ABSTRACT

A SIGNAL CONDITIONING APPROACH FOR THE EXTRACTION OF THE OSCILLATORY POTENTIAL FROM THE ELECTRORETINOGRAM

by
Peter Haines Derr

The oscillatory potential (OP), a signal component of the electroretinogram (ERG), was investigated to determine correlation of the OP and pathological conditions of the inner retina. Large transients characterize the ERG. Such transients stimulate a filter's natural response. Since these responses can co-occur with the OP, a distorted OP will be extracted. A proposed signal windowing and padding technique for conditioning the ERG signal has been implemented for the extraction of a minimally distorted OP.

Windowing is used to capture only the OP period. The windowed ERG signal is then signal conditioned to generate initial values for the filter's state variables. Such correct initial conditions eliminate the perturbations created from filtering the windowed ERG. OPs were successfully extracted from a database of fifty human ERGs. The extracted OPs did not display any filter-induced oscillations and did provide some indication of the retina's pathology.

A SIGNAL CONDITIONING APPROACH FOR THE EXTRACTION OF THE OSCILLATORY POTENTIAL FROM THE ELECTRORETINOGRAM

by
Peter Haines Derr

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
In Partial Fulfillment of the Requirements for the Degree
Master of Science in Electrical Engineering

Department of Electrical and Computer Engineering

May 1999

APPROVAL PAGE

A SIGNAL CONDITIONING APPROACH FOR THE EXTRACTION OF THE OSCILLATORY POTENTIAL FROM THE ELECTRORETINOGRAM

Peter Haines Derr

Dr. Andrew U. Meyer, Thesis Advisor — Date
Professor of Electrical Engineering, New Jersey Institute of Technology

Dr. Edward J. Haupt, Committee Member — Date
Associate Professor of Psychology, Montclair State University

Dr. Timothy Chang, Committee Member — Date
Associate Professor of Electrical Engineering, New Jersey Institute of Technology

Dr. Stanley Reisman, Committee Member — Date
Professor of Electrical Engineering, New Jersey Institute of Technology

BIOGRAPHICAL SKETCH

Author: Peter Haines Derr

Degree: Master of Science in Electrical Engineering

Date: May 1999

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering
  New Jersey Institute of Technology, Newark, NJ, 1999
- Bachelor of Science in Electrical Engineering
  New Jersey Institute of Technology, Newark, NJ, 1993
- Associate of Science in Engineering Science
  Bergen Community College, Paramus, NJ, 1990

Major: Electrical Engineering

Presentations and Publications:

P. Derr, A.U. Meyer, M.G. Brigell, and E.J. Haupt,
"A signal conditioning approach for extraction of the oscillatory potential from the electroretinogram," The Association for Research in Vision and Ophthalmology (ARVO) Annual Meeting, Fort Lauderdale, Florida, May 1998.

E.J. Haupt, P. Derr, A.U. Meyer and M.G. Brigell
"Model fitting for a frequency parameter in signal conditioned oscillatory potential from electroretinograms of CRVO eyes," The Association for Research in Vision and Ophthalmology (ARVO) Annual Meeting, Fort Lauderdale, Florida, May 1998.

To my loving and supportive wife, Francine S. Derr.

ACKNOWLEDGMENT

I would like to express my deepest appreciation to Drs. Andrew U. Meyer and Edward J. Haupt for introducing me to the field of retina electrophysiology and encouraging me to pursue research in this field. Through their mentoring and guidance, I am able to propose a technique which may aid in future research in identifying the source of the oscillatory potential and the pathological ramification of the degradation of such. I would also extend deep respect and appreciation for the previous work that Drs. Meyer and Haupt have accomplished and without such there would not have been the motivation to pursue the present research. I wish them health and success in their future endeavors and look forward to the opportunity to work with them in the future.

TABLE OF CONTENTS

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

CHAPTER 1

INTRODUCTION

1.1 Objective

The electroretinogram (ERG) has been recognized as a diagnostic tool for the identification of retinal disease for many years. The ERG is primarily useful for diagnosing problems of the outer retina, i.e., the receptor cell layer and the retinal pigment epithelium. However, some specific components of the ERG signal may merit consideration as a prognostic and or diagnostic tool for diseases of the inner retina. The focus of this research will be the oscillatory potential (OP), which has been studied for more than thirty years as a possible diagnostic tool for diabetic retinopathy, with a technique being proposed for filtering the ERG to extract a minimally distorted OP from the ERG. Diseases of the inner retina, which may be detected by the OP, include diabetes, in which retinal damage can be calamitous and which has few obvious indicators in the slow course of this disease.

The medical and electrophysiology professions hypothesize that a correlation exists between the existence/quality of the OP and diseases of the inner retina, such as Central Retina Vein Occlusion (CRVO) and diabetic retinopathy. Since the inner retina is nourished via the central artery and the central vein, any deterioration of the vein's function will affect the systems responsible for the transformation of photon energy to electrical signals. To understand the possible significance of the OP, a basic summary of the retina's anatomy and a brief history of the ERG is necessary.

1.2 Background Information

1.2.1 Basic Anatomy of the Retina

The retina is a curved physiological system about 200 to 250 micrometers thick lining one half of the inside of the eye [5], [15] (see figure 1.0 for a general anatomy of the eye and the location of the retina). The retina, comprising of light sensitive cells [15], converts photon energy to multiplexed electrical signals, which are sent to the lateral geniculate nucleus [15]. The lateral geniculate nucleus accepts inputs from the optic nerve and sends fibers to the cortical receiving area for vision [3].

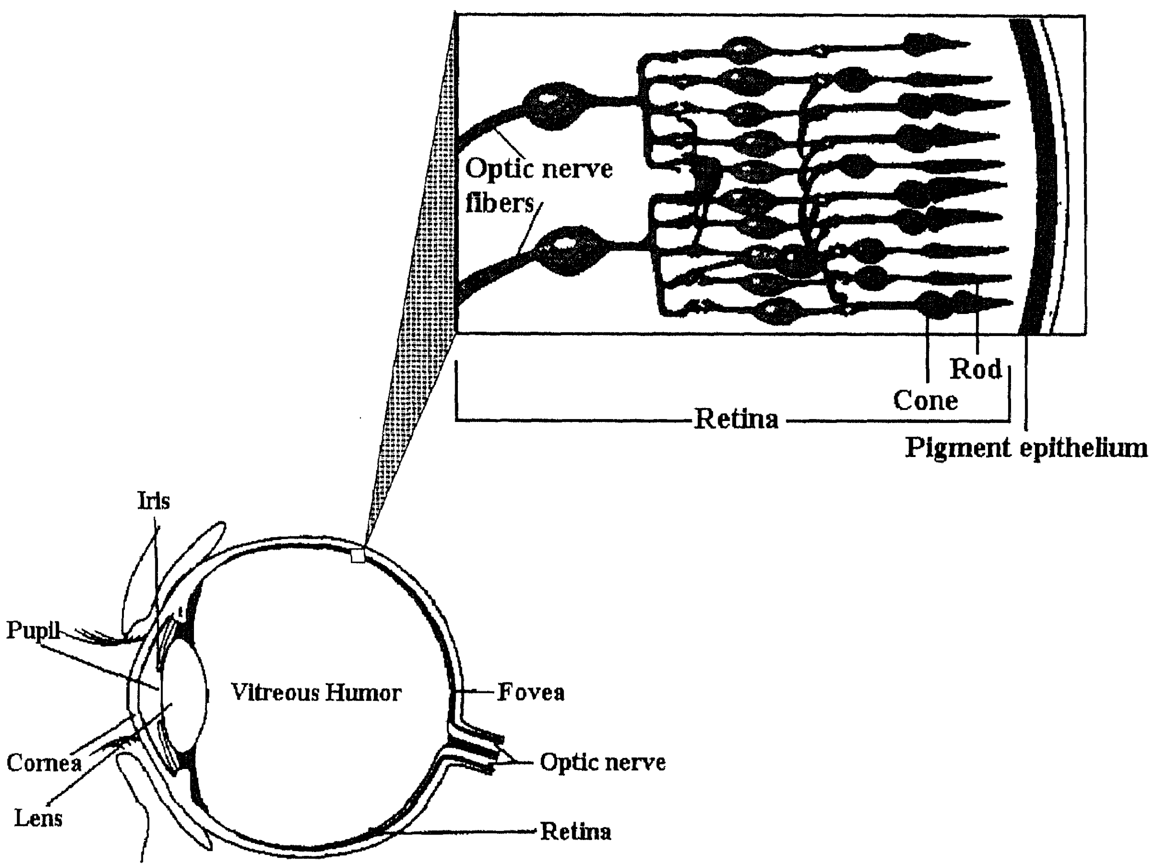


Figure 1.0 - Basic anatomy of the human eye (Adapted from E. Bruce Goldstein, *Sensation & Perception*, fifth edition, Brooks/Cole Publishing Company, New York, NY., 1999, pp. 29-70)

The brain interprets these electrical signals to provide visual perception [3]. The description of the retina can be simplified as being comprised of five neuron* layers. Referring to figure 1.1, the layers are: 1) rod and cone receptors, 2) horizontal cells; 3) bipolar cells; 4) amacrine cells and the 5) ganglion cells [3].

The photons enter the eye, passing through the translucent neurons of the retina, where they are reflected off of the pigment epithelium (see figure 1.0). The pigment epithelium provides the nutrients to the receptor neurons and also contains melanin, which prevents the scattering of light within the retina [3]. The rod and cone photoreceptors are oriented toward the pigment epithelium to intercept the reflected photons [3], [14]. The photoreceptors are excited by the isomerization** of the pigment molecule caused by the impact of a photon [3]. The photoreceptors hyperpolarize during illumination***. The receptors respond identically to a spot (100 μm in diameter) or an annulus (250 μm to 500 μm in diameter) stimulus [2], indicating that the individual receptor's response is stimulated by a single photon at a time. The photoreceptors synapse to the horizontal cells in the outer plexiform layer. A horizontal cell provides a summation of a wide field of photoreceptor responses and has a large hyperpolarizing response [2]. The horizontal cells appear to provide negative feedback to the individual photoreceptors, controlling spatial contrast and color properties of the cones by causing selected photoreceptors to depolarize[2].

---

* a neuron is a cell that generates and transmits electrical impulses.
** isomerization is defined as the pigment molecule's change in shape in response to the absorption of a quantum of light [3].
*** A reduction of a cell's membrane potential is termed depolarization as opposed to hyperpolarization or an increase in the cell membrane's potential.

