao, Yaonan and Jie). In our design we used a miller capacitor which connects two stages of gain in a three stage amplifier. As a result the size of capacitor is multiplied by the gain of two stages; consequently smaller capacitor will result in lower frequency pole. Compensation of this integrator should be done between every two stage of gain.

Figure 60 shows the block diagram of the devised integrator, capacitors C2 and C3 work as Miller capacitors, C2 is between two gain stages so it will act as a larger capacitor. The values are these capacitors are much smaller than previous designs with maximum of 800fF. C4 is just a compensation capacitor and C1 is used for tweaking the location of the high frequency pole. Figure 61 shows the calculated bode diagram of the integrator the main pole in this case sits at 1mHz.

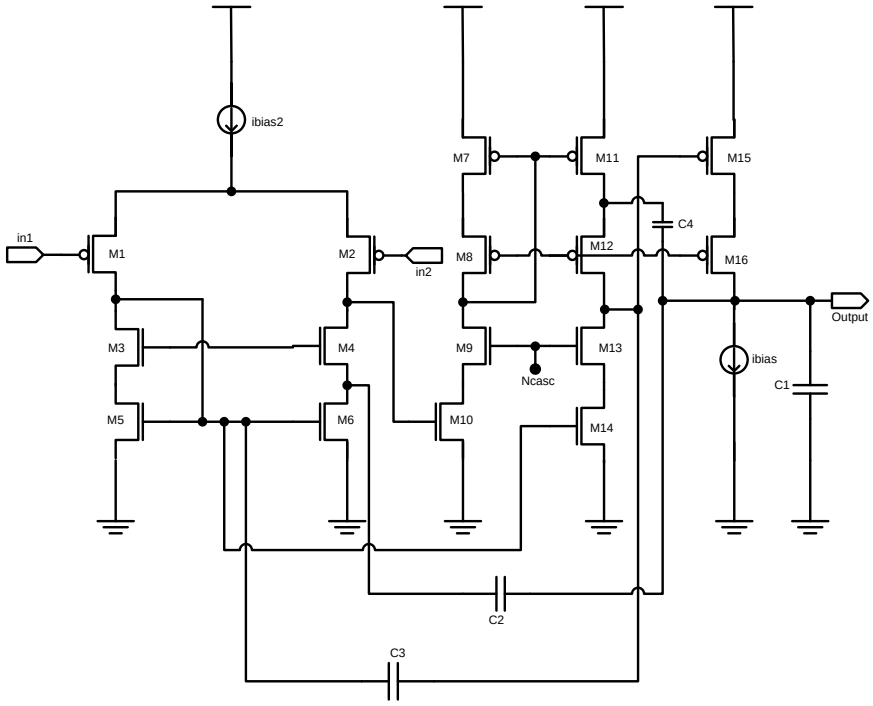



**Figure 60.Basic structure of low frequency integrator**

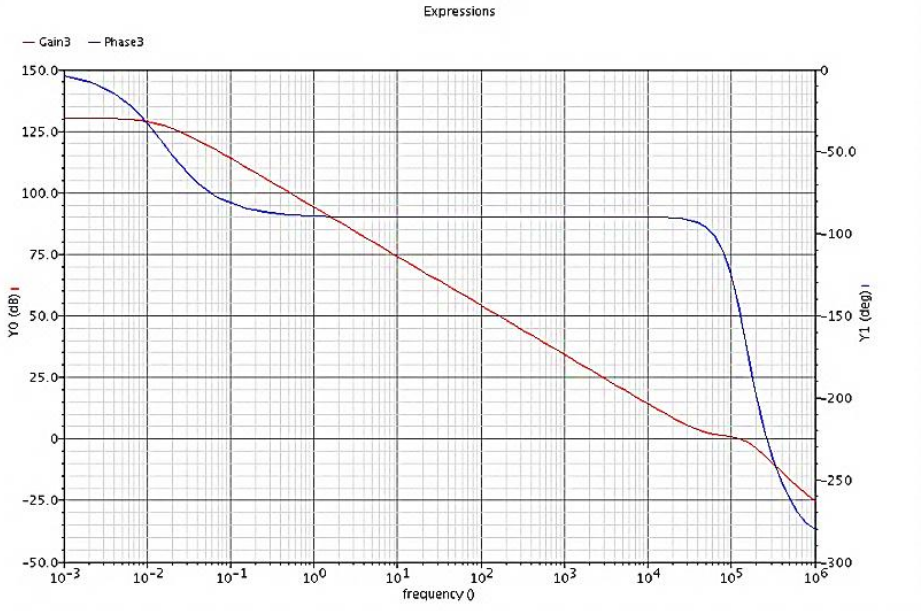



**Figure 61. Bode diagram for our designed integrator**

## B. Comparator:

One of the issues encountered in analog computation in finer scale technologies, especially for subthreshold circuits, is process variation; this might even affect transistors laid out using good matching techniques. In the case of very accurate comparison, the comparator should be periodically calibrated to remove the offset. Commonly two methods are used for auto zeroing the comparator either time domain (Enz and Temes) or frequency domain (chopping) (Stefanou and Sonkusale). In (Yu and Geiger) a digital method for programming the bias current is presented for offset removal.

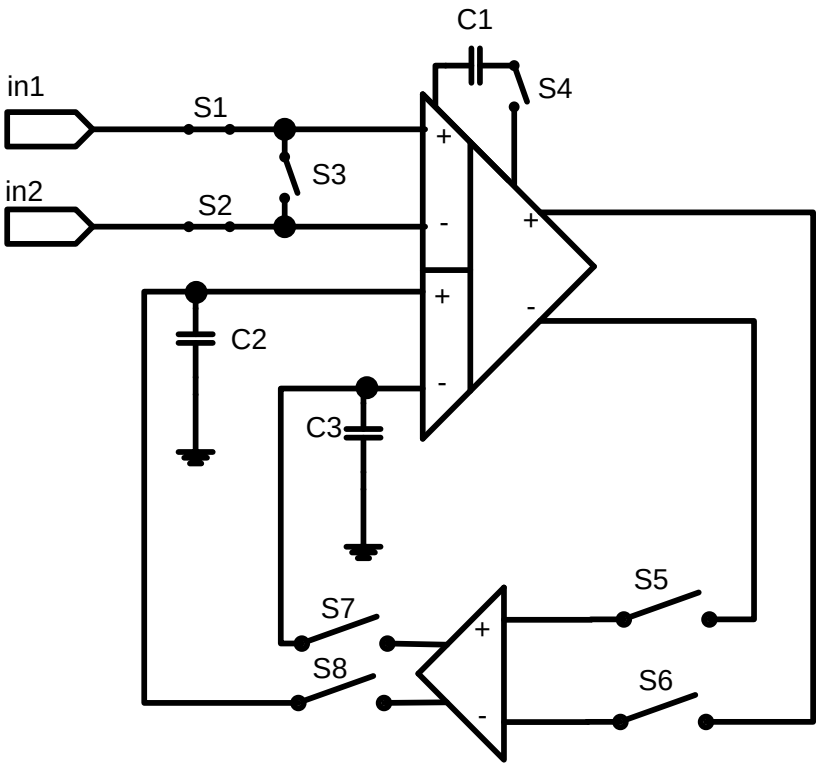



**Figure 62. Basic architecture of the comparator auto-zeroing circuit**

In our design we use time based auto zeroing. In this method an offset voltage is added to the input of the comparator. The purpose of this offset input is to give the correct output if two inputs are equal. As we discussed earlier storing this value needs an analog memory which is not reliable. In order to overcome this two different solutions are proposed. First the offset cancellation will be done every clock cycle; consequently the memory element should keep the offset value for one clock cycle. Second the value is stored as a differential input. Since symmetrical capacitors are most likely to lose their charge with the same rate, using two capacitors instead of one will prevent the loss of information stored on them as shown as C2 and C3 in Figure 62.

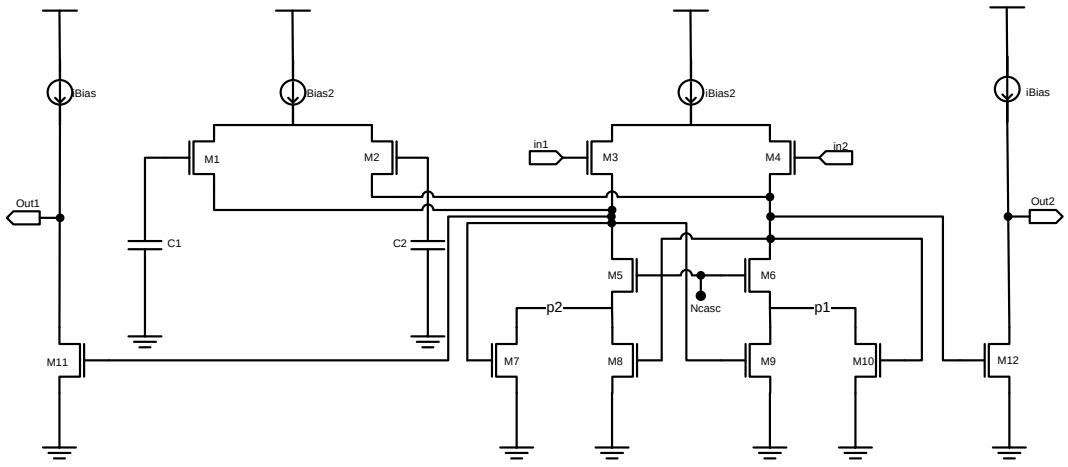



**Figure 63. Circuit diagram of comparator**

The comparator shown in Figure 63 has two differential inputs, one for the values and one for offsets. In offset cancellation phase, the comparator is transformed to an amplifier by adding compensation capacitors (C1 in Figure 62) and it is inserted

into the feedback loop with a very low gain amplifier shown in Figure 64. This amplifier imposes the constraint of equal outputs to the original comparator. These modifications will remove the need of huge capacitors for storing the offset value.

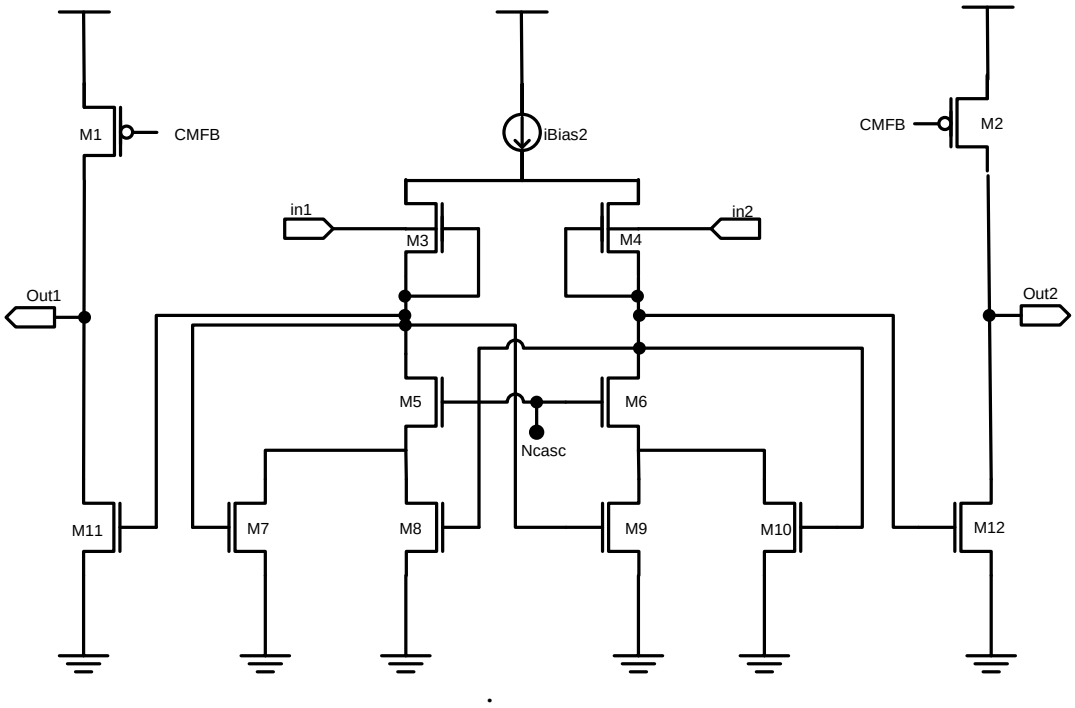



**Figure 64. Circuit diagram of a low gain amplifier**

The reason for putting a low gain amplifier in the feedback is that at the beginning of offset cancellation phase the outputs of comparator are sitting in Vdd and GND. Therefore, this large input would result in huge swing in the output of feedback amplifier and the system would not result in a stable mode. Figure 65 shows the simulation output of comparator in when two sine waves are the inputs, as can be seen in the midsection the output is calibrated, if the feedback amplifier had a big gain instead then the result would have been a rail-rail oscillation.