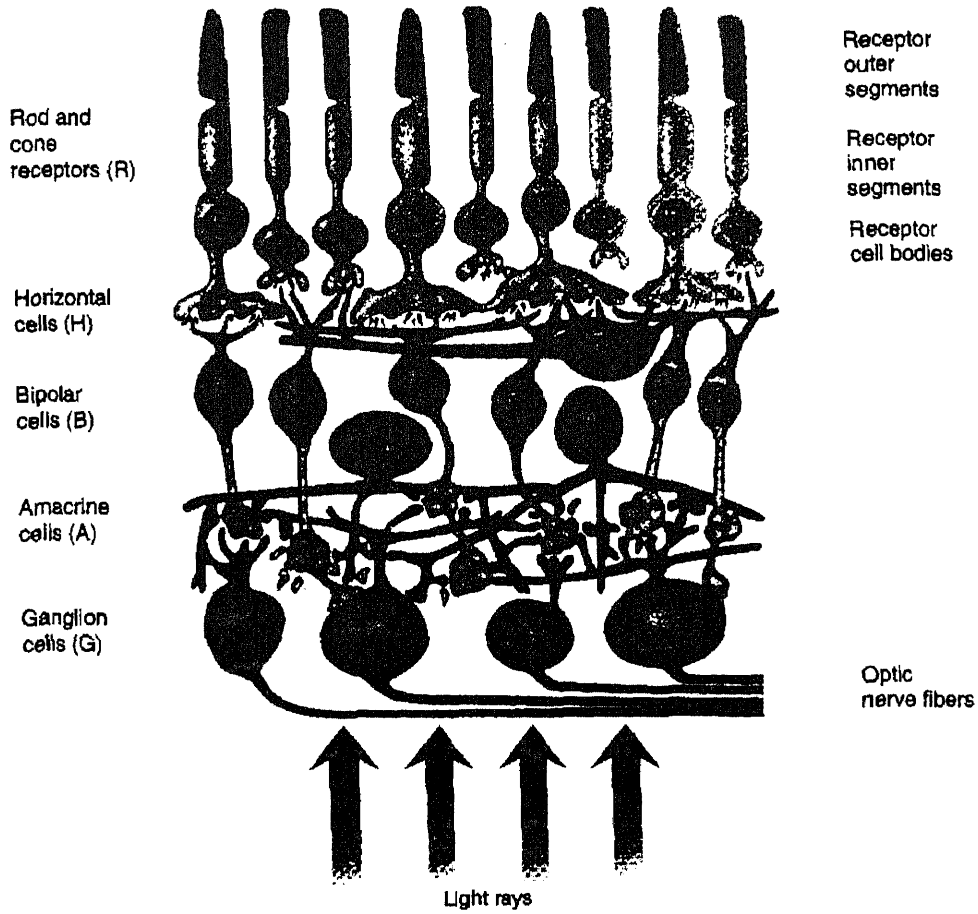


Figure 1.1 – Basic anatomy of the human retina. (From E. Bruce Goldstein, *Sensation & Perception*, fifth edition, Brooks/Cole Publishing Company, New York, NY., 1999, pp. 29-70)

The retinal neural network is comprised of the bipolar, amacrine and ganglion neurons. The cell bodies of bipolar and amacrine neurons are spatially located within the inner nuclear layer while the ganglion neurons are in the ganglion cell layer. Multiple rod and cones axons synapse with the bipolar dendrites (cone bipolars and rod bipolars) in the outer plexiform layer. The bipolar neurons respond to a mixture of spot and annuli stimulation [2]. The bipolar cells initiate the on and off channels as a result of their hyperpolarizing or, add responses before depolarizing [18]. The bipolar neurons synapse to both the ganglion cells and the amacrine cells [2],[18]. The amacrine neurons has multiple bipolar axons synapsed to its dendrite in the inner plexiform layer. The amacrine neurons provides on/off channel responses to illumination any where in their receptive fields [2]. The amacrine cells may provide directional and motion information to the ganglion cells [2]. The amacrine neuron also appears to provide depolarizing negative feedback to the bipolar neurons [2]. There are two types of ganglion neurons. The first type receives inputs from multiple synapsed bipolar neurons. The ganglion neuron is depolarized during spot illumination while annuli stimulus inhibits the depolarization [2]. Multiple amacrine neurons are synapsed in the inner plexiform layer to the second type of ganglion neuron [2]. In general, the ganglion neurons provide information about both plexiform layers to the brain via the optic nerve fibers [3]. The ganglion cells generates three types of responses. The first is an "On" response. When a light stimulus is applied, the ganglion cell responds with a rapid burst of impulses and then decays to a slower discharge. The second, "On-Off", response to light stimulus generates a burst of impulses and stops completely. The third, "Off", response of rapid impulses occurs when the light stimulus is turned off [2]. The synapses of the bipolar, amacrine and ganglion neurons can be inhibitory, excitatory or a combination of both [3]. These synapses appears to provide a weighting function to the converging electrical responses to multiple photon excitation [3]. This weighting guarantees that a spot stimulation in close proximity to an annuli stimulation will be perceived [3].

The two diseases of interest in this study, which affect the inner retina neurons, ie, the bipolar, amacrine and ganglion cells, are Diabetic Retinopathy and Central Retinal Vein Occlusion. Diabetic retinopathy is the constriction and ultimately the death of some of the small blood vessels of the inner retina [14]. The remaining blood vessels may hemorrhage, leaking blood into the retina and causing a permanent reduction in sharpness of vision [14]. In some pathological cases neovascularization may occur, which results in fragile new blood vessels growing within the retina and ultimately leaking blood into the vitreous humor [14] (see figure 1.0). Central Retinal Vein Occlusion (CRVO) is the blockage of the central vein, which returns blood supplied to the inner retina cells, or one of its branches by a blood clot [14]. As a result of the clot, blood leaks out of the blocked vein and passes into the vitreous humor causing blurred or loss of vision.

1.2.2 Electroretinogram

Frithiof Holmgren's discovery of the ERG (see figure 3.3 for the ERG of a non-diagnosed eye) in 1866 has provided the ophthalmology field with a powerful tool for observing a retina's response to light stimulus [1]. The prior year Holmgren observed an alteration in the electrical potential when light fell on the retina [2]. In 1877, Dewar recorded the first human ERG [2], thus creating the necessity to decompose the ERG into primary signal components. The first real attempts at decomposition of the ERG was in the 1930s by Ragnar Granit.

Granit identified three components of the ERG, which disappeared successively from a cat ERG. He termed the processes P-I, P-II and P-III, and considered the ERG as a summation of the processes (see figure 1.2) [2]. Following Granit, Eithoven and Jolly derived four sub-signals of the ERG (see figure 1.3). When subjecting a dark-adapted eye to white light, they observed the following sequentially occurring components: 1) a

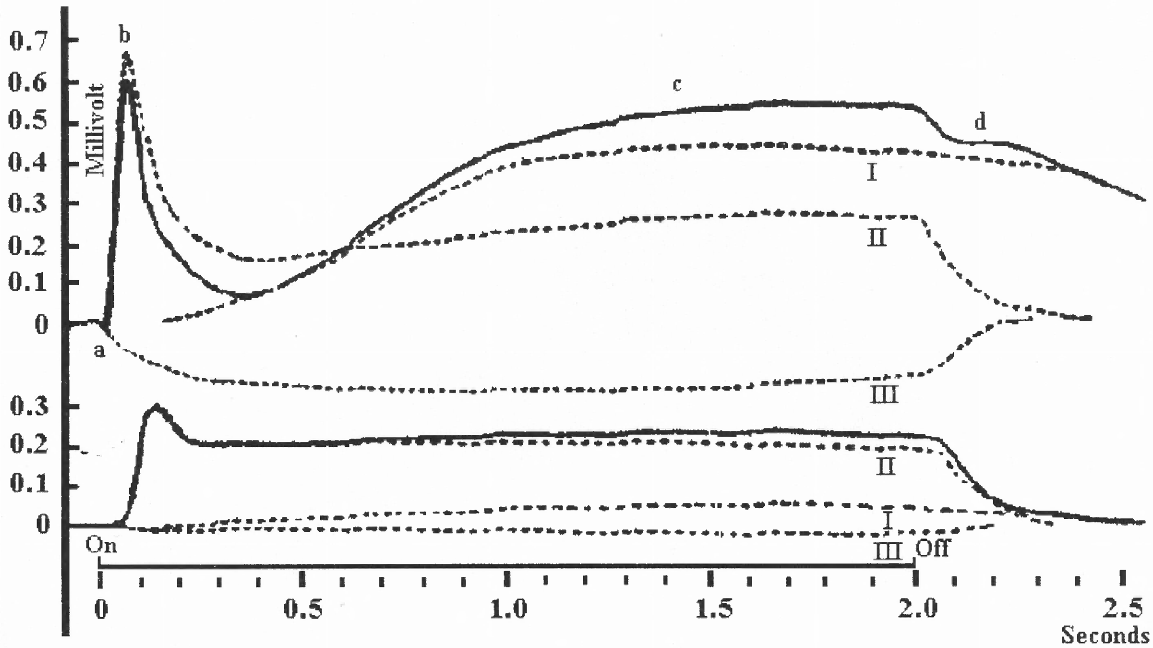


Figure 1.2 - Granit identified three components of the ERG, which disappeared successively from a cat ERG. He termed the processes P-I, P-II and P-III, and considered the ERG as a summation of the process. The top plot had a stimulus ten times larger than the bottom plot. (Adapted from F.H. Adler, *Physiology of the Eye*, The C.V. Mosby Company, Saint Louis, Mo., 1975, pp. 453-499).

cornea negative a-wave; 2) a cornea positive b-wave; 3) a slower cornea positive c-wave; and 4) a d-wave, which coincided with the elimination of stimuli in some mammals [2]. With the identification of the prime components of the ERG signal, the origin of the components had to be investigated.

Tomita observed that components of the ERG could be localized within the frog retina, by depth recording utilizing microelectrodes. He also obtained similar waveforms from corneal recordings [2]. Brown and Wiesel determined that the different components of the ERG could be localized to different layers of the retina. They also determined that the a-wave is generated from a location different from the b-wave's source [2]. Brown and Watanbe found that a large a-wave could be obtained from the photoreceptor

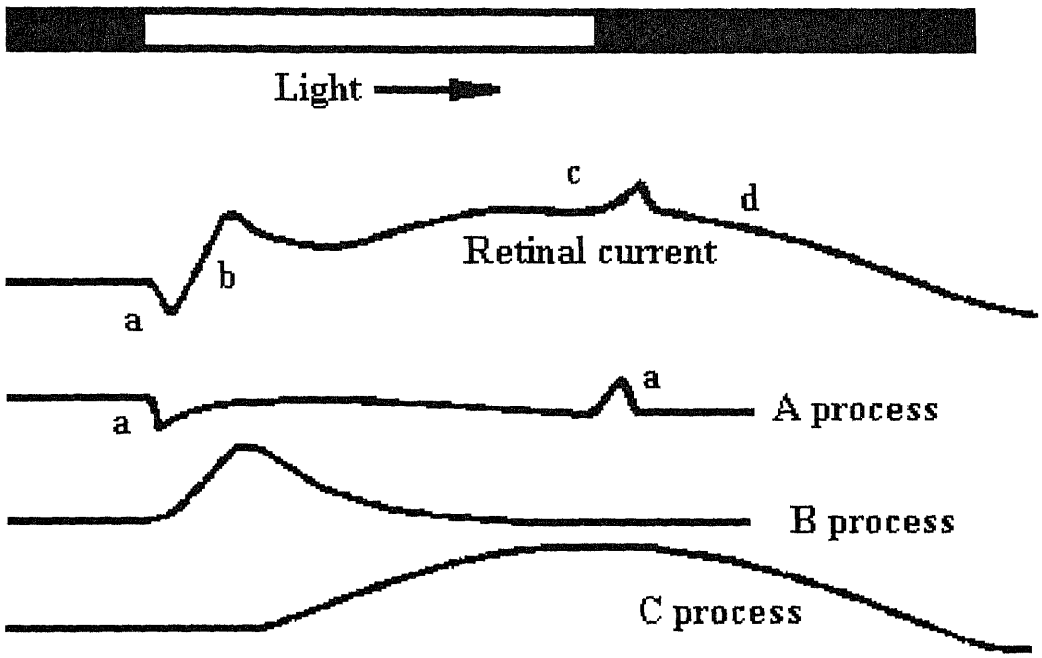


Figure 1.3 - Eithoven and Jolly derived four sub-signals of the ERG. When subjecting a dark-adapted eye to white light, they observed the following sequentially occurring components: 1) a cornea negative a-wave; 2) a cornea positive b-wave; 3) a slower cornea positive c-wave; and 4) a d-wave, which coincided with the elimination of stimuli in some mammals. (Adapted from F.H. Adler, *Physiology of the Eye*, The C.V. Mosby Company, Saint Louis, Mo., 1975, pp. 453-499)

abundant foveal of the monkey. A large b-wave with a relatively smaller a-wave could be found in the inner nuclear prominent peripheral retina [2]. Murakami and Kameko, using Granit's terminology of the ERG's three processes, subdivided the P-III process into a distal and a proximal component. The distal P-III was isolated to the photoreceptor layer and the proximal was isolated in the inner nuclear layer. The distal P-III is the a-wave, however the proximal P-III, even though thought to be the b-wave, has not been observed in all mammals [2]. With the a-wave's generation being isolated to the photoreceptor layer, the photoreceptor layer is subdivided into rod and cone systems.