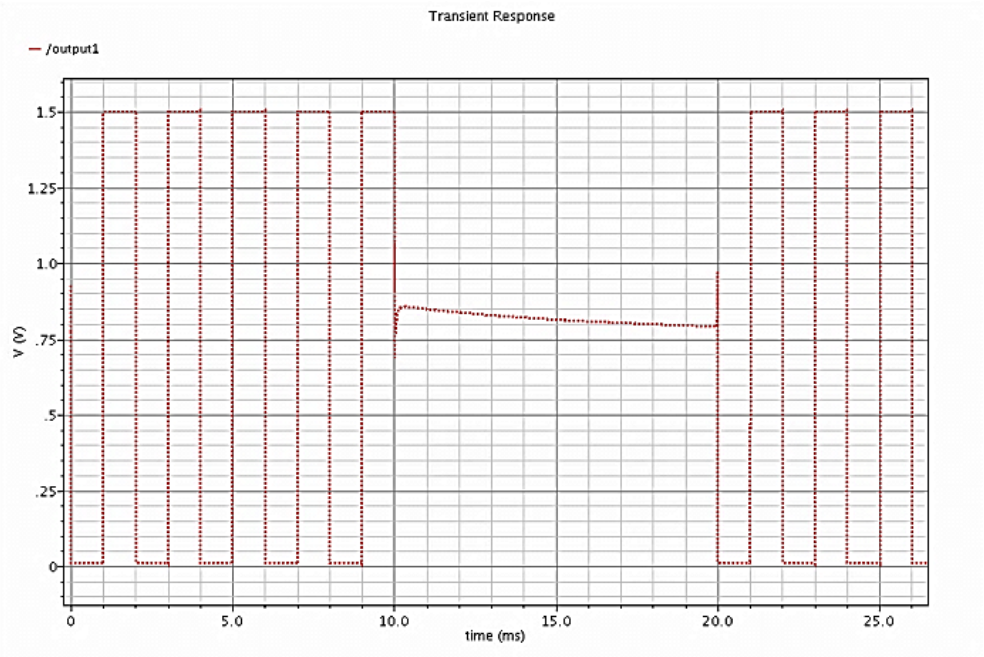



**Figure 65. . Simulation result of comparator**

## C. Divider:

Sub threshold biasing of a transistor offers us exponential function, having this property transforms the complicated operations such as multiplication and division to simple addition and subtraction respectively. This is one of main reasons for migrating from digital domain to analog design; in digital in vivo signal processing, millions of multiplications per second are needed for the simple step such as Fourier Transform. Each of these multipliers would require huge digital circuitry (Oweiss, Anderson and Papaefthymiou, Optimizing Signal Coding in Neural Interface System-on-a-Chip Modules). In our design we employ two different current-mode designs for a divider.

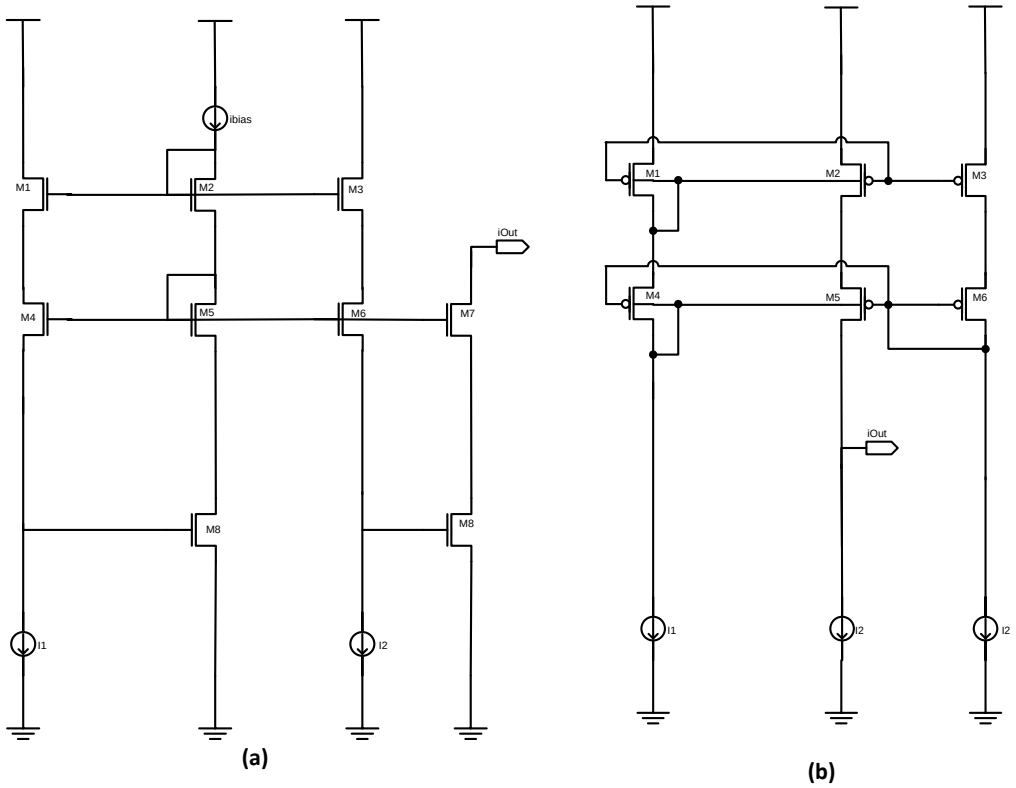
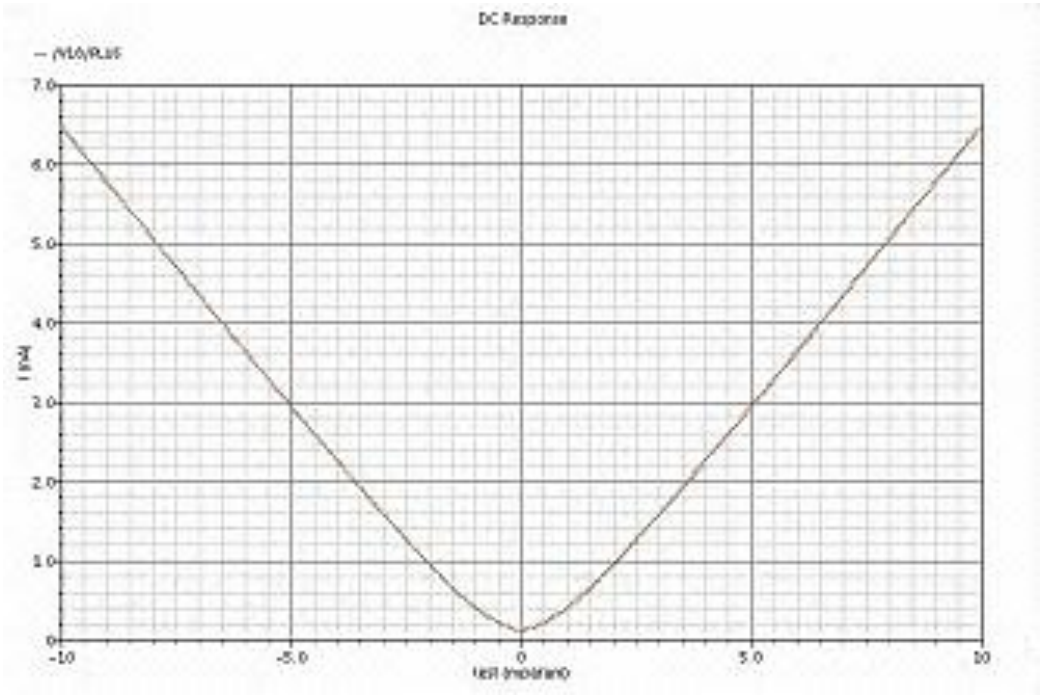
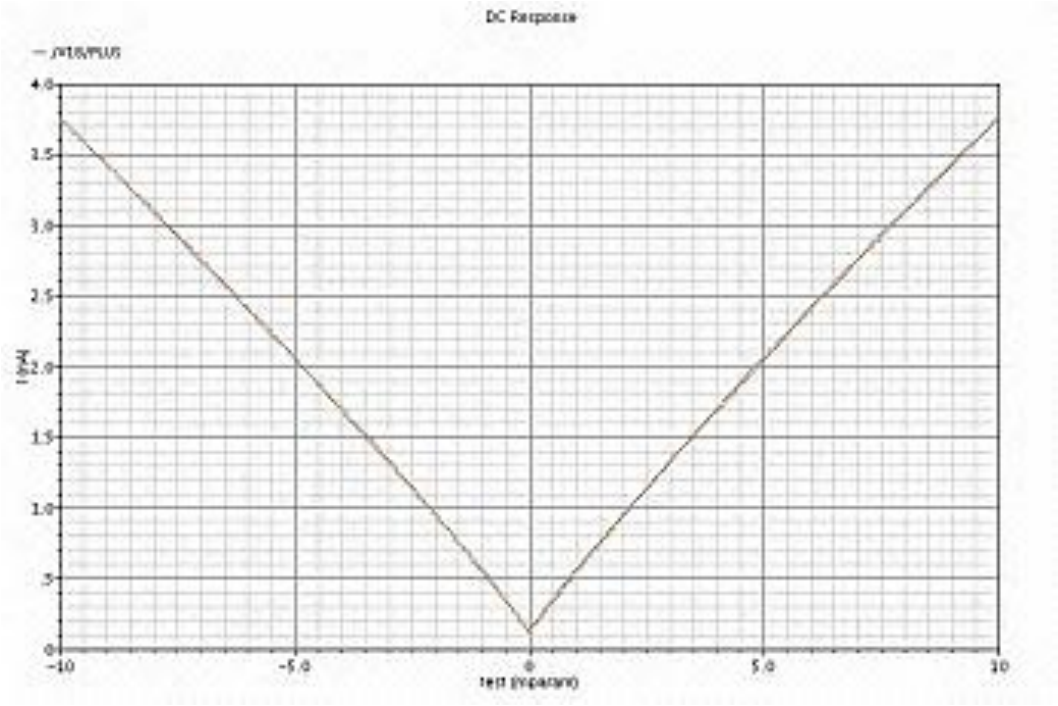



**Figure 66. Two different analog divider used in or design (a) uses normal subthreshold design for implementing division (b) backgate voltage is used to implement division**

Figure 66 represents two possible methods of implementing the divider, the first divider introduced in (Wilamowski) uses sub-threshold conduction mode of transistor in order to implement logarithmic subtraction or division of two currents. Figure 67(a) shows the result of a simple division on this design. It can be seen that the divider loses its accuracy when the input range is small. To rectify this we include another divider.