Armington, Johnson and Riggs recorded the a-wave on the cornea of a dark-adapted human eye (see figure 1.4). The a-wave showed the summation of the photopic response of the cones and the scotopic response of the rods. The cone spectral sensitivity, a_p, was found be unaffected by dark-adaptation, while the rod spectral sensitivity, a_s, was observed to have a significant attenuation and latency after dark-adaptation [2]. The spectral sensitivity responses of the dark-adapted photoreceptors are plotted in a "Dark adaptation curve" (see figure 1.5) [3]. The plot verifies that the photoreceptors comprise of two cell types, the rods and cones [3].

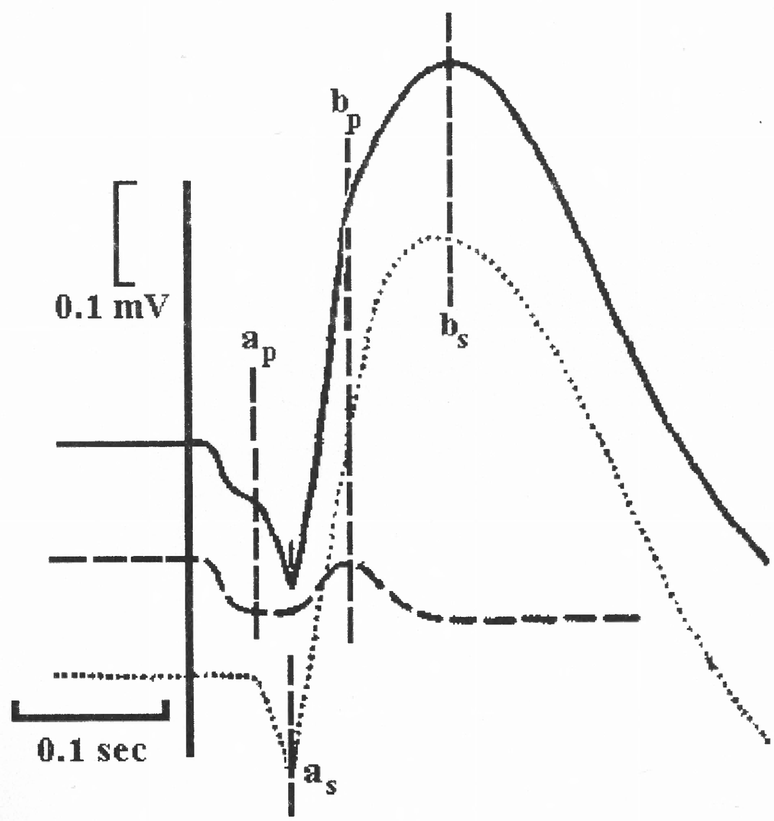


Figure 1.4 - Armington, Johnson and Riggs recorded the a-wave on the cornea of a dark-adapted human eye. The a-wave showed the summation of the photopic response of the cones and the scotopic response of the rods. They further observed that the a-wave and b-wave components of the cornea measured ERG of the dark-adapted human eye contained photopic and scotopic a and b waves. The top curve is the photopic response, while the bottom curve is the scotopic response. (Adapted from F.H. Adler, *Physiology of the Eye*, The C.V. Mosby Company, Saint Louis, Mo., 1975, pp. 453-499)

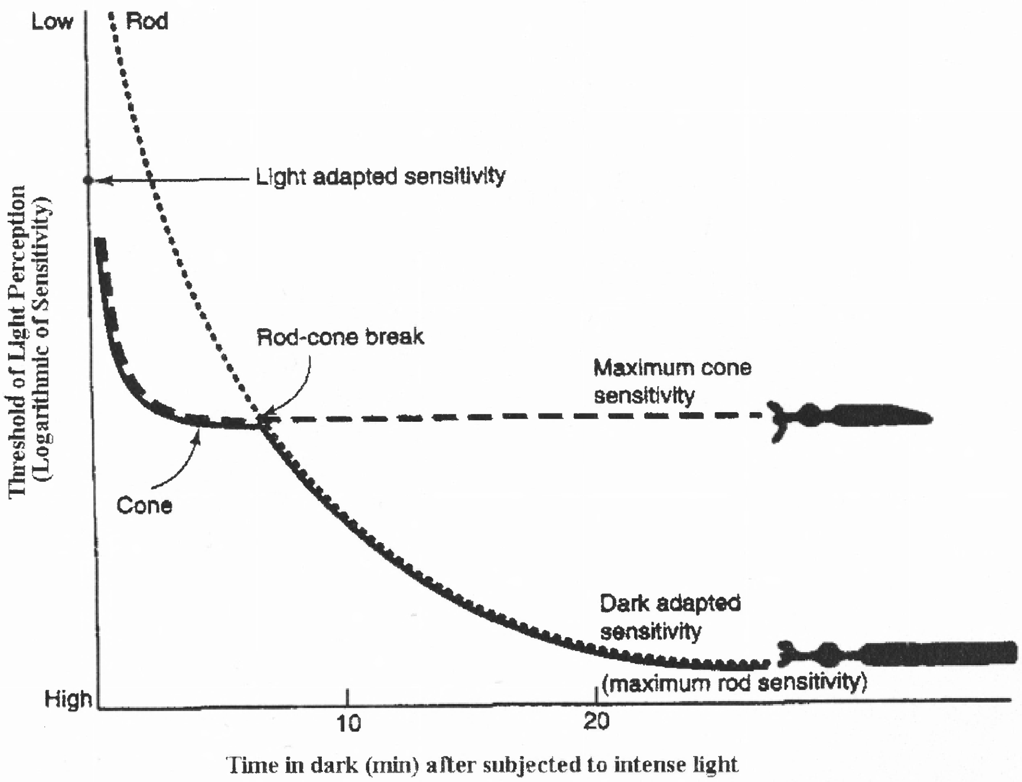


Figure 1.5 - The "Dark adaptation curve" illustrates evidence that the retina's photoreceptor are comprised of two cell systems (rods and cones). The eye is subjected to an intense light. After which it is dark adapted for a specific time. At that time, the eye is subjected to a manually increasing point light source until the patient is able to detect the light. That point is the threshold of light perception. The plot contains the rod and the cone spectral sensitivities and the summation of both. [3]. (Adapted from E. Bruce Goldstein, *Sensation & Perception*, fifth edition, Brooks/Cole Publishing Company, New York, NY., 1999, pp. 29-70)

This differentiation of rod and cone responses is useful in tailoring the ERGs for obtaining the response of a specific receptor of the retina. Using a light adaptation of 1.60 log cd/m^2* and a white xenon flash stimuli of 0.85 log cd s/m^2* [**4**], Peachey

---

* The luminance level is defined as candela per square meter (cd/m^2). The time-integrated luminance for brief flashes of light is defined as candela-seconds per square meter ($cd\text{-}s/m^2$). Therefore, 1.60 log cd/m^2 = 39.81 cd/m^2 [19]. One candela radiates 4π lumens of light flux. One lumen is defined as a one candela point light source illuminating a 1 ft^2 of a sphere with a 1 ft radius.

tailored a mouse ERG to measure the response of the cone system [4]. Armington, Johnson and Riggs further observed that the a-wave and b-wave components of the cornea measured ERG of the dark-adapted human eye contained photopic and scotopic a and b waves (see figure 1.4) [3]. Both the photopic and scotopic b-waves are cornea positive signals; however the scotopic b-wave contains a significant larger amount of spectral energy than the photopic b-wave. It can be surmised that the a-wave is generated in the photoreceptors. The location of the b-wave origin has not been clearly isolated. However, the photoreceptors are probably not the source. The ERGs of retinas with central artery occlusion contain an a-wave but the b-wave is not present [2]. The photoreceptors are not affected but the disease destroys the inner nuclear layer [2].

Even though the b-wave is not generated in the photoreceptors, the photoreceptors strongly influence the profile of the b-wave in peak amplitude and latency. Adrian observed that stimuli (photopic) that favored cone response produced small b-waves with a short latency where as stimuli (scotopic) that favored rod response produced large cornea positive b-waves with a longer latency. The photoreceptors are made up of cones and rods; however both the rods and cones can be sub-divided into spectral sensitive specific rods and cones. Stiles and Wald identified three different cone mechanisms, each with a specific spectral sensitivity curve, that could be separated from the cynomologus monkey's b-wave [2].

Muller and Dowling, using a mudpuppy ERG, showed that müller cells* have evidence to be the site of the generation of the b-wave. Using microelectrodes, they

* The müller cell is a macroglial cell, which spans the entire thickness of the retina. It forms the inner limiting membrane of vitreal surface and the outer limiting membrane at the photorecptor's inner segments [17].

measured across the cells of the inner nuclear layer, including the horizontal, bipolar, amacrine and müller cells. The müller cell response closely matched the b-wave. An a-wave was not present in the measurement [2]. The c-wave appears to begin at the occurrence of the peak amplitude of the b-wave and has been measured at the pigment epithelium.

Steinberg et al measured the c-wave intercellularly from the pigment epithelium of a cat and Noell reduced the amplitude of the c-wave of a rabbit by severely damaging the pigment epithelium using sodium iodate [2]. The generation location of the c-wave is not specific, but the pigment epithelium appears to have significant importance to its existence. Dowling and Riggs experimented on an all-rod skate retina; suspended in an eyecup. Applying sodium aspartate, they suppressed the proximal response (b-wave) without affecting the a and c waves. Leaving the pigment epithelium, the retina was removed from the eyecup. The c-wave vanished. They concluded that the pigment epithelium must be present for the c-wave to exist. However, the dark-adapted c-wave has the response corresponding to the absorption spectrum of rhodopsin found in the rods and not the melanin in the pigment epithelium [2].

1.2.3 Application of the Electroretinogram

The full-field ERG is a non-invasive standardized ocular electrophysiologic test for extracting specific responses of the retina [6]. The typical responses of interest are a) dark-adapted rod response; b) maximal response of the dark adapted eye; c) oscillatory potential; d) cone response and e) response from a flicker stimulus [6]. Adhering to the International Society for Clinical Electrophysiology of Vision (ISCEV) protocol, the

ERG is administered by placing a contact lens electrode on the cornea of the eye; a reference electrode, usually part of the contact lens electrode, is contacted with the conjunctiva; and a grounding electrode is attached to the ear or forehead [6]. The retina is subjected to Ganzfeld stimulation, which guarantees equal illumination of the retina. The retina is stimulated by a combination of a luminance flash (cd-s/m^2) and background illumination (cd/m^2). The background illumination is utilized to provide a calibrated level of light adaptation. The eye is dark adapted for a period of twenty minutes to allow for the rods and cones to desensitize [6]. Dependent on the adaptation of the retina prior to a calibrated light stimulation, the response of the ERG can be rod dominant or cone dominant.