(a)




(b)

**Figure 67. Result of each of two implementation of divider represented in Figure 37**

Figure 66(b) represents the divider implementation which uses the differential property of gate voltage and bulk voltage has on the current of transistor in order to do the division. Figure 67(b) presents the output of this divider for the same division. This divider, contrary to the earlier one, performs better for small range input but in larger values this design saturates. Combining these two designs we introduced a divider that has the accuracy of the first design for larger inputs and that of the second design for smaller inputs. In this design the output of two designs are multiplied by a constant and added to each other, the constant used is process related and in our specific design we used the ratio of one to five. Figure 68 shows the response of the final design for the same DC sweep of input; the final output is accurate in large input range.

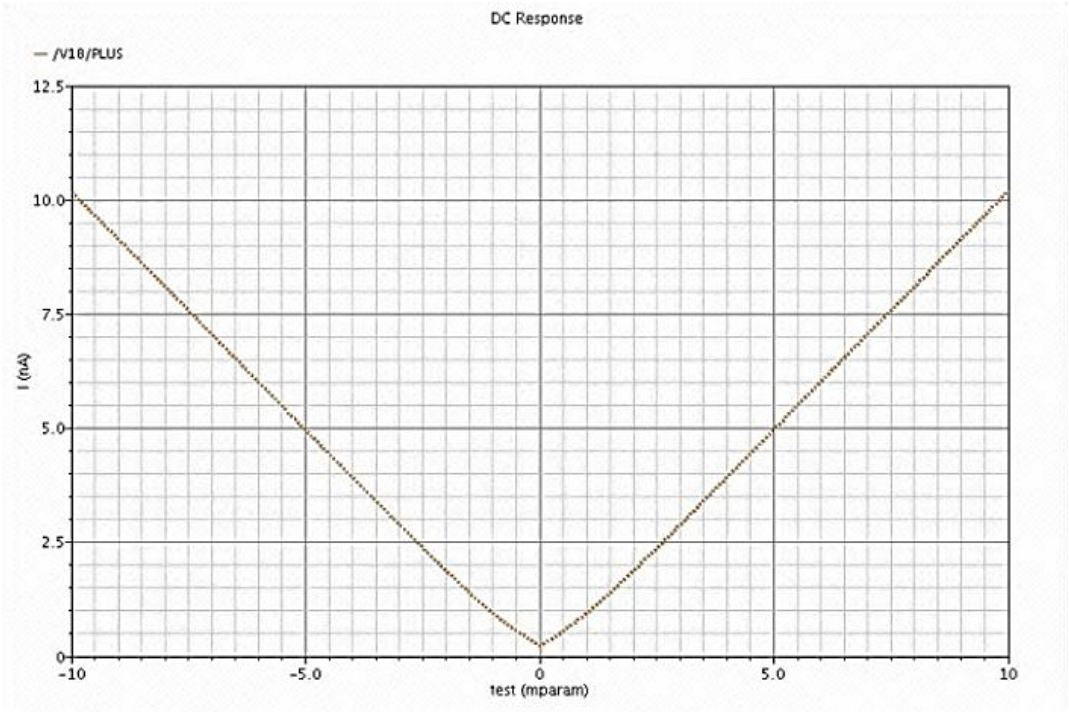



**Figure 68. The output of final divider for dc sweep input of -10mV to 10mV**

As mentioned earlier these dividers work only in one quadrant of the input. In order to extend our divider for all four quadrants, we devised a simple comparator and rectifier in order to modify the inputs to positive value. The final result of the division would be negated if the divider and dividend have different signs.

## D. Winner Take-all

The basic Winner Take-all (WTA) circuit is presented in (Lazzaro, Ryckebusch and Mahowald), this component can also be used as a simple comparator. We introduced a gain stage in the feedback loop of the common WTA in order to increase the accuracy of the comparison. Figure 46 represents the architecture of our modified WTA. As it is shown in Figure 47 the accuracy of our comparator is close to 2uV.

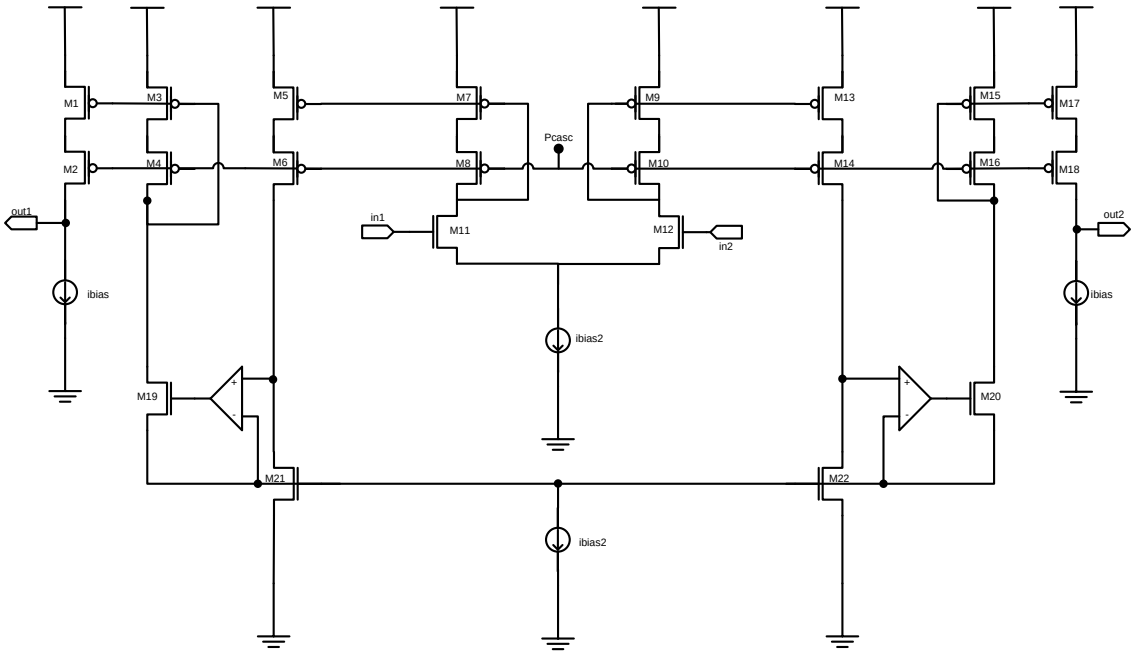



**Figure 69. Basic Architecture of Winner Take-all**

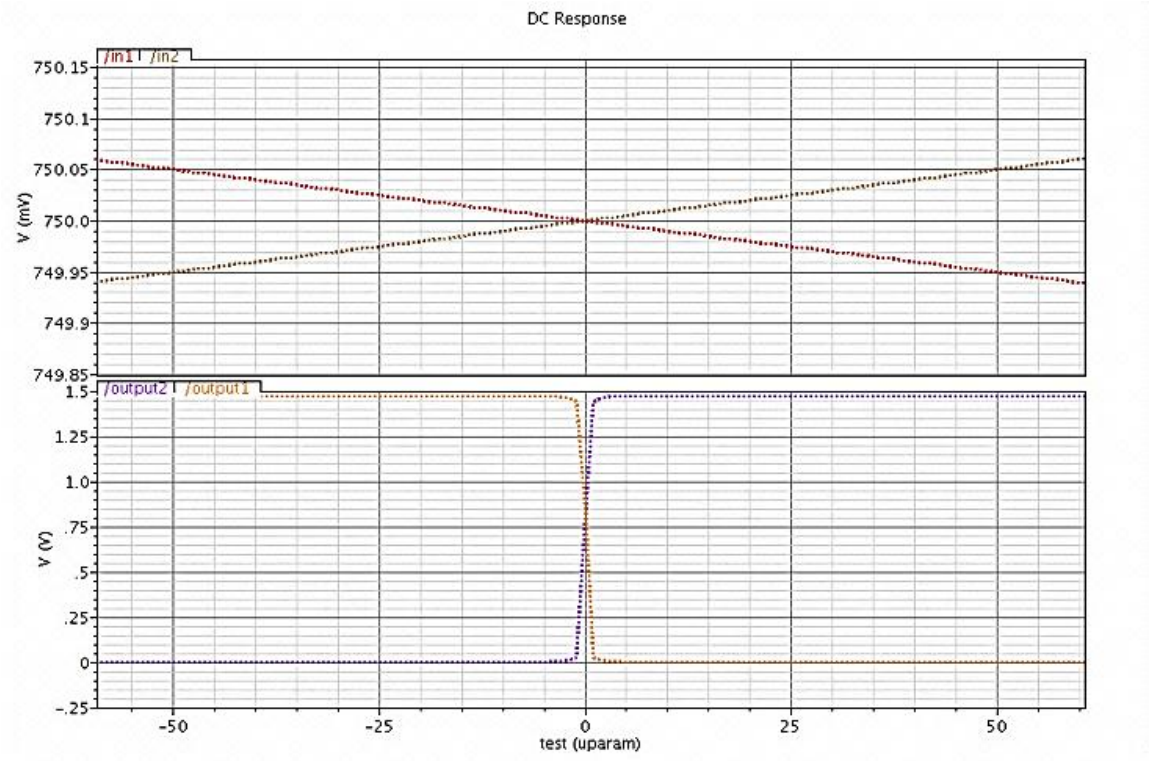



**Figure 70. The result of sweeping two inputs of Winner-Takes-All**
**Above: the input sweep of the comparator**
**Below: the output of WTA in response to the input above**

## E. Rectifier

The WTA introduced in last section was used in order to modify our inputs of the divider to only positive values. Figure 71 represents the main architecture of our rectifier used for this purpose. In order to remove the effect of common-mode input, a transconductance gain is added to design, which calibrates the output of the rectifier.

For input presented in Figure 72 (below) the output of the rectifier is shown in Figure 72 (above). As can be noticed in the result, the new WTA and the calibration

of output stage result in a very accurate rectifier. By using the rectifier and only one digital gate we converted our one quadrant divider to a four quadrant divider.

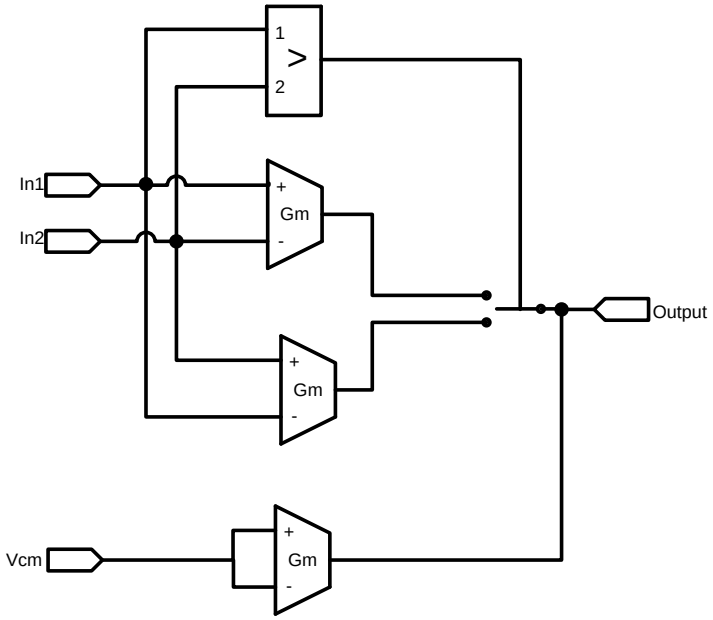



**Figure 71. Main architecture of rectifier**

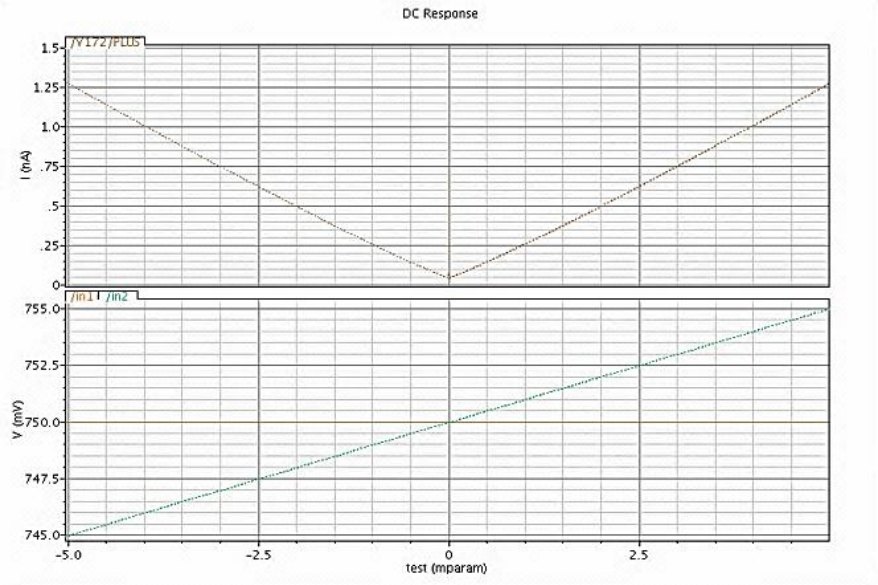



**Figure 72. Output and Input of the rectifier used in the input of divider**

## F. Multiplier:

In our design we need a multiplier which magnifies the amplitude of a signal by a digital constant value. The digital input helps us to choose from more simple designs. The common method used in these cases is using switched capacitors (Martin and Sedra), (Changyue, Peng and Yizhong), but the clocking and charge injection would leave us with an inaccurate calculation.