1.2.4 Oscillatory Potential

The oscillatory potential (OP) shows promise as a possible indicator of the early development of inner retinal diseases. Eyes with certain pathological conditions, such as diabetic retinopathy, appear to produce ERGs with a degraded or missing OP component, while other components of the ERG such as the a-wave and b-wave are unaffected [7].

The OP is observed as a envelope of sinusoids (see figure 3.3 for an extracted OP) riding on the positive cornea b-wave of the electroretinogram (ERG) [7], [20]. It is observed that the onset of the OP occurs when the polarity of the ERG's slope changes from negative to positive, as a result of the ERG's positive cornea b-wave dominating over the negative cornea a-wave. The OPs termination is an exponential decay. The OP appears to has frequency bandwidth between 100 and 200 Hz. Wachtmeister has documented the bandwidth to be 105 to 150 Hz [20]. Since the objective of this paper is to present a technique for extracting a minimally distorted OP from the ERG, the currently used methods for evaluating the characteristics of the OP will be addressed in Chapter 2, Literature Survey. However, a history of the observance and evaluation of the OP is warranted.

The OP was first observed in the ERGs recorded by Riggs and Armington [7]. But Cobb and Morton provided the first attempt at describing the OP found in the human ERG [7],[20]. They observed "... 4-6 rhythmic wavelets having an interpeak interval of about 7 msec" [7]. Yonemura et al were the first to term the "wavelets" as oscillatory potentials and postulated that the OPs were a separable component of the ERG based on the observance that the OP was degraded or missing while the a and b-waves were unchanged in certain pathological conditions [7], [20].

The generation of the OP is highly dependent on the level of light stimulation and background adaptation The OP with the largest energy level was observed to be generated under mesopic adaptation conditions and rapid flash stimulation [20]. When the time interval between flashes increased, the OP's energy content decreased.[20].

Dawson and Stewart brought up an important observation that cautioned the use of the term of oscillatory potential, since the signal was not a true oscillation. They also observed that the variability of the OPs characteristics is highly dependent on the test conditions in which the ERG was obtained [7]. Adams and Dawson showed that oscillatory potentials are produced under mesopic conditions whereas under photopic and scotopic conditions, oscillatory potentials had reduced amplitudes or were not observed [9]. Not to take Adams and Dawson out of context, they were evaluating the "Fast Retinal Potential", which is an umbrella term which includes the oscillatory potential [9].

The literature supports that the OP is dependent on retinal circulation, which eliminates the photoreceptors and horizontal cells as sources [8],[20]. Initially Brown localized the OP to the inner retinal layers [20]. Yonemura and Hatta found the largest OPs in the region of the bipolar cells [20]. Using depth recording and principal component analysis, Heynen et al suggest that a feasible source of the OP is the bipolar cells[8]. The bipolar cells significantly spans the retina laterally and in depth to support the resulting "current-source density profiles" [8].

Since the OPs are elicited by mesopic conditions, which is a mixture of photopic and scotopic processes, the responses of the rod and cone may be part of the OP generation process [20]. This further supports the generation of the OPs by the bipolar cells, since they are stimulated by both the rods and cones.

CHAPTER 2

LITERATURE SURVEY

A survey of the ophthalmologic literature on the oscillatory potential (OP) resulted in numerous papers supporting the hypothesis of using the OP as an indicator of circulatory diseases of the inner retina [7]. Even though mentioned as a concern, filtering techniques for extracting a minimally corrupt OP from the ERG have not been the objective of the surveyed literature.

The standards for ophthalmologic electrophysiological measurements are established by ISCEV [6]. Presently, ISCEV's protocol for the filtering of the ERG to extract the OP is limited to defining the bandwidth of a bandpass filter [75 to 300Hz]. However, ISCEV does caution the clinician to test for corrupting artifacts and filter-induced ringing [6].

The surveyed literature suggests three methods for evaluating the OP: 1) measuring the peak amplitudes of each OP peak relative to a baseline [7]; 2) evaluating the ERG in the frequency spectrum, i.e. measuring the spectral power of the ERG [7],[10],[11]; and 3) evaluating the OP in the time domain [12].

The first method uses the "oscillatory index" or "caliper-square method" [7]. This method sums the first order approximation of each oscillation peak. The baseline traverses the troughs of each oscillation [7]. Error can be induced as a result of the measured amplitude of the OP peaks being subjected to the slope of the b-wave [7].

The second method, rationalized by Speros et al [7] and discussed by Van der Torren et al [11] and Wachtmeister [20], incorporates Fourier analysis of the ERG in the frequency domain. The method attempts to obtain a benchmark of the power density of the frequency spectrum of an ERG without an OP and comparing the result to an ERG containing an OP. Van der Torren et al illustrates the method [11],[12]. The time domain ERG of an eye without arterial occlusion is transposed into the frequency domain. Twelve hours after an arterial occlusion, another ERG is performed on the patient's eye and the ERG is transposed into the frequency domain. Two weeks after the occlusion, another ERG is taken on the eye and again transposed into the frequency domain. The frequency spectrum of the second ERG is subtracted from the first ERG. The residual is suggested to be the spectral energy of the OP, without the a-wave component. The artery of a normal patient cannot be blocked, therefore, this method cannot be used for human testing. Wachtmeister used Fourier analysis to measure the energy content and the primary frequency component of the OP [20].

The third method was proposed by Meyer et al [12]. The method suggests the use of a time domain OP signal model (see figure 4.2), which attempts to describe the dynamics of the OP signal [12] in a finite set of parameters which can be evaluated statistically. The OP is extracted from the ERG using a bilateral filter which induces zero phase shift. Haupt et al [13] performed a statistical study on twenty five patients diagnosed with CRVO eyes. The residual filtering technique, smoothes out a large portion of the distorting artifacts of the ERG. Prior to bilateral bandpass filtering, the ERG signal is conditioned by subtracting a bilateral lowpass filtered ERG signal from the original ERG; creating a residual signal. Meyer's single envelope OP model [12] was fitted to a windowed segment of the bandpass filtered ERG signal. The use of this method is subjected to limitations of MATLAB's bilateral filter algorithm, filtfilt.m..

CHAPTER 3

MATERIAL AND METHODS

3.1 Electroretinogram Data

Dr. Mitchell G. Brigell of Loyola University of Chicago supplied 50 electroretinogram (ERG) data files, obtained from 25 patients, with one normal (fellow) eye and one diagnosed eye, (see appendix A1 for file format). The latter eyes had Diabetic Retinopathy, of which some had progressed to completion of neovascularization and the remaining eyes remained unchanged. The ERG data used in this study contained large flash artifacts, possible attributed to the method of measuring the ERG, occurring instantaneously after the flash stimulus. The flash artifact can possibly be attributed to method in which ERG was obtained. The artifact obscures any possible negative cornea a-wave. However, the positive cornea b-wave appears to be unaffected.

3.2 Filtering Method

To extract the OP, the ERG was filtered using a bandpass bilateral Butterworth digital filter, with 80 dB/decade rolloff. The bandwidth of the filter was chosen to attenuate the a and b wave components of the ERG. The bandpass used in this study was 65 to 250 hertz. The filter was constructed with two cascaded 4^{th} order digital bandpass filters. Each digital filter was a Direct Form Type II with a realization illustrated in Figure 3.0. To accomplish the bilateral function, which eliminates phase delay, the signal y(k) was reversed and re-filtered with the second bandpass filter. After re-filtering, the signal was reversed to the correct direction.

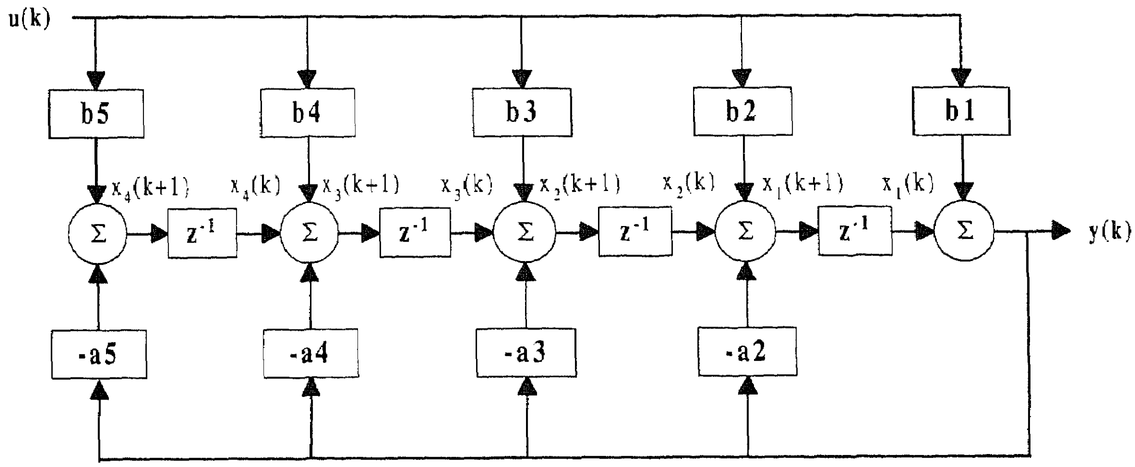


Figure 3.0 - 65 to 250 hertz Direct Form Type II bandpass digital filter with 4th order butterworth coefficients applied. The coefficients are calculated using MATLAB's butter.m function. b=[0.0591 , 0 , -0.1183 , 0 , 0.0591] and a=[1 , -2.9433 , 3.4358 , -1.9157, 0.4407]. u(k) is the input signal and y(k) is the unilateral filtered output signal. Reference Appendix A2 for a state space model derivation and the state variable values in response to a unit impulse.

3.3 Signal Conditioning

When bandpass filtering the ERG, for the extraction of the OP, care must be taken in the process to minimize the filter from generating a non-OP response as a result of the flash artifact and fast changes in the slope of the ERG. Figure 3.1 illustrates how a single stage of the filter's natural response is obtained from a unit impulse stimulus. For example, the typical bandpass filter response to an impulse may be of the form,

$$V_{out}(t) = e^{-\alpha t}(B_1 \cos(2\pi f t) + B_2 \sin(2\pi f t)) \qquad (3.1)$$

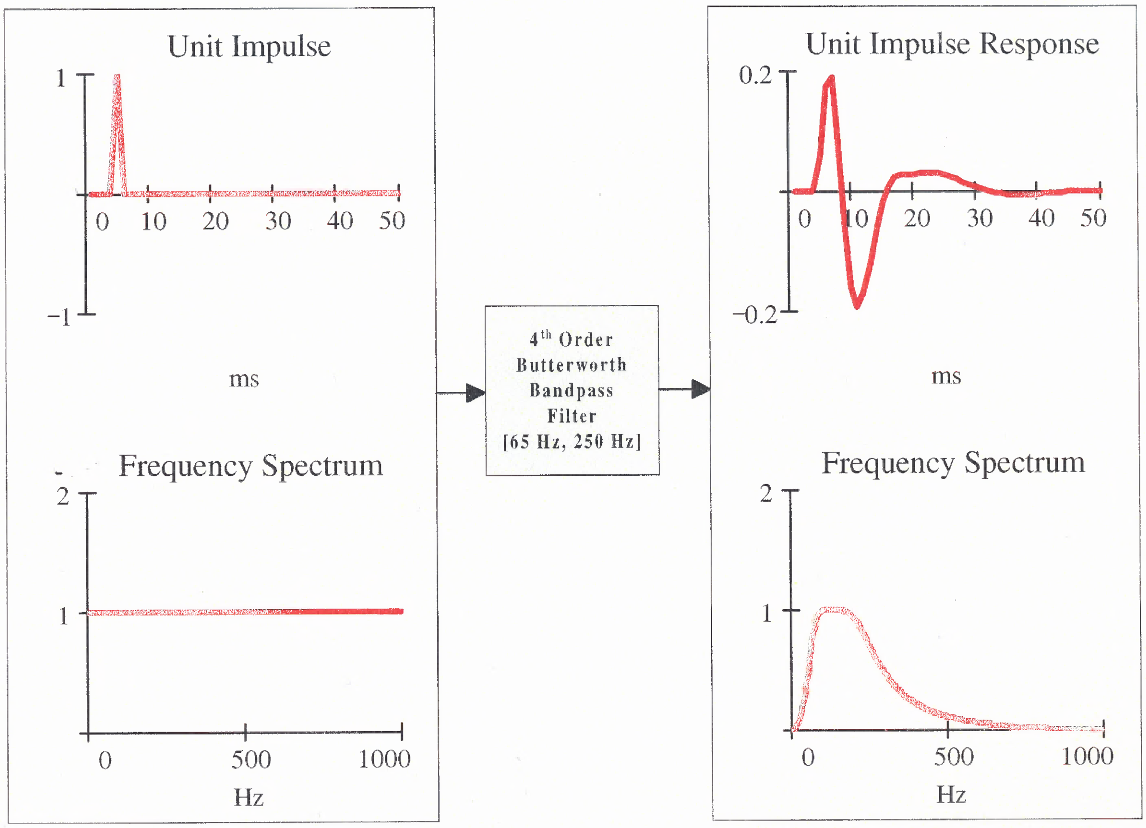


Figure 3.1 - If a system is excited by an impulsive source, the resulting output is the natural response of the system. When a bandpass filter, [65 and 250 Hz] is excited by an impulse, the expected response of the filter is generated.

Parameter α is the system's rate of decay. B_1 and B_2 are the initial values of $V_{out}(t)$ and $dV_{out}(t)/dt$,respectively. The dynamics of the filter determines the natural response. The filter's dynamics are defined by the time required for the filter's state variables to reach a steady state value. For the impulse response, the steady state value is zero. For the second order bandpass filter, the dynamics of the four state variables are illustrated in Figure 3.2. Reference Appendix A2 for the state space model of the filter and the values of the system's state variables after being stimulated by an impulse.

The transient response of the filter, which unfortunately by nature is of similar form and bandwidth as the OP, is superimposed on the extracted OP. Since these non-OP transients may occur over the temporal epoch of the OP, a distorted OP may be extracted. Fortunately, the major impulsive sources are outside the temporal epoch of the OP. A technique comprising signal windowing and conditioning has been utilized. The ERG signal is first windowed in the time domain and the resulting signal is conditioned. The windowing removes the impulsive sources and conditioning corrects for the units steps generated by windowing. Figure 3.3 illustrates an example of a fellow eye's ERG, a poorly extracted OP and a properly extracted OP.

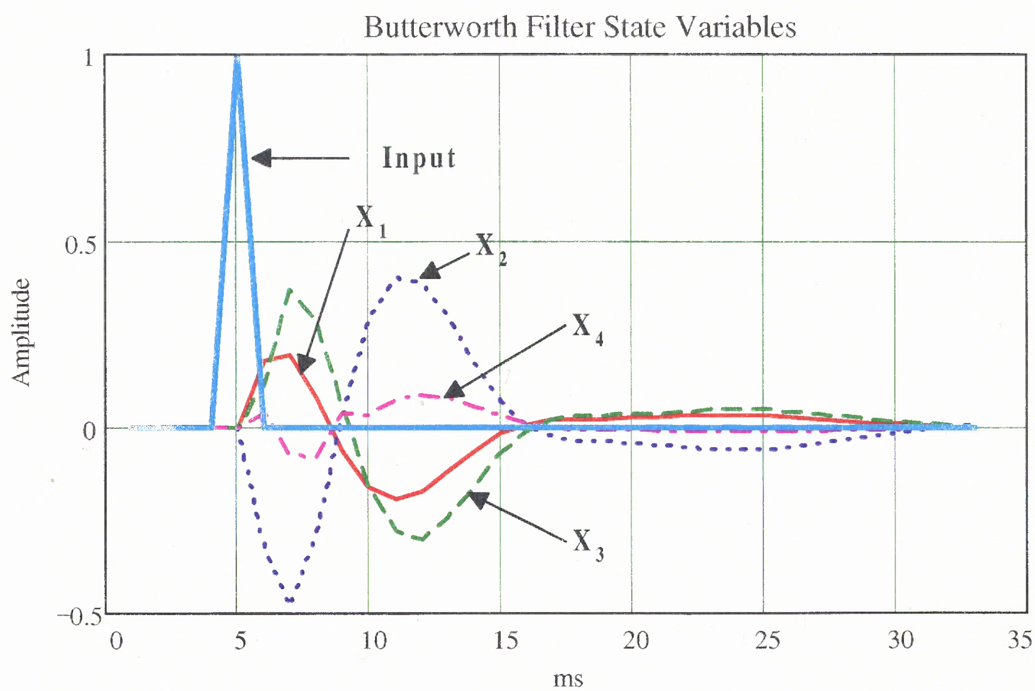


Figure 3.2 – The response of the filter's individual state variables to a unit impulse stimulus. Each variable contains a finite amount of energy, which must decay over a finite period of time. When all the states have reached a steady state, the filter, as a system, will also reach steady state.

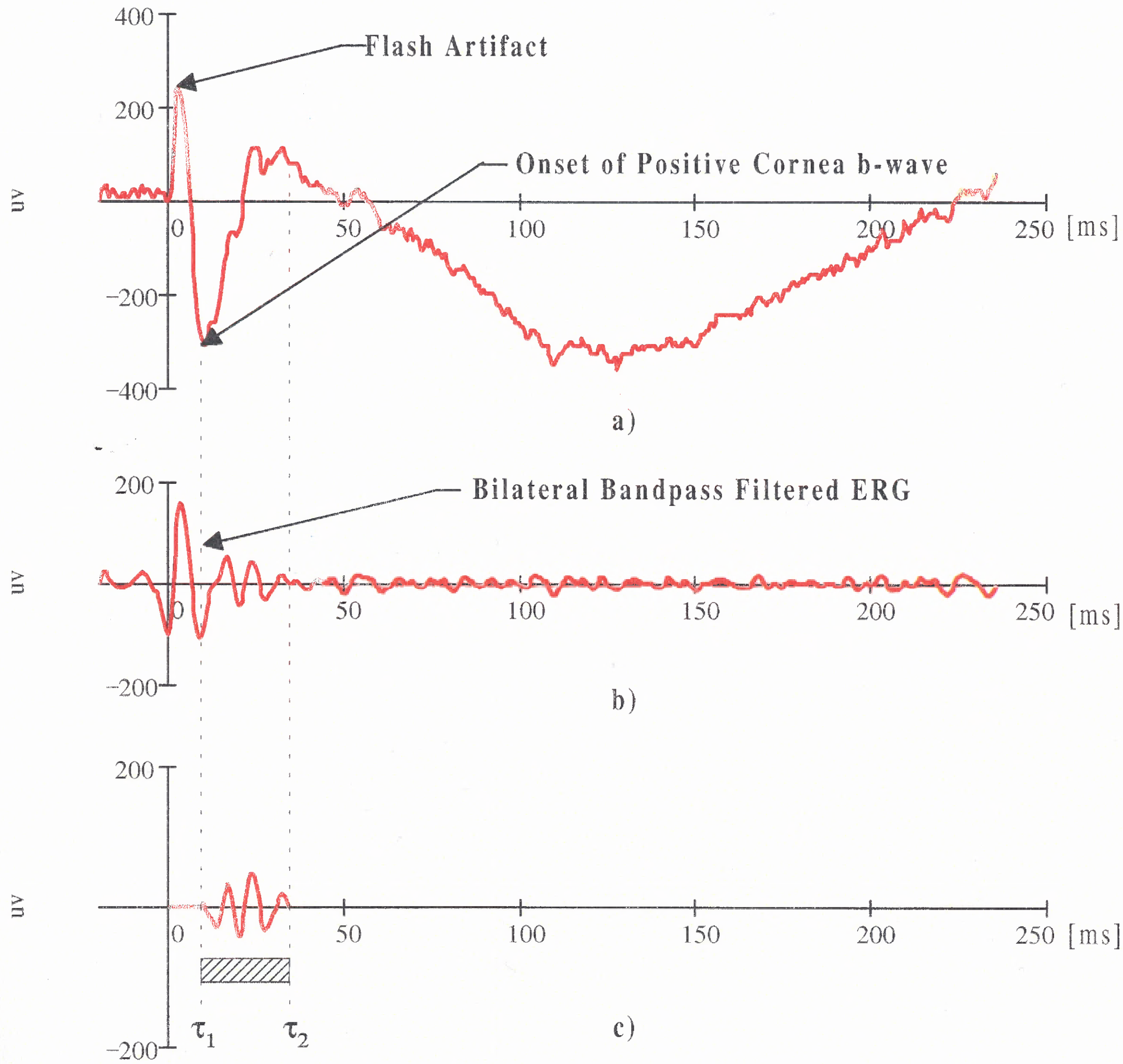


Figure 3.3 -a) An ERG of a fellow eye with the flash artifact and the onset of the positive cornea b-wave identified. **b)** The result of bilateral bandpass filtering the ERG, without signal conditioning the ERG to remove the large flash artifact. The OP is not discernable within the period of $\tau_1 \leq t \leq \tau_2$. **c)** The result of bilateral bandpass filtering after the proposed signal conditioning technique is applied to the ERG. The OP is clearly identifiable within the period of $\tau_1 \leq t \leq \tau_2$.

3.3.1 Windowing

The OP, which is of similar form and bandwidth of natural response of the filter, must be sectioned from the ERG. If the filter's natural response is present in the temporal epoch of the OP, a severe distortion in the interpretation of the statistical evaluation will occur. An aggressive windowing approach must be utilized.

Windowing has been employed to process only the portion of the ERG containing the OP. Consider the segment of the ERG illustrated in Figure 3-4 a). The region of the ERG that may contain an OP is located between τ_1 and τ_2. Windowing is chosen for the removal of all the impulse sources prior to τ_1 and the removal of the ERG signal that does not contain the OP signal for $t > \tau_2$. Since the OP is located between $\tau_1$and τ_2, parameter fitting an OP model to the extracted OP for $t > \tau_2$ will produce statistical errors and cause unnecessary computational delay.

The windowing technique chosen is the rectangular pulse. By using a uniform pulse, all impulsive sources outside of the OP's temporal epoch are removed. Since the OP extraction employs bilateral filtering, impulsive sources pre and post OP must be eliminated.

The windowing signal, W(t), multiplies ERG(t) by unity inclusive of the range $[\tau_1, \tau_2]$ and by zero elsewhere. Multiplying ERG(t) and W(t) together will produce S(t), illustrated in Figure 3.4 c).

$$W(t) = u(\tau_1 - t) - u(\tau_2 - t) \tag{3.2}$$

$$S(t) = ERG(t) \bullet W(t) \tag{3.3}$$

The occurrence of the OP is located on the ascending b-wave[7],[20]. The onset of the observable positive cornea b-wave is termed τ_1. For the 50 eyes used in this study, τ_1

was observed to be a local minimum of the ERG signal. Therefore, this location was chosen for the application of the unit step u(t-τ_1). The local minimum is easily identified using the OP extraction algorithm (see section 3.4). For this study, the unit step -u(t-τ_2), is located at twice the OP's center time, which is well past the evidence of an OP.