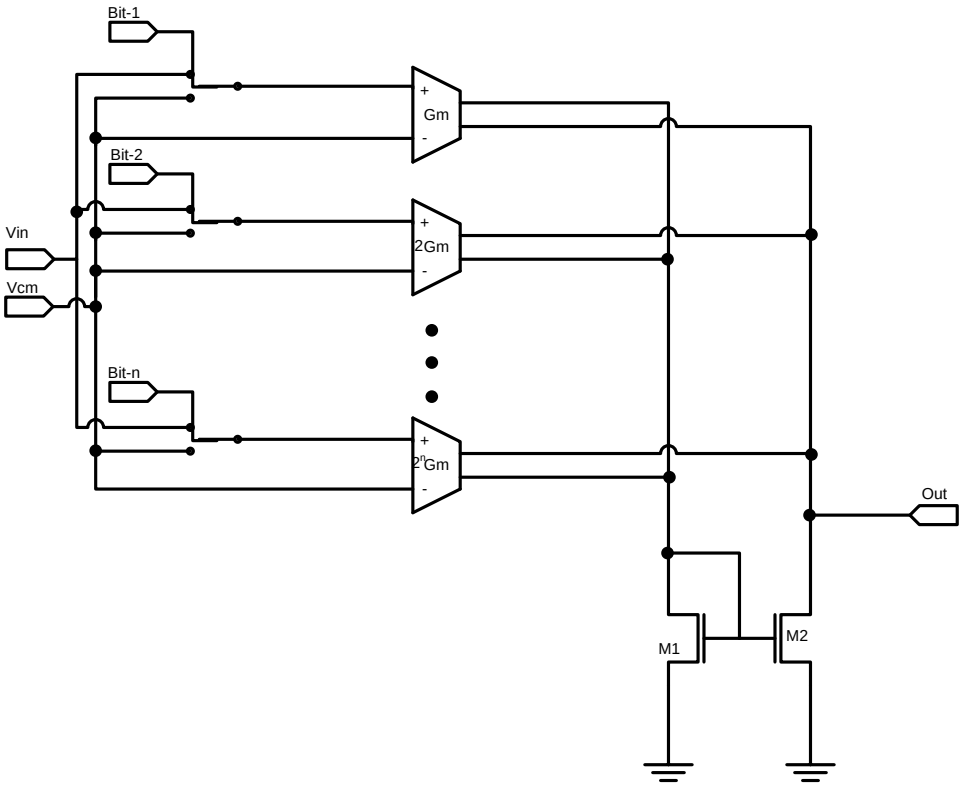



**Figure 73. Block diagram of multiplier (DAC)**

Instead, In our design, many differential input pairs are added to a simple unity gain amplifier. Each of these new pairs of inputs is sized in a way that it will

multiply the input signal by a predefined constant. Figure 73 represents the block diagram of our multiplier. This component can be used as a normal A/D converter. We swept the digital input from zero to 32 and the output is presented in Figure 74.

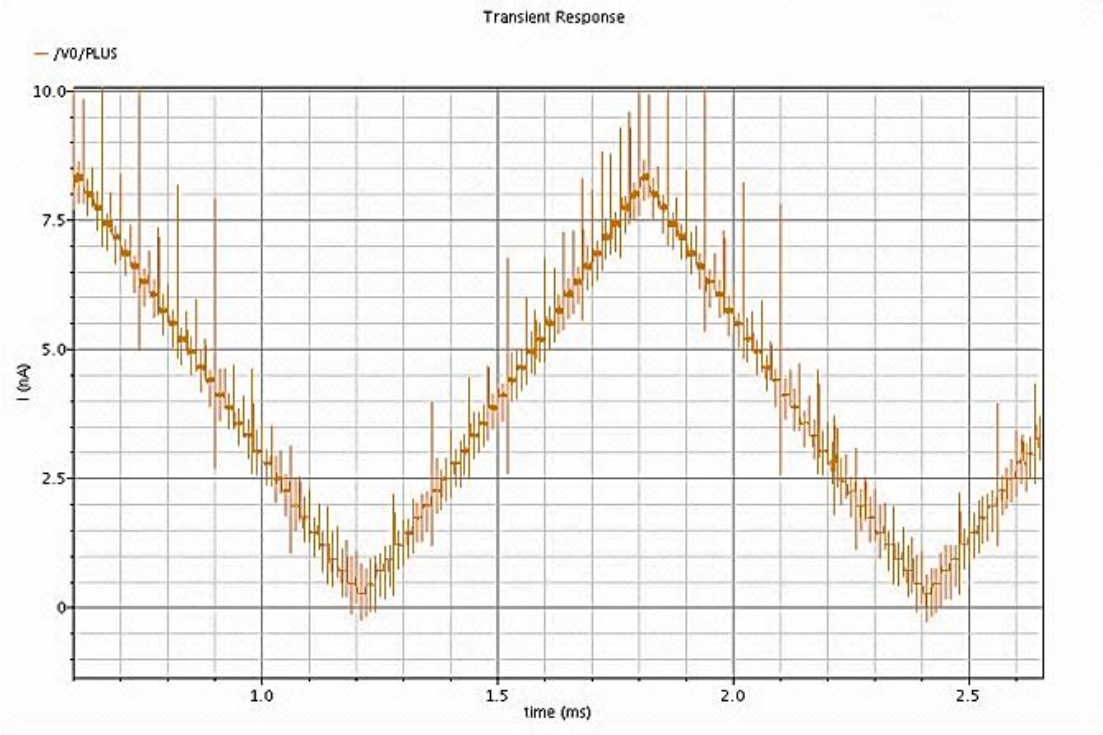



**Figure 74. Result of multiplying a small signal by digital value ranged from zero to 32**

## G. I-V converter

The final step is to convert our differential current to voltage Figure 75 shows the circuit diagram of the very accurate current to voltage converter we used in order to observe the output of our components.

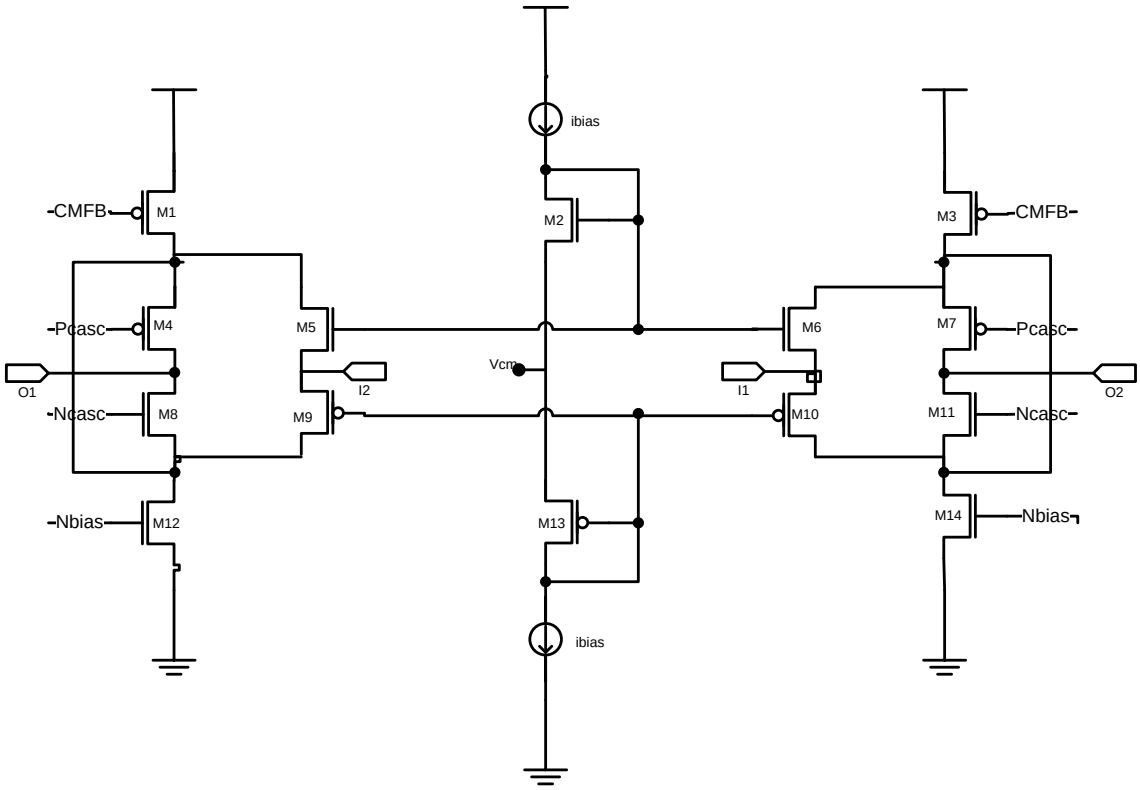



**Figure 75. Circuit diagram of an I-V converter**

## 3. Implementation

We implemented our components in IBM 0.13µm CMRF process and the final layout of different blocks are presented in Figure 76 Each block size is mentioned in the caption of the figure. As it was mentioned, the divider has two equivalent rectifiers for the divisor and dividend. Each of these rectifiers includes the simple WTA as a simple comparator. Figure 77 shows the layout the whole system, the automatic crosstalk removal module is on the first quadrant of the chip.

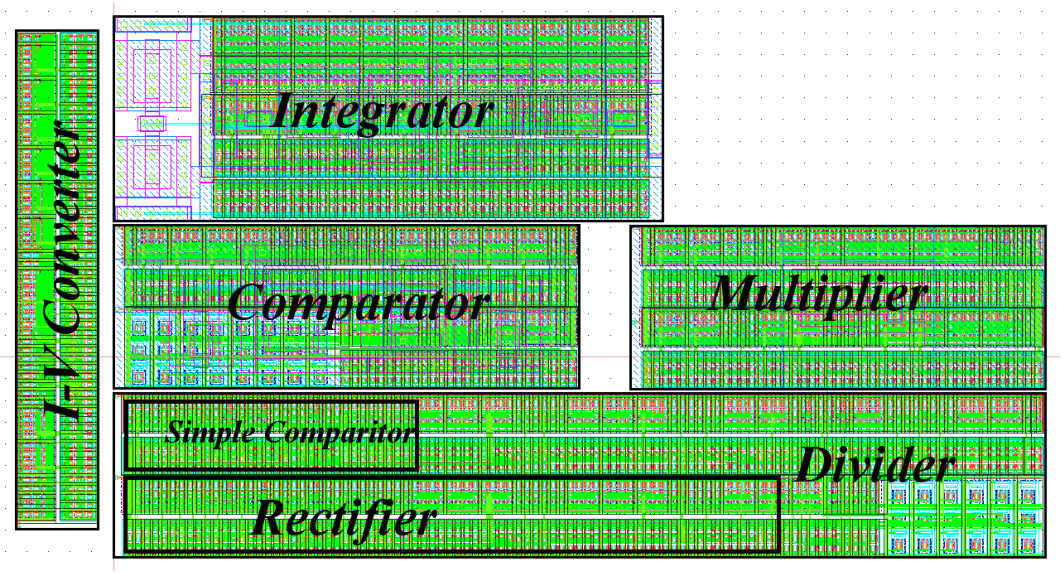



**Figure 76. Layout of different components implemented in 0.13um IBM cmrf technology. The size of each block is WTA= 1000um2, Comparator = 4000um2, Divider = 8000um2, Integrator = 6000um2, IV-Converter = 2000um2, Multiplier = 4000um2, and Rectifier = 30**

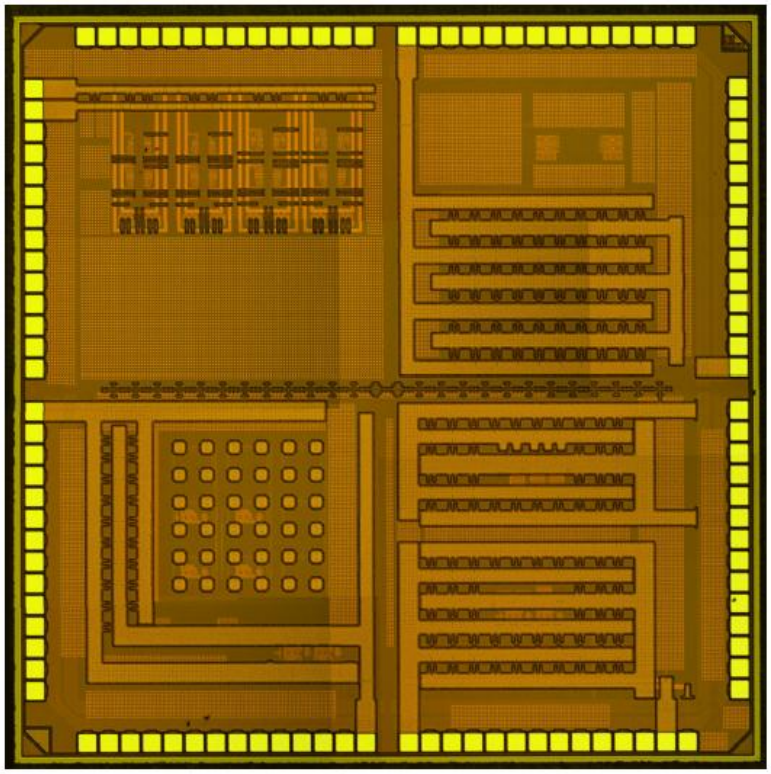


**Figure 77. The layout of fabricated chip**

## A. Test Environment

A PCB-Board is designed for testing the chip. Additionally, capacitors are used to remove the noise from sources. The whole test is embedded in a metal box to shield the design from outside noise (Figure 78). The input is produced by two function generator for clock and a LabView board is used to create the input signals. Two audio transformers is used to convert single ended inputs to differential pairs and fed to the system. The digital output of the test set is recorded using a logic analyzer and the analog test points are observed using a normal oscilloscope. The complete test bench is shown in Figure 79.

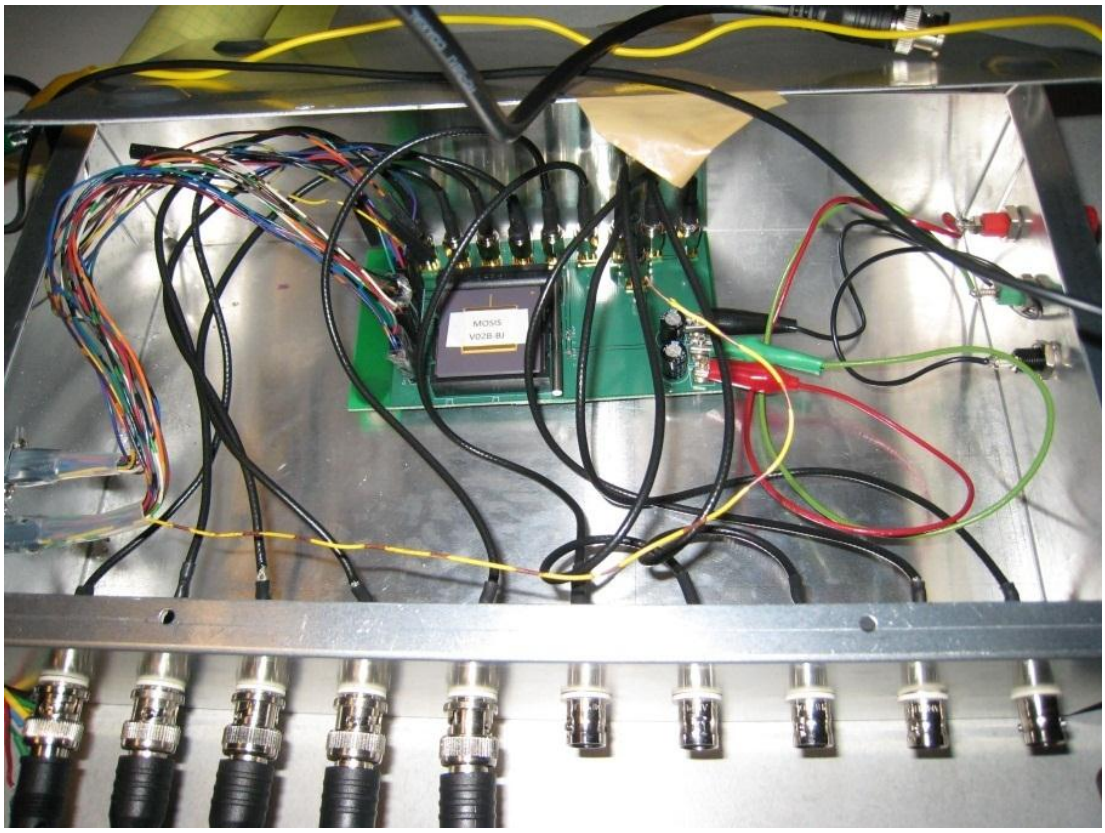


**Figure 78. The PCB Board and the chip inside the shielding box**

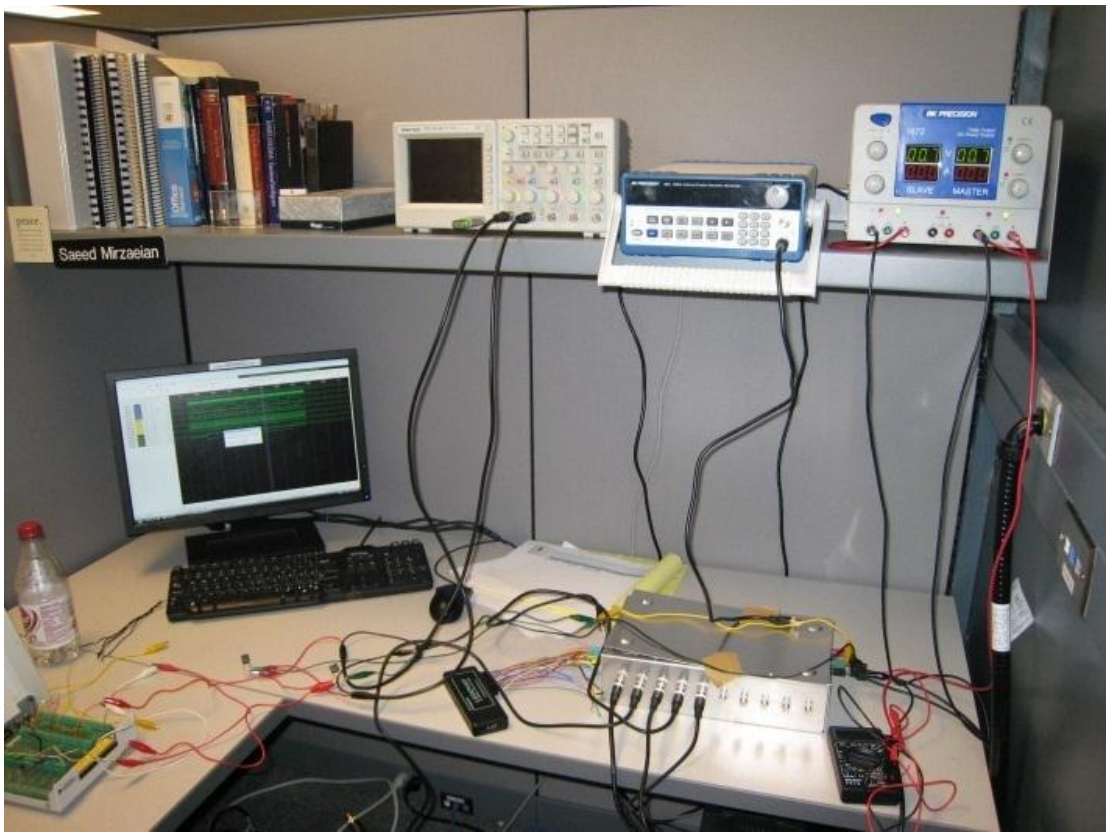



**Figure 79. Complete Test Bench of the chip**

## B. Test Results

As it was mentioned earlier, this design has two very distinctive sections the first section is the digital part and second part is the analog computation circuitry. We will discuss each of these two steps separately. While testing the digital section, we had a complete consistency with the required specifications Figure 80 shows the output of our designed counter which is used to coordinate different sections of the processor. However, defect appears as a result of interfacing with asynchronous analog part. This defect is the direct result of the JK-Flip-flop design. In common JKFF the master latch loses its feedback in memory stage. If initial inputs of the latch

request a change of value, but in the middle of stage request to keep the previous value, the previous oputput is already overwritten by the new value. To prevent this problem we modified our latch so that the value of slave latch is used to retrieve the previous value of the FF in memory stage of master latch.

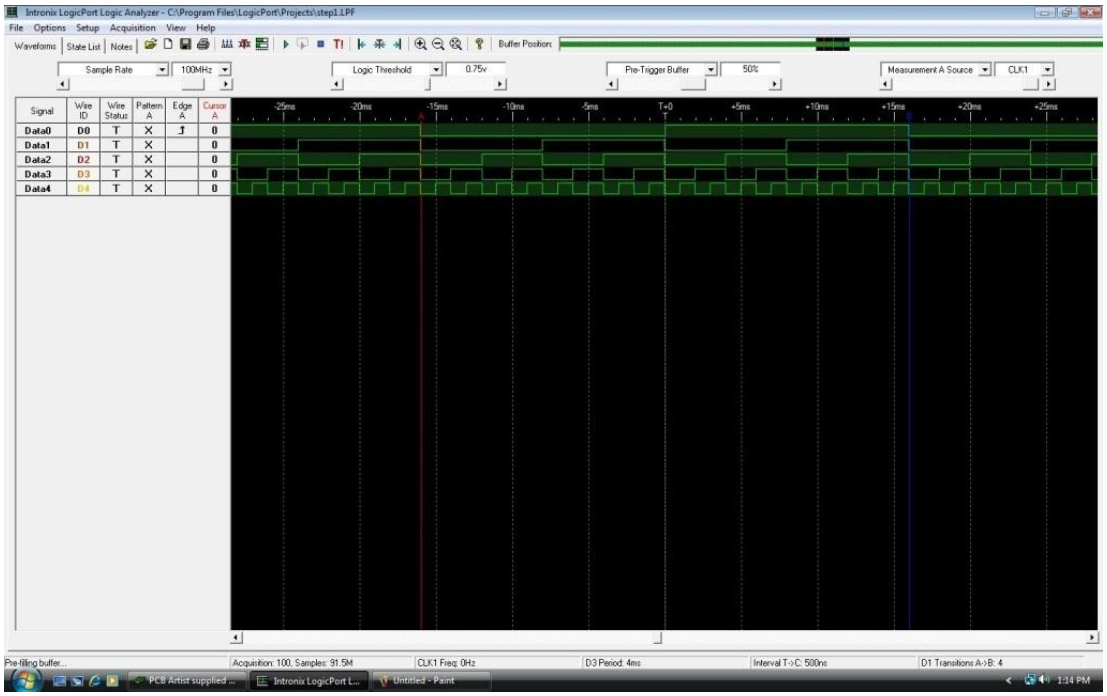



**Figure 80. The output result of counter on logic analyzer output**

On the contrary the analog subsection of the design suffers from an unnoticed bug which results in large increase in gain. This increase of gain derives the integrators of the design into saturation resulting in the failure of the whole design.