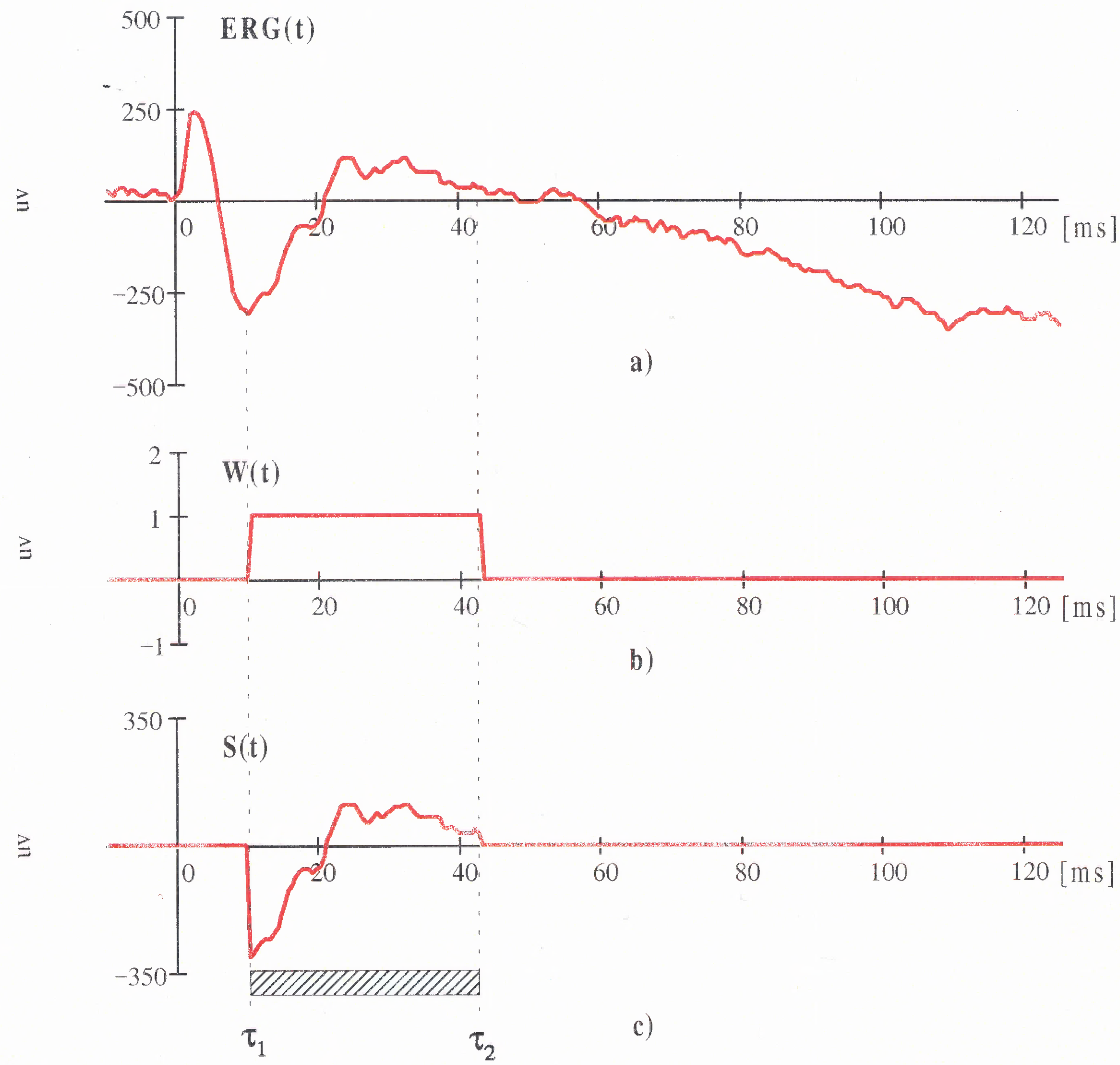


Figure 3.4 – a) The signal ERG(t), of a fellow eye, with the of location of the OP identified as occurring between $\tau_1 \leq t \leq \tau_2$. **b)** The rectangular window W(t), is assigned the value of 1 for $\tau_1 \leq t \leq \tau_2$ and 0 elsewhere. **c)** The signals ERG(t) and W(t) are multiplied together to produce signal S(t); the windowed ERG.

3.3.2 Pre filtering Signal Conditioning

The goal of pre filtering signal conditioning is to eliminate the filter's natural response resulting from the stimulation of the abrupt roll-offs of the windowed ERG signal, S(t). These abrupt roll-offs can be characterized as unit steps. Just as the impulse response is determined by the filter's dynamics, the filter's step response is also resultant to the same dynamics. Figure 3.5 illustrates the step response of a unilateral bandpass filter. The response, which may corrupt the shape of the extracted OP, may be of the form,

$$V_{out}(t) = V_{final} + e^{-\alpha t}(B_1 \cos(2\pi f t) + B_2 \sin(2\pi f t)) \tag{3.4}$$

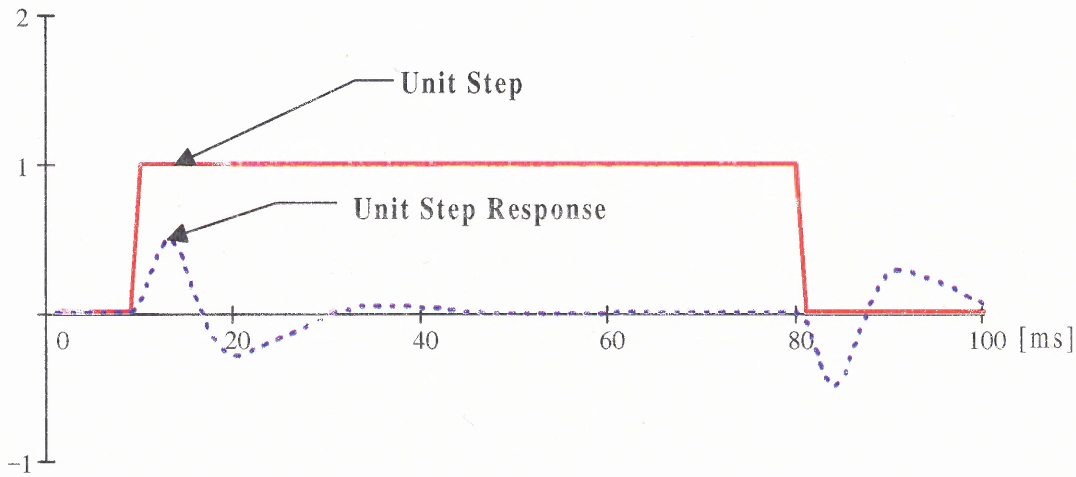


Figure 3.5 – The step response of a unilateral second order butterworth bandpass filter. Since the step response has the same form and bandwidth as the OP, the extracted OP may be distorted. The period of this example signal is typical to the period of a human ERG.

To eliminate the effect of the filter's step response, the abrupt changes in S(t) are removed by the attachment of padding signals at the discontinuities. The padding signals are designed to maintain the fundamental slope of S(t). Equation 3.5 represents the discontinuities of S(t) [see figure 3-6b],

$$S(t) = ERG(t)[u(t-\tau_1) - u(t-\tau_2)] \tag{3.5}$$

Since bilateral filtering is employed, two padding signals will need to be created, one for filtering in the forward direction and the other for filtering in the reverse direction. The two padding signals are lowpass filtered inverted mirrored images of the entire signal S(t). The images are lowpass filtered at a corner frequency of 60 Hz, which retains the bandwidth of the positive cornea b-wave.

Since the number of data points contained in the padding signals one data point less than , S(t) has to be time shifted for the left padding signal to fit between $0 \le t \le (\tau_2 - \tau_1)$. The time shifted S(t) is referred to as $S_{shift}(t)$. $S_{shift}(t)$ is defined by equation (3.6) (see figure 3-6c),

$$S_{shift}(t) = \begin{cases} 0 & 0 \le t < \tau_2 - \tau_1 \\ ERG(t + \tau_1 - \tau_2) & (\tau_2 - \tau_1) \le t \le 2(\tau_2 - \tau_1) \\ 0 & 2(\tau_2 - \tau_1) < t \le T; \quad T = duration\ of\ ERG(t) \end{cases} \tag{3.6}$$

With respect to the $S_{shift}(t)$, the unfiltered left and right padding signals are the inverted mirror images of $S_{shift}(t)$ pivoted at $(\tau_2 - \tau_1)$ and $2(\tau_2 - \tau_1)$, respectively. The left inverted mirrored image, $S_{LM}(t)$, and the right inverted mirrored image, $S_{RM}(t)$, are defined in equations 3.7 and 3.8, (see figures 3-6d and 3-6h respectively),

$$S_{LM}(t) = \begin{cases} 2S_{shift}(\tau_2 - \tau_1) - S_{shift}(2(\tau_2 - \tau_1) - t) & 0 \le t < (\tau_2 - \tau_1) \\ 0 & t \ge (\tau_2 - \tau_1) \end{cases} \tag{3.7}$$

$$S_{RM}(t) = \begin{cases} 0 & 0 \le t < 2(\tau_2 - \tau_1) \\ 2S_{shift}(2(\tau_2 - \tau_1)) - S_{shift}(4(\tau_2 - \tau_1) - t) & 2(\tau_2 - \tau_1) \le t < 3(\tau_2 - \tau_1) \\ 0 & t > 3(\tau_2 - \tau_1) \end{cases} \tag{3.8}$$

$S_{LM}(t)$ and $S_{RM}(t)$ are lowpass filtered for the extraction of the fundamental slope. The lowpass filter has the same form as the filter in figure 3.0 with the following butterworth coefficients: b=[0.0078 , 0.0156 , 0.0078] and a=[1 , -1.7347 , 0.7660]. The resulting filtered signals, equations 3.9 and 3.10, are summed with $S_{shift}(t)$ to produce

$$S_{FLM}(t) = LPF(S_{LM}(t)) \tag{3.9}$$

$$S_{FRM}(t) = LPF(S_{RM}(t)) \tag{3.10}$$

the conditioned windowed ERG signal, $S_{cond}(t)$, and is defined by equation 3.11. $S_{cond}(t)$

$$S_{cond}(t) = \begin{cases} S_{FLM}(t) & 0 \le t < (\tau_2 - \tau_1) \\ S_{shift}(t) & (\tau_2 - \tau_1) \le t \le 2(\tau_2 - \tau_1) \\ S_{FRM}(t) & 2(\tau_2 - \tau_1) < t \le 3(\tau_2 - \tau_1) \\ 0 & 3(\tau_2 - \tau_1) < t \le T; \quad T = Period\ of\ ERG(t) \end{cases} \tag{3.11}$$

is filtered using the filter described in section 3.2. The resulting signal must have the filtered padding signals removed and be time shifted back to S(t) original location. The signal that is remaining is the oscillatory potential, providing the ERG contained an OP. Figure 3.6 details the entire procedure for the extraction of an oscillatory potential from the ERG of a non-diagnosed eye. To digress, prior to lowpass filtering of the left and right padding signals pre filtering conditioning is also required. The pre conditioning of these signals is accomplished by the attachment of padding signals. Figure 3.6 l)-g) illustrates the derivation of the left padding signal $S_{FLM}(t)$ and Figure 3.6 h)-k) illustrates the derivation of the right padding signal $S_{FRM}(t)$. Left and right padding signals are generated for both $S_{LM}(t)$ and $S_{RM}(t)$, however these padding signals do not require filtering.