## 4. Conclusion

A series of novel or modified computational blocks are presented in this section. Starting with a very small low frequency integrator, then we introduced a very accurate but compact analog comparator. A very accurate four quadrant divider was presented, which employs the advantages of two mainstream dividers in order to achieve high range and accuracy. Finally, a very simple multiplier and IV converter are presented which are used calibrate and test the implemented circuits. These blocks are fabricated in the IBM 0.13µm process.

# V. Conclusion and Future Work

In this thesis we first present a brief overview of the neurons and brain functionality and introduced the challenges in the way of understanding human brain. Then we introduced the state-of-the-art method of recording neural activity implantable in the brain. Using a MATLAB-based model we investigate the possibility of advancement in the previously defined methods. This investigation led to a very critical observation, which shows the current algorithms lack the capability of application on general purpose devices, and mainly applicable in certain conditions. This limitation is rooted in the limited power budget and high volume of data recorded from probes.

This motivated us to devise an algorithm which is capable of circumventing the limitation of the state-of-the-art algorithm by removing redundancy which is embedded in the recording from different probes. This redundancy is the result of coupling of same neuron on differ probes. And it was shown that the redundancy removal would decrease the information embedded in the recording.

In the next step, we verified our algorithm using two different prototypes. These two prototypes were implemented using discrete electrical analog and digital components. Finally we devised a series of analog sub-threshold components that can be used in developing low power implantable version of the algorithm.

This algorithm opens the road for development of many new techniques and algorithms in order to make the implantable neural activity recorder more feasible. For example a small list of future work is presented below:

1- Developing a distributed linear prediction method based on attention: as we know not only neighboring probes might have crosstalk, but because of the structure of brain probes from different sections of the neural activity recorder might observe the same activity coupled in different neurons.

2- Developing new methods of spike detection: With the proposed algorithm, the signals would contain lower number of detectable spikes. This might open the road for spike detection algorithms that are fully implantable, even the learning and clustering sections.

3- Nonuniform quantization and companding: using the novel algorithm the signal would have less information embedded specially in the background neuron noise. Consequently, newer and more compact quantization method such as Logarithmic companding can be used to convert the analog raw data to digital.

4- Developing the signal routing mesh: The learning unit can be multiplexed between many different probes using a signal multiplexing mesh, this results in smaller area and lower power consumption.

## References


Moehlis, Jeff, Eric Shea-Brown and Herschel Rabitz. "Optimal inputs for phase models of spiking neurons." ASME Journal of Computational and Nonlinear Dynamics 1 (2006): 358-367.

Owens, Anthony L., et al. "Multi-electrode array for measuring evoked potentials from surface of ferret primary auditory cortex." Journal of Neuroscience Methods (1995): 209-220.

Rivara, Claire-Bénédicte. Betz cells of primary motor cortex Stereological and functional analysis: Thesis presented to the Faculty of Medicine of the University of Geneva. Geneva: University of Geneva, 2003.

Theogarajan, Luke Satish Kumar. Supramolecular Architectures for Neural Prostheses. Massachusetts: Massachusetts Institute of Technology, 2007.

Van Essen, D. C. "Visual areas of the mammalian cerebral cortex." Ann. Rev. Neurosci.,w (1979): 227-263.

Abeles, M. Corticonics: neural circuits of the Celebral Cortex. Cambridge University, 1991.

Adrian, E. D. The Physical background of Perception. Oxford: Claredon Press, 1946.

Aziz, Joseph N. Y., et al. "256-channel Neural Recording and Delta Compression Microsystem with 3D Electrodes." IEEE Journal of Solid-State Circuits (2009): 995-1005.

Chae, Moosang, et al. "A 128-Channel 6mW Wireless Neural Recording IC with On-the-Fly Spike Sorting and UWB Transmitter." IEEE international Solid-State Circuits Conference. IEEE, 2008. 146-147.

Chakrabartty, Shantanu, Amit Gore and Karim G. Oweiss. "An Adaptive multple-input multiple-output analog-to-digital converter for high density neuroprosthetic electrode arrays." 28th IEEE EMBS Annual International Conference. New York City: IEEE, 2006. 656-659.

Changyue, Ouyang , Chen Peng and Xie Yizhong. "Study of Switched Capacitor Multiplier." International Conference on Circuits and Systems. IEEE, 1991. 234-237.

Delbruck, T. "Bump Circuits for computing similarity and dissimilarity of analog voltages." CNS Memo. Pasadena: California Institute of Thechnology, Computation and Neural Systems Program, May 1993.

Enz, Christian C. and Gabor C. Temes. "'Circuit techniques for reducing the effects of Op-Amp imperfections: Auto zeroing, correlated, double sampling and chopper stabilization." Proceedings of the IEEE November 1996: 1584-1614.

Fischl, Bruse and Andres M. Dale. "Measuring the thickness of the human cerebral cortex from magnetic resonance images." PNAS 26 September 2000: 11050-11055.

Gold, Carl, et al. "On the Origin of the Extracellular Action Potential Waveform: A Modeling Study." Journal of Neurophysiology (1995): 3113-3128.

Gosselin, Benoit and Mohamad Sawan. "an ultra low power cmos action potential detector." IEEE Biomedical Circuits and Systems Conference. IEEE, 2008. 2733-2736.

Gosselin, Benoit, Amer Elias Ayoub and Mohamad Sawan. "a mixed signal multichip neural recording interface with bandwidth reduction." Biomedical Circuits and Systems Conference. IEEE, 2007. 49-52.

Helmholtz, H. Popular Scietific Lectures. London: Longmans, 1889.

Henze, Darrell A., et al. "Intracellular Features Predicted by Extracellular Recordings in the Hippocampus In Vivo." Journal of Neurophysiology (2000): 390-400.

I. Obeid, P.D. Wolf. "Evaluation of Spike-Detection Algorithms for a Brain-Machine Interface Application." IEEE transactions on biomedical Engineering (2004): 905-911.

J. Han, M. Kamber. Data Mining concepts and techniques. Morgan Kaufmann, 2000.

Johnston , Daniel and Samuel Miao-Sin Wu. Foundations of Cellular Neurophysiology. The MIT Press, 1994.

Jones, E. G. "Viewpoint: The core and Matrix of Thalamic organization." Neuroscience (1998): 331-345.

Kaiser, J. F. "On simple algorithm to calculate the energy of a signal." IEEE int. conf. acoust. Albuquerque: IEEE, 1990. 381-384.

Lazzaro, J., et al. "Winner-Take-All Networks of O(N) Complexity." Advances in neural information processing (1989): 703 - 711.

Lewicki, M. S. "Bayesian modeling and classification of neural signals." Advances in Neural Information Processing Systems (1994): 590-597.

LLano, Isabel and Hersch M. Gerschenfeld. "Inhabitory Synaptic Currents in Stellate Cells of Rat Cerebellar Slices." Journal of Physiology (1993): 177-200.

Lu, P. -H., C. -Y. Wu and M. -K. Tsai. "The Design of Fully Differential CMOS Operational Amplifiers without Extra Common-Mode Feedback Circuits." Analog Integrated Circuits and Signal Processing (1993): 173-186.

M. A. L. Nicolelis, D. Dimitrov, J. M. Carmena, R. Crist, G. Lehew, J. D. Kralik. "Chronic, multisite, multielectrode recording in macaque monkeys." PNAS (2003): 11041-11046.

Mallat, S. G. "A theory for multiresolution signal decomposision: the wavelet representation." IEEE transaction on Pattern Analysis and machine intelligance (1989): 674-693.

Martin, Ken and Adel S. Sedra. "Switched-Capacitor Building Blocks for Adaptive Systems." IEEE Transaction on Circuits and Systems (1981): 576-584.

Mason, A. and A. Larkman. "Correlations between morphology and electrophysiology of pyramidal neurons in slices of rat visual cortex. II. Electrophysiology." Journal of Neuroscience (1990): 1415-1428.

Mehta, Ashesh D., Istvan Ulbert and Charles E. Schroeder. "Intermodal Selective Attention in Monkeys. II:Physiological Mechanisim of Modulation." Celebral Cortex April 2000: 359-370.

—. "Intermodel Selective attention in Monkeys I: Distribution and Timing of Effects across Visual Area." Cerebral Cortex Apr 2000: 343-358.

Nicholls, John G., A. Robert Martin and Bruce G. Wallace. From Neuron to Brain: A Cellular and Molecular Approach to the Function of the Nervous System. Sinauer Associates Inc, 1992.

Nordhausena, Craig T., Edwin M. Maynarda and and Richard A. Normann. "single unit recording capabilities of a 100 microelectrode array." Brain Res. (1996): 129-140.

Olsson, Roy H. and Kensall D. Wise. "a three-dimensional neural recording microsystem with implantable data compression circuitery." IEEE journal of solid-state circuits (2005): 2796-2804.

Oweiss, Karim G., David J. Anderson and Marios M. Papaefthymiou. "Optimizing Signal Coding in Neural Interface System-on-a-Chip Modules." 25th annual International Conference of the IEEE EMBS. Cancun: IEEE, 2003. 2216-2219.

—. "Optimizing Signal Coding in Neural Interface System-on-a-Chip Modules." the 25th annual International Conference if the IEEE EMBS. Cancun: IEEE, 2003. 2216-2219.

R. Q. Quiroga, Z. Nadasdy, Y. Ben-Shaul. "unsupervised spike detection and sorting with wavelets and paramagnetics clustering." Neural Computation (2004): 1661-1687.

Rizk, Micheal, et al. "A single-chip signal Processing and telemetry engine for an implantable 96-channel neural data acquisition system." Journal of Neural Engineering (2007): 309-321.

Rubio-Gerrido, Pablo, et al. "Thalamic Input to Distal Apical Denderites in Neocortical LAyer 1 is Massive and Highly Convergent." Celebral Cortex (2009): 2380-2395.

S. Gibson, J. W. Judy, D. Markovic. "Comparison of spike-sorting Algorithm for Future Hardware Implementation." 30th Anuual international IEEE EMBS Conference. Vancouver: IEEE, 2008. 5015-5020.

Shutao, Li, Wang Yaonan and Wu Jie. "Design of low voltage and Low Power Fully Integrated Filer Based on Log-Domain Current-mode Integrator." Journal of Electronics (2001): 346-350.

Stefanou, N. and Sameer R. Sonkusale. "An Average Low Offset Comparator for 1.25 GSamples/S ADC in 0.18um CMOS." IEEE International Conference on Electronics, Circuits and Systems. IEEE, 2004. 246-249.

Suzuki, T. ; Mabuchi, K. ; Takeuchi, S. "A 3D flexible parylene probe array for multichannel neural recording." First International IEEE EMBS Conference on Neural Engineering. IEEE, 2003. 154 - 156 .

Wilamowski, Bogdan M. "VLSI analog multipler/divider circuit." IEEE (1998): 493-496.