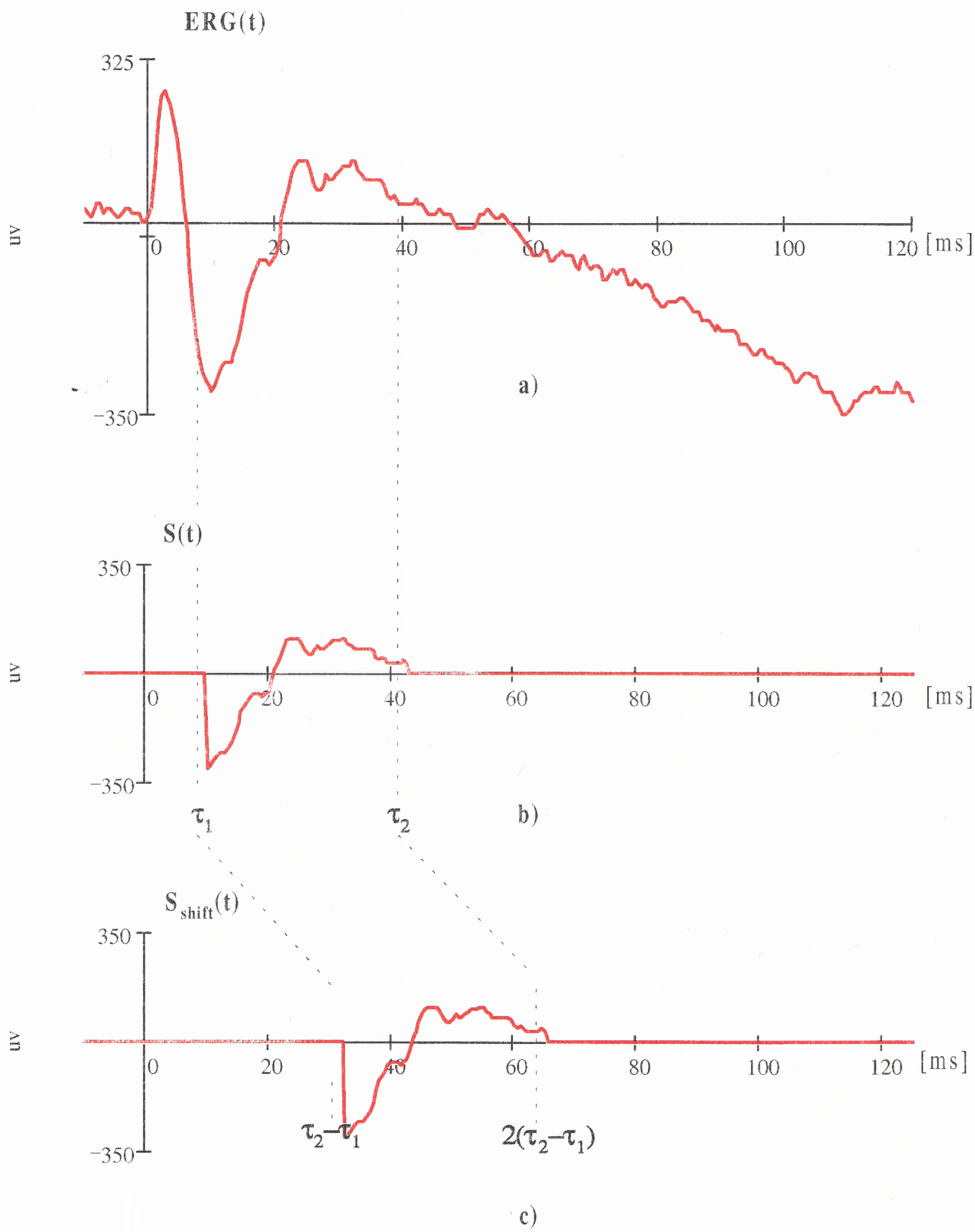


Figure 3.6 – a) As explained in the windowing section, the electroretinogram ERG(t) is windowed, S(t) in **b)**, to capture only the OP which is present between $\tau_1 \leq t < \tau_2$. **c)** S(t) is time shifted, $S_{shift}(t)$, to allow for the left padding signal to be inserted between $0 \leq \tau < (\tau_2 - \tau_1)$.

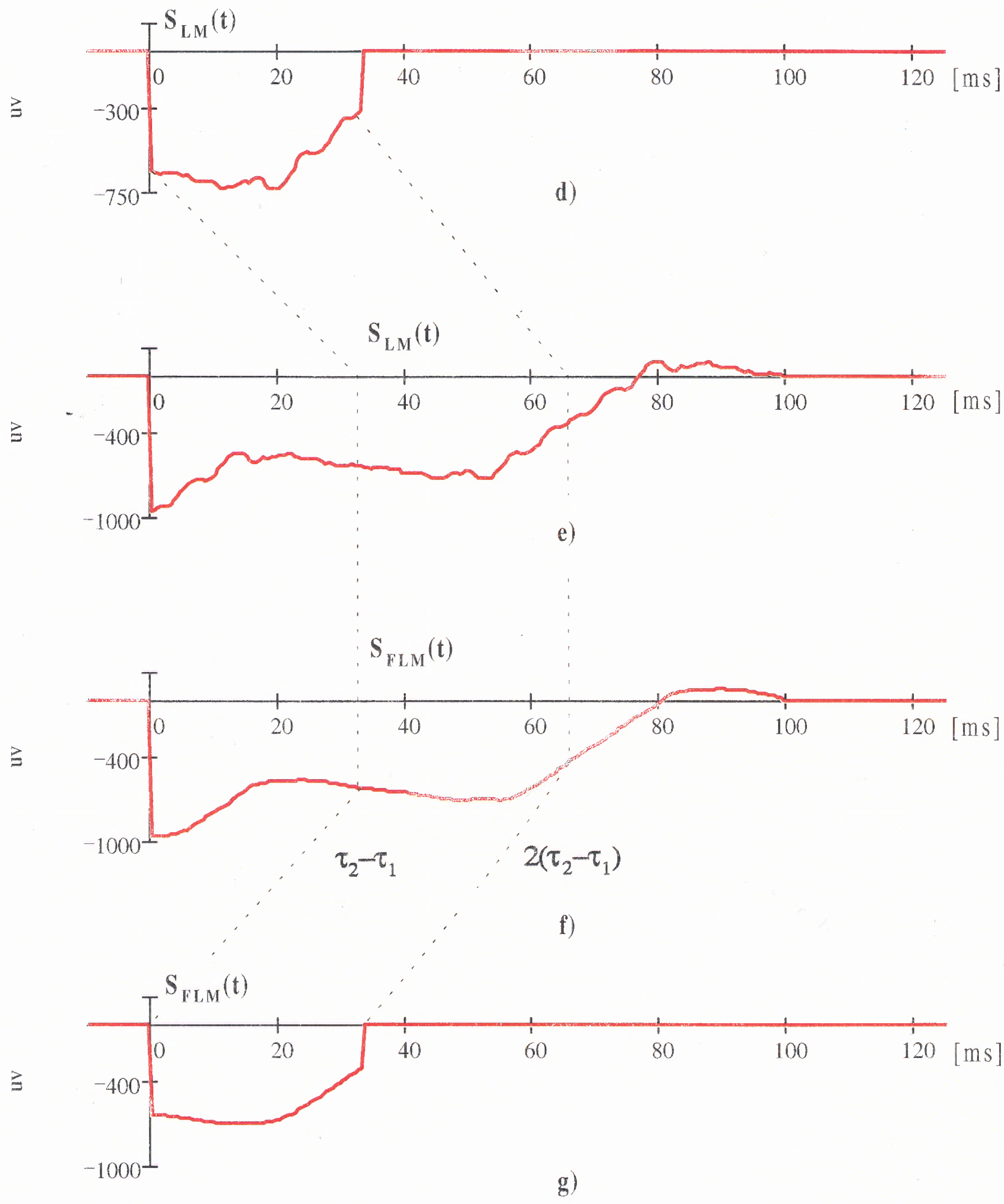


Figure 3.6 (continued) – d) The pre filtering conditioning technique requires the attachment of padding signals. $S_{LM}(t)$ is the unfiltered left padding signal and is the inverted mirror image of $S_{shift}(t)$ pivoted at $\tau_2 - \tau_1$. **e)** To extract the fundemental slope by filtering, $S_{LM}(t)$ is time shifted and has left and right padding signals attached for endpoint matching. **f)** The conditioned left padding signal is bilaterally lowpass filtered at 60 hertz, to create the left filtered padding signal, $S_{FLM}(t)$. **g)** $S_{FLM}(t)$ has the reminants of the padding signals removed and is shifted back to its correct location.