Wise, Kensall D., et al. "Microelectrodes, Microelectronics, and Implantable Neural Microsystems." Proceedings of the IEEE (2008): 1184-1202.

Yu, Chong-Gun and Randall L. Geiger. "An Automatic Offset Compensation Scheme with Ping-Pong Control for CMOS Operational Amplifiers." IEEE Journal of Solid-State Circuits (1994): 601-610.

Yu, H, et al. "a wireless microsystem for multichannel neural recording microprobes." Solid-State censor, Actuactor, and Microsystem Workshop. 2004. 107-110.

Z. Yang, Q. Zhao, and W. Liu. "Spike Feature Extraction Using Informative Samples." Advances in Neural Information Processing Systems (NIPS 21) (2009): 1865 - 1872.

Zarabadi, Seyed R., Mohammed Ismail and Chung-Chih Hung. "High Performance Analog Computational Circuits." IEEE Journal of Solid-State Circuits (1998): 644-649.

# Appendix

## Appendix I

MATLAB code for generating the brain model & synthesizing neural signal

```
clear;
load('C:\Users\Saeed\Documents\MATLAB\Neuron
Distribution\Data\Data_cut_noDC.mat');
distancebetweenprobes = 100;
numberofprobes = 12;
probeheight = 200;
rows = ceil(sqrt(numberofprobes));
Border = 200;
Box = 2 * Border + (rows - 1) * distancebetweenprobes;
PinRad = 25;
for i=1:numberofprobes
    PinX(i) = Border + (mod(i - 1, rows))* distancebetweenprobes;
    PinY(i) = Border + (floor((i - 1)/rows))* distancebetweenprobes;
    PinZ(i) = Box - probeheight;
end;
PinShld = 0;
distance = 140;
BetzMu = (2745/10^9)*Box^3;
BetzSD = (605.54/10^9)*Box^3;
NoBetzBox = floor(normrnd(BetzMu, BetzSD))
PyrmMu = (19755/10^9)*Box^3;
PyrmSD = (9241.75/10^9)*Box^3;
NoPyrmBox = floor(normrnd(PyrmMu, PyrmSD))
SmthMu = (4500/10^9)*Box^3;
SmthSD = (1872.32/10^9)*Box^3;
NoSmthBox = floor(normrnd(SmthMu, SmthSD))
SpinMu = (3000/10^9)*Box^3;
SpinSD = (1248.21/10^9)*Box^3;
NoSpinBox = floor(normrnd(SpinMu, SpinSD))
Color = ones(2,21);
Color = Color ./ (3/4);
[a,b,c] = cylinder;

for i=1:numberofprobes
    surf(a.*PinRad + PinX(i),b.*PinRad +PinY(i),c.*(Box -PinZ(i)) +
PinZ(i),  'FaceColor','yellow','EdgeColor','none')
    hold on;
end;


NoNeurBox = NoSpinBox + NoSmthBox + NoPyrmBox + NoBetzBox;
vol = 0;
NoNeuProc = 0;
```

```
for i=1:NoBetzBox
    BetzVolMu = 86685.9;
    BetzVolSD = 19934;
    VolBetz(i) = normrnd(BetzVolMu, BetzVolSD);
    RadBetz(i) = ((VolBetz(i)*3)/(4 * pi))^(1/3);
    fail = 1;
    while fail > 0
        fail = 0;
        LXBetz(i) = randi(Box);
        LYBetz(i) = randi(Box);
        LZBetz(i) = randi(Box);
        for x=1:numberofprobes
            if ((((LXBetz(i) - PinX(x))^2 + (LYBetz(i) - PinY(x))^2)
< (RadBetz(i) + PinRad)^ 2) && (LZBetz(i) + RadBetz(i) > PinZ(x)))
                if (LZBetz(i) > PinZ(x))
                    current = sqrt((LXBetz(i) - PinX(x))^2 +
(LYBetz(i) - PinY(x))^2);
                    LXBetz(i) = (RadBetz(i) + PinRad) *(LXBetz(i) -
PinX(x))/current+ PinX(x);
                    LYBetz(i) = (RadBetz(i) + PinRad) *(LYBetz(i) -
PinY(x))/current+ PinY(x);
                else
                    LZBetz(i) =  PinZ(x) - RadBetz(i);
                end;
            end;
        end
        for j =1:NoNeuProc
            if (((LXBetz(i) - LXBetz(j))^2 + (LYBetz(i) -
LYBetz(j))^2 + (LZBetz(i) - LZBetz(j))^2 ) < (RadBetz(i) +
RadBetz(j))^ 2)
              fail = 1;
            end;
        end;
    end;
    vol = vol + VolBetz(i);
    NoNeuProc = NoNeuProc + 1;
end;

[x,y,z] = sphere;
Color = zeros(20);
for i=1:NoBetzBox
    surf((x.*RadBetz(i) + LXBetz(i)),(y.*RadBetz(i) +
LYBetz(i)),(z.*RadBetz(i) + LZBetz(i)),
'FaceColor','red','EdgeColor','none');

end;

NoNeuProc = 0;
for i=1:NoPyrmBox
    PyrmVolMu = 4274;
    PyrmVolSD = 438.3;
    VolPyrm(i) = normrnd(PyrmVolMu, PyrmVolSD);
    RadPyrm(i) = ((VolPyrm(i)*3)/(4 * pi))^(1/3);
```

```
    fail = 1;
    while fail > 0
        fail = 0;
        LXPyrm(i) = randi(Box);
        LYPyrm(i) = randi(Box);
        LZPyrm(i) = randi(Box);
        for x=1:numberofprobes
            if ((((LXPyrm(i) - PinX(x))^2 + (LYPyrm(i) - PinY(x))^2)
< (RadPyrm(i) + PinRad)^ 2) && (LZPyrm(i) + RadPyrm(i) > PinZ(x)))
                if (LZPyrm(i) > PinZ(x))
                    current = sqrt((LXPyrm(i) - PinX(x))^2 +
(LYPyrm(i) - PinY(x))^2);
                    LXPyrm(i) = (RadPyrm(i) + PinRad) *(LXPyrm(i) -
PinX(x))/current+ PinX(x);
                    LYPyrm(i) = (RadPyrm(i) + PinRad) *(LYPyrm(i) -
PinY(x))/current+ PinY(x);
                else
                    LZPyrm(i) =  PinZ(x) - RadPyrm(i);
                end;
            end;
        end;
        for j =1:NoNeuProc
            if (((LXPyrm(i) - LXPyrm(j))^2 + (LYPyrm(i) -
LYPyrm(j))^2 + (LZPyrm(i) - LZPyrm(j))^2 ) < (RadPyrm(i) +
RadPyrm(j))^ 2)
              fail = 1;
            end;
        end;
        for j =1:NoBetzBox
            if (((LXPyrm(i) - LXBetz(j))^2 + (LYPyrm(i) -
LYBetz(j))^2 + (LZPyrm(i) - LZBetz(j))^2 ) < (RadPyrm(i) +
RadBetz(j))^ 2)
              fail = 1;
            end;
        end;
    end;
    vol = vol + VolPyrm(i);
    NoNeuProc = NoNeuProc + 1;
end;
[x,y,z] = sphere;
Color = ones(20);
for i=1:NoPyrmBox
    surf((x.*RadPyrm(i) + LXPyrm(i)),(y.*RadPyrm(i) +
LYPyrm(i)),(z.*RadPyrm(i) + LZPyrm(i)),
'FaceColor','blue','EdgeColor','none')
    hold on;
end;

NoNeuProc = 0;
for i=1:NoSmthBox
    SmthVolMu = 321.55;
    SmthVolSD = 60.15;
    VolSmth(i) = normrnd(SmthVolMu, SmthVolSD);
```

```
    RadSmth(i) = ((VolSmth(i)*3)/(4 * pi))^(1/3);
    fail = 1;
    while fail > 0
        fail = 0;
        LXSmth(i) = randi(Box);
        LYSmth(i) = randi(Box);
        LZSmth(i) = randi(Box);
        for x=1:numberofprobes
          if ((((LXSmth(i) - PinX(x))^2 + (LYSmth(i) - PinY(x))^2) <
(RadSmth(i) + PinRad)^ 2)  && (LZSmth(i) + RadSmth(i) > PinZ(x)))
              if (LZSmth(i) > PinZ(x))
                  current = sqrt((LXSmth(i) - PinX(x))^2 +
(LYSmth(i) - PinY(x))^2);
                  LXSmth(i) = (RadSmth(i) + PinRad) *(LXSmth(i) -
PinX(x))/current+ PinX(x);
                  LYSmth(i) = (RadSmth(i) + PinRad) *(LYSmth(i) -
PinY(x))/current+ PinY(x);
              else
                  LZSmth(i) =  PinZ(x) - RadSmth(i);
              end;
          end;
        end;
        for j =1:NoNeuProc
            if (((LXSmth(i) - LXSmth(j))^2 + (LYSmth(i) -
LYSmth(j))^2 + (LZSmth(i) - LZSmth(j))^2 ) < (RadSmth(i) +
RadSmth(j))^ 2)
              fail = 1;
            end;
        end;
        for j =1:NoPyrmBox
            if (((LXSmth(i) - LXPyrm(j))^2 + (LYSmth(i) -
LYPyrm(j))^2 + (LZSmth(i) - LZPyrm(j))^2 ) < (RadSmth(i) +
RadPyrm(j))^ 2)
              fail = 1;
            end;
        end;
        for j =1:NoBetzBox
            if (((LXSmth(i) - LXBetz(j))^2 + (LYSmth(i) -
LYBetz(j))^2 + (LZSmth(i) - LZBetz(j))^2 ) < (RadSmth(i) +
RadBetz(j))^ 2)
              fail = 1;
            end;
        end;
    end;
    NoNeuProc = NoNeuProc + 1;
    vol = vol + VolSmth(i);
end;
[x,y,z] = sphere;
Color = Color ./ 2;
for i=1:NoSmthBox
    surf((x.*RadSmth(i) + LXSmth(i)),(y.*RadSmth(i) +
LYSmth(i)),(z.*RadSmth(i) + LZSmth(i)),
'FaceColor','green','EdgeColor','none')
    hold on;
```

```
end;


NoNeuProc = 0;
for i=1:NoSpinBox
    SpinVolMu = 321.55;
    SpinVolSD = 60.15;
    VolSpin(i) = normrnd(SpinVolMu, SpinVolSD);
    RadSpin(i) = ((VolSpin(i)*3)/(4 * pi))^(1/3);
    fail = 1;
    while fail > 0
        fail = 0;
        LXSpin(i) = randi(Box);
        LYSpin(i) = randi(Box);
        LZSpin(i) = randi(Box);
        for x=1:numberofprobes
            if ((((LXSpin(i) - PinX(x))^2 + (LYSpin(i) - PinY(x))^2)
< (RadSpin(i) + PinRad)^ 2) && (LZSpin(i) + RadSpin(i) > PinZ(x)))
                if (LZSpin(i) > PinZ(x))
                    current = sqrt((LXSpin(i) - PinX(x))^2 +
(LYSpin(i) - PinY(x))^2);
                    LXSpin(i) = (RadSpin(i) + PinRad) *(LXSpin(i) -
PinX(x))/current+ PinX(x);
                    LYSpin(i) = (RadSpin(i) + PinRad) *(LYSpin(i) -
PinY(x))/current+ PinY(x);
                else
                    LZSpin(i) =  PinZ(x) - RadSpin(i);
                end;
            end;
        end;
        for j =1:NoNeuProc
            if (((LXSpin(i) - LXSpin(j))^2 + (LYSpin(i) -
LYSpin(j))^2 + (LZSpin(i) - LZSpin(j))^2 ) < (RadSpin(i) +
RadSpin(j))^ 2)
                fail = 1;
            end;
        end;
        for j =1:NoPyrmBox
            if (((LXSpin(i) - LXPyrm(j))^2 + (LYSpin(i) -
LYPyrm(j))^2 + (LZSpin(i) - LZPyrm(j))^2 ) < (RadSpin(i) +
RadPyrm(j))^ 2)
                fail = 1;
            end;
        end;
        for j =1:NoBetzBox
            if (((LXSpin(i) - LXBetz(j))^2 + (LYSpin(i) -
LYBetz(j))^2 + (LZSpin(i) - LZBetz(j))^2 ) < (RadSpin(i) +
RadBetz(j))^ 2)
                fail = 1;
            end;
        end;
        for j =1:NoSmthBox
```

```
            if (((LXSpin(i) - LXSmth(j))^2 + (LYSpin(i) -
LYSmth(j))^2 + (LZSpin(i) - LZSmth(j))^2 ) < (RadSpin(i) +
RadSmth(j))^ 2)
              fail = 1;
            end;
        end;
    end;
    NoNeuProc = NoNeuProc + 1;
    vol = vol + VolSpin(i);
end;

[x,y,z] = sphere;
Color = Color ./ 2;
for i=1:NoSpinBox
    surf((x.*RadSpin(i) + LXSpin(i)),(y.*RadSpin(i) +
LYSpin(i)),(z.*RadSpin(i) + LZSpin(i)),
'FaceColor','yellow','EdgeColor','none')
    hold on;
end;

axis equal;
camlight left; lighting phong

filled = vol / Box^3

distance = 140;
k = 0;
nrndtrb = zeros(1,1);

for i=1:NoBetzBox
    k = k + 1;
    nrndtrb(k,1) = 1;
    for x=1:numberofprobes
        if (LZBetz(i) < PinZ(x) - RadBetz(i) )
            nrndtrb(k,1+x) = sqrt((LZBetz(i) - PinZ(x))^2 +
(LXBetz(i) - PinX(x))^2 + (LYBetz(i) - PinY(x))^2) - RadBetz(i);
        elseif (LZBetz(i) > PinShld + RadBetz(i) )
            nrndtrb(k,1+x) = sqrt((LZBetz(i) - PinShld)^2 +
(LXBetz(i) - PinX(x))^2 + (LYBetz(i) - PinY(x))^2) - RadBetz(i);
        else
            nrndtrb(k,1+x) = sqrt((LXBetz(i) - PinX(x))^2 +
(LYBetz(i) - PinY(x))^2) - PinRad - RadBetz(i);
        end
    end;
end

for i=1:NoPyrmBox
    k = k + 1;
    nrndtrb(k,1) = 1;
    for x=1:numberofprobes
        if (LZPyrm(i) < PinZ(x) - RadPyrm(i) )
```

```
            nrndtrb(k,1+x) = sqrt((LZPyrm(i) - PinZ(x))^2 +
(LXPyrm(i) - PinX(x))^2 + (LYPyrm(i) - PinY(x))^2) - RadPyrm(i);
        elseif (LZPyrm(i) > PinShld + RadPyrm(i) )
            nrndtrb(k,1+x) = sqrt((LZPyrm(i) - PinShld)^2 +
(LXPyrm(i) - PinX(x))^2 + (LYPyrm(i) - PinY(x))^2) - RadPyrm(i);
        else
            nrndtrb(k,1+x) = sqrt((LXPyrm(i) - PinX(x))^2 +
(LYPyrm(i) - PinY(x))^2) - PinRad - RadPyrm(i);
        end
    end;
end

for i=1:NoSmthBox
    k = k + 1;
    nrndtrb(k,1) = 1;
    for x=1:numberofprobes
        if (LZSmth(i) < PinZ(x) - RadSmth(i) )
            nrndtrb(k,1+x) = sqrt((LZSmth(i) - PinZ(x))^2 +
(LXSmth(i) - PinX(x))^2 + (LYSmth(i) - PinY(x))^2) - RadSmth(i);
        elseif (LZSmth(i) > PinShld + RadSmth(i) )
            nrndtrb(k,1+x) = sqrt((LZSmth(i) - PinShld)^2 +
(LXSmth(i) - PinX(x))^2 + (LYSmth(i) - PinY(x))^2) - RadSmth(i);
        else
            nrndtrb(k,1+x) = sqrt((LXSmth(i) - PinX(x))^2 +
(LYSmth(i) - PinY(x))^2) - PinRad - RadSmth(i);
        end
    end;
end

for i=1:NoSpinBox
    k = k + 1;
    nrndtrb(k,1) = 1;
    for x=1:numberofprobes
        if (LZSpin(i) < PinZ(x) - RadSpin(i) )
            nrndtrb(k,1+x) = sqrt((LZSpin(i) - PinZ(x))^2 +
(LXSpin(i) - PinX(x))^2 + (LYSpin(i) - PinY(x))^2) - RadSpin(i);
        elseif (LZSpin(i) > PinShld + RadSpin(i) )
            nrndtrb(k,1+x) = sqrt((LZSpin(i) - PinShld)^2 +
(LXSpin(i) - PinX(x))^2 + (LYSpin(i) - PinY(x))^2) - RadSpin(i);
        else
            nrndtrb(k,1+x) = sqrt((LXSpin(i) - PinX(x))^2 +
(LYSpin(i) - PinY(x))^2) - PinRad - RadSpin(i);
        end
    end;
end

clear('LXSpin');
clear('LXSmth');
clear('LXBetz');
clear('LXPyrm');
clear('LYSpin');
clear('LYSmth');
clear('LYBetz');
```

```
clear('LYPyrm');
clear('LZSpin');
clear('LZSmth');
clear('LZBetz');
clear('LZPyrm');
clear('RadSpin');
clear('RadSmth');
clear('RadBetz');
clear('RadPyrm');
clear('VolSpin');
clear('VolSmth');
clear('VolBetz');
clear('VolPyrm');

temp = sum(nrndtrb, 1);
nnurons = temp(1, 1);
lambda = 14.6;
RunLength = 20;
runsamples = RunLength * 50000;
maxnbrspk = ceil((runsamples * 2) / (lambda / 0.02));
n1 = exprnd(lambda,nnurons,maxnbrspk);
n1sum = cumsum(n1,2);
spike = zeros(1, runsamples);

[d1,d2] = size(nrndtrb);
result = zeros(numberofprobes,runsamples);

for n=1:nnurons
    spike = zeros(1, runsamples);
    for k=1:maxnbrspk
        spiketime = n1sum(n,k);
        sample = round(spiketime/0.02);
        if sample < runsamples
            for s=1:72
                spike(1, sample+s) = Data2(1,s);
            end;
        end;
    end;
    tab = 0;
    place = 0;
    while tab < n,
        place = place + 1;
        tab = tab + nrndtrb(place,1);
    end;
    for x=1:numberofprobes
        distance = nrndtrb(place,1+x)+ 1;
        neurons = nrndtrb(place,1);
        for j=1:neurons
            result(x,:) = result(x,:) + ( spike(1, 1:runsamples) ./
(distance^2));
        end;
    end;
end;
```

```
figure;
hold off;
for x=1:numberofprobes
    result1 = zeros(2,runsamples);
    result1(1,:) = 0:runsamples - 1;
    result1(2,:) = result(x,:);
    filename = strcat('probe',int2str(x),'.mat');
    save(filename,'result1');
    subplot(numberofprobes,1,x);plot(result1(2,:).*100,'green');
    hold on;

end;
clear();
```

## Appendix II

MATLAB code for deriving the PDF of synthesizing neural signal

```
numberofprobes = 12;
quantizer = 10000;
color = ['grbygrbygrby'];
fy = 100000;
for x=1:numberofprobes
    filename =
strcat('C:\Users\Saeed\Documents\MATLAB\probe',int2str(x),'.mat');
    load(filename);
    result2(1,:) = result1(2,:);
    y = (quantizer * 0.8) / (max(result2,[],2) - min(result2,[],2));
    if (y < fy)
        fy = y;
    end
end
y = fy;
for x=1:numberofprobes
    filename =
strcat('C:\Users\Saeed\Documents\MATLAB\probe',int2str(x),'.mat');
    load(filename);
    s = zeros(2, quantizer);
    result2(1,:) = result1(2,:);
%    y = (quantizer * 0.8) / (max(result2,[],2) -
min(result2,[],2));
    means(x) = mean(result2);
    vars(x) = var(result2);

    for k = 1:quantizer
        s(1, k) = floor(k - (quantizer / 2)) / y;
    end;

    for R = 1:1000000
        s(2, floor(result2(R)* y + (quantizer / 2))) = s(2,
floor(result2(R)* y + (quantizer / 2))) + 1;
    end;

    ent(x) = 0;
    z(x) = 0;
    for k= 1:quantizer
        if (s(2,k) ~= 0)
            p = s(2,k) /1000000;
            z(x) = z(x) + p;
            ent(x) = ent(x) - p * log2(p);
        end
    end
   hold on;
end;
```

## Appendix III

MATLAB code for deriving the entropy of the compressed synthesizing neural signal

```
%mult = [ 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5];
%mult = [0 6.5 1.75 0 0 5.25 4 0.5 1 3.75]
mult = [ 0 0 0 0 0 0 0 0 0 0];
times = size(mult);
times = times(2);
steps = 10;
s = zeros(2, quantizer);
filename = strcat('C:\Users\Saeed\Documents\MATLAB\probe6.mat');
load(filename);
result2(1,:) = result1(2,:);
filename = strcat('C:\Users\Saeed\Documents\MATLAB\probe7.mat');
load(filename);
result3(1,:) = result1(2,:);

for W = 1:steps
    for R = 1:(1000000 /times)
        index = (W-1) * (1000000/times)+ R;
        result3(index) = result3(index) - mult(W) * result2(index);
    end;
end;

result3 = result3((1):(steps*1000000/times));
 size(result3)
y = (quantizer * 0.8) / (max(result3,[],2) - min(result3,[],2));
%y = fy;
    for k = 1:quantizer
        s(1, k) = floor(k - (quantizer / 2)) / y;
    end;

    for R = 1:(1000000*steps/times)
      s(2, floor(result3(R)* y + (quantizer / 2))) = s(2,
floor(result3(R)* y + (quantizer / 2))) + 1;
    end;
    entres = 0;
    z = 0;
    for k= 1:quantizer
        if (s(2,k) ~= 0)
            p = s(2,k) /(1000000*steps/times);
            z = z + p;
            entres = entres - p * log2(p);
        end
    end
entres
```

## Appendix IV

MATLAB code for deriving the joint PDF of synthesizing neural signal

```
clear();
quantizer = 1000;
filename = strcat('C:\Users\Saeed\Documents\MATLAB\probe6.mat');
load(filename);
result2(1,:) = result1(2,:);
s = zeros(quantizer, quantizer);
filename = strcat('C:\Users\Saeed\Documents\MATLAB\probe7.mat');
load(filename);
result3(1,:) = result1(2,:);
y1 = (quantizer * 0.8) / (max(result2,[],2) - min(result2,[],2));
y2 = (quantizer * 0.8) / (max(result3,[],2) - min(result3,[],2));

    for R = 1:10000
        for T = 1:10000
            s(floor(result2(R)* y1 + (quantizer / 2)),
floor(result3(T)* y2 + (quantizer / 2))) = s(floor(result2(R)* y1 +
(quantizer / 2)), floor(result3(T)* y2 + (quantizer / 2))) + 1;
        end;
    end;

     for k = 1:quantizer
        x(k) = floor(k - (quantizer / 2)) / y1;
     end;

     for k = 1:quantizer
        y(k) = floor(k - (quantizer / 2)) / y1;
     end;

     z= 0;
     muent = 0;
  for j = 1:quantizer
     for k= 1:quantizer
        if (s(j,k) ~= 0)
            p = s(j,k) /100000000;
            z = z + p;
            muent = muent - p * log2(p);
        end
     end
  end

  surf(x,y,s./1000000,'EdgeColor','none');
  colormap hsv
  colorbar
  hold on;
```