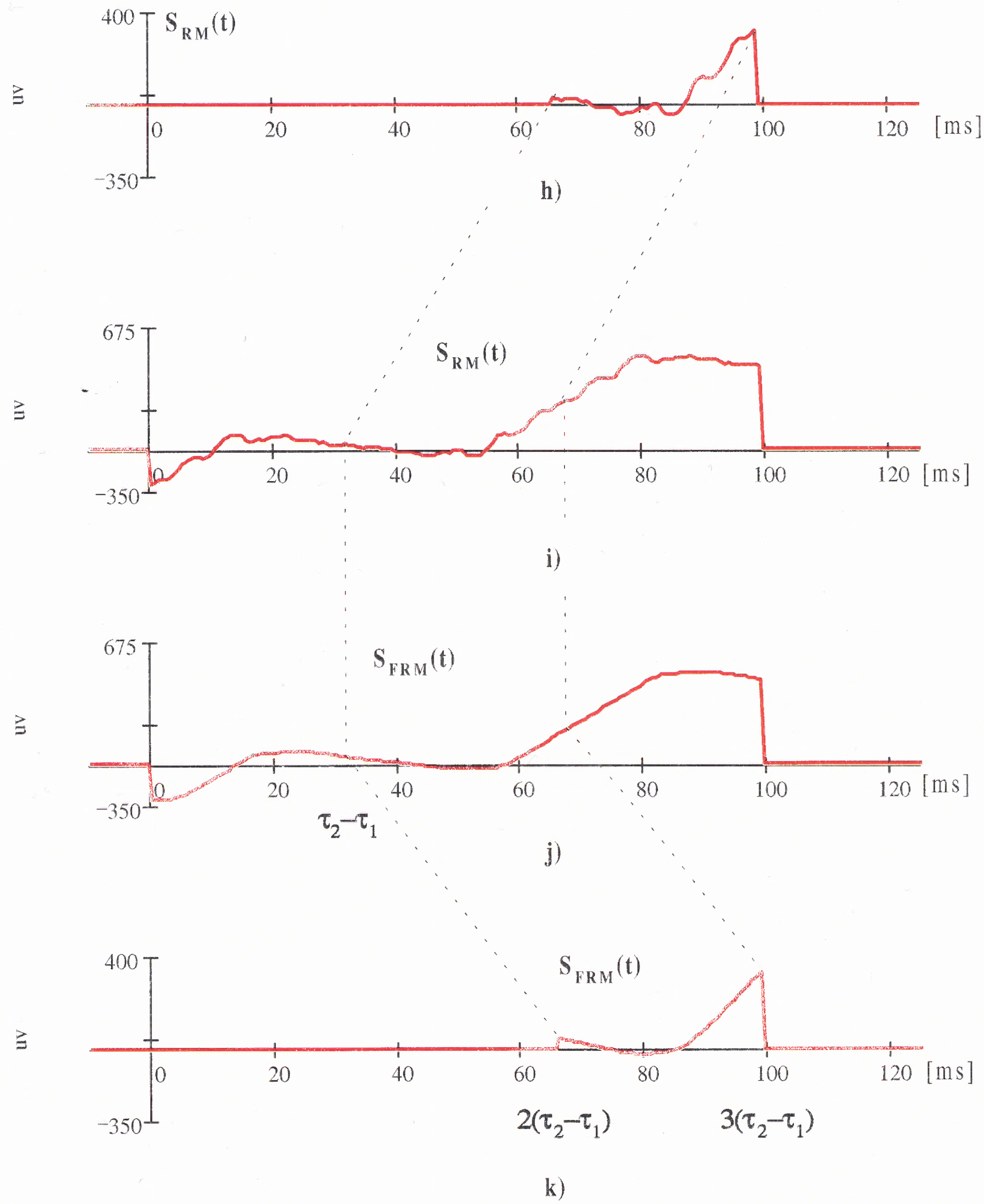


Figure 3.6 (continued) – h) The pre filtering conditioning technique requires the attachment of padding signals. $S_{RM}(t)$ is the unfiltered right padding signal and is the inverted mirror image of $S_{shift}(t)$ pivoted at $2(\tau_2-\tau_1)$. **i)** To extract the fundemental slope by filtering, $S_{RM}(t)$ is time shifted and has left and right padding signals attached for endpoint matching. **j)** The conditioned right padding signal is bilaterally lowpass filtered at 60 hertz, to create the right filtered padding signal, $S_{FRM}(t)$. **k)** $S_{FRM}(t)$ has the reminants of the padding signals removed and is shifted back to its correct location.

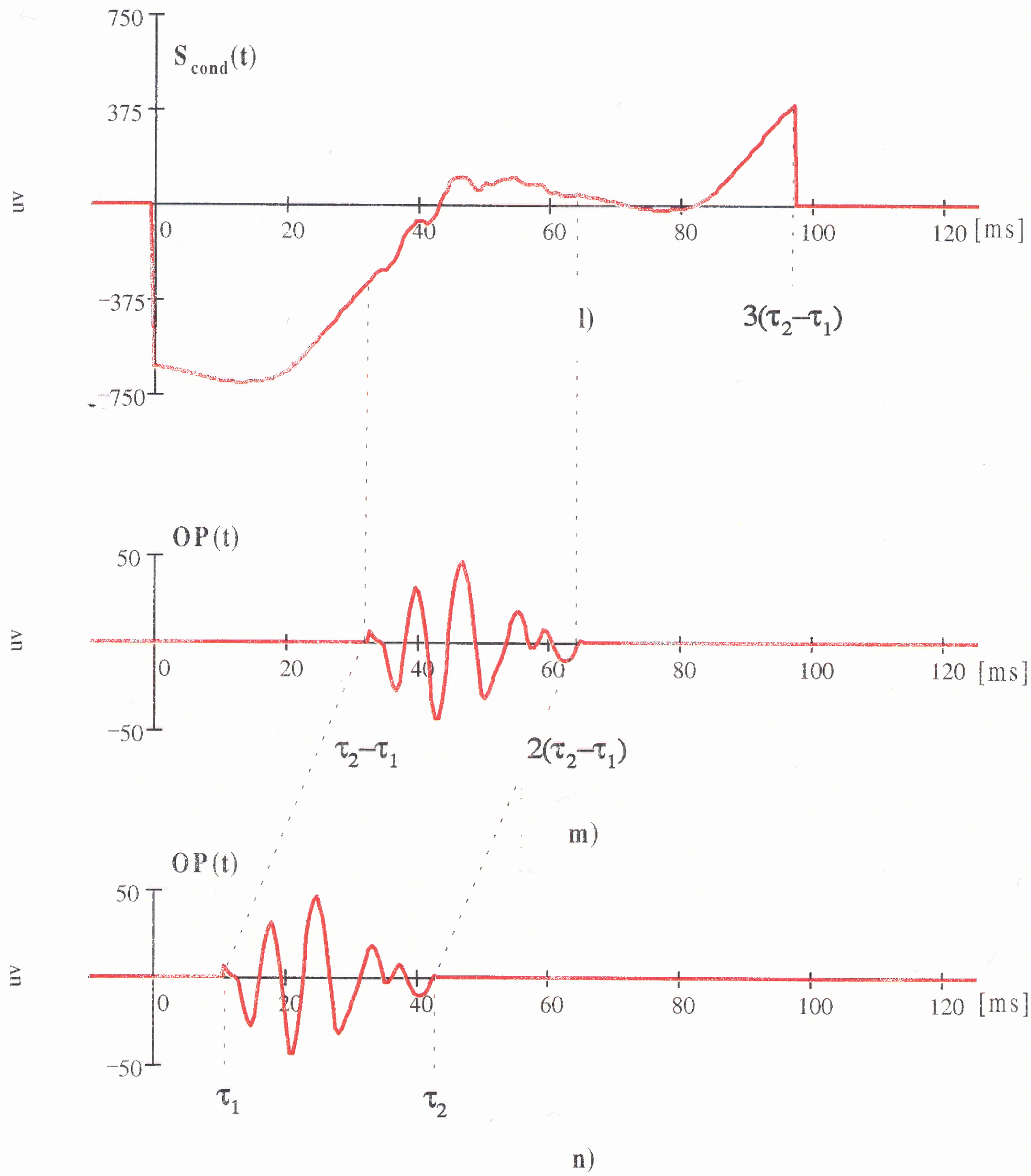


Figure 3.6 (continued) – l) The pre filtering conditioning technique requires the attachment of padding signals prior to filtering. $S_{cond}(t)$ is the time shifted windowed ERG signal, $S_{shift}(t)$, with padding signals $S_{LM}(t)$ and $S_{RM}(t)$ attached. **m)** To extract the OP, $S_{cond}(t)$ is bilateraly bandpass filtered. **n)** OP(t) has the reminants of the padding signals removed and is shifted back to its correct location.

3.4 Oscillatory Potential Extraction Program

To extract the OP from the ERG signal, an oscillatory potential extraction program was developed. The program realizes the pre-filtering signal conditioning technique discussed in section 3.3. The program provides versatility to the user in the extraction and evaluation of OPs. The user has the ability to extract OPs from multiple ERG data files in a batch process and then to manually reprocess discrete ERG data files for closer evaluation. The OP model used in this study has its parameters optimized to the extracted OP to minimize the root mean square error. The program provides the ability to view extracted OPs and to save the results in output data files. Figures 3.7 through 3.24 provides a summary of the theory of operation of the "Oscillatory Potential Extraction Program" through a sequential explanation of the graphical user interfaces. Appendix B lists the custom program modules and source code utilized in the extraction program..

The algorithm was implemented within the MATLAB Student Edition Version 5.0 environment. The program was developed using a module concept and was coded based on object oriented programming. Both built-in and custom functions were utilized in the realization of the algorithm. Custom functions were required to implement the ERG signal conditioning and filtering technique discussed in sections 3.2 and 3.3.

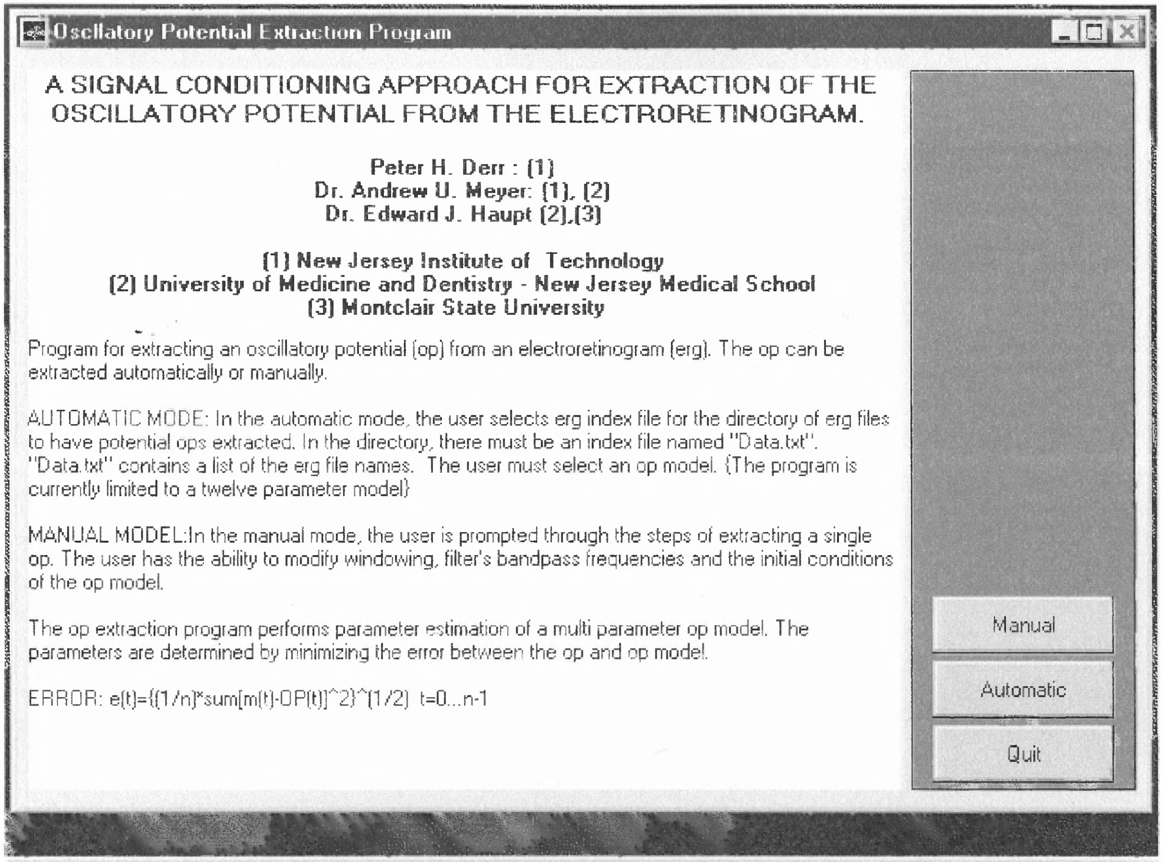


Figure 3.7 – The initial graphical user interface (GUI) of the Oscillatory Potential Extraction Program. The program provides two modes of OP extraction, Manual and Automatic. In Automatic mode, the OPs of a batch of ERGs are automatically extracted. The bandpass filter uses default corner frequencies in the OP extraction algorithm. The optimized parameters of the previous OP extraction are used as the OP model's parameters initial conditions for the current OP extraction. After the batch extraction, ERG header information, optimized OP model parameters and quality of extracted OP is stored in a file named "Evalute.dat". Also an ascii file containing the matrix with the column vectors ["Time", "OP model", "Extracted OP"] is created and name {ERG NAME}.OP. In the Manual Mode, the user has control over the selection of the windowing location and the corner frequencies of the bandpass filter used to extract the OP and selection of the OP model's parameters initial condition in the OP model optimization. After the manual extraction/optimization, ERG header information, optimized OP model parameters and quality of extracted OP is stored in a file named "Single.dat". Also an ascii file containing the matrix with the columns vectors ["Time", "OP model", "Extracted OP"] is created and name {ERG NAME}.OP.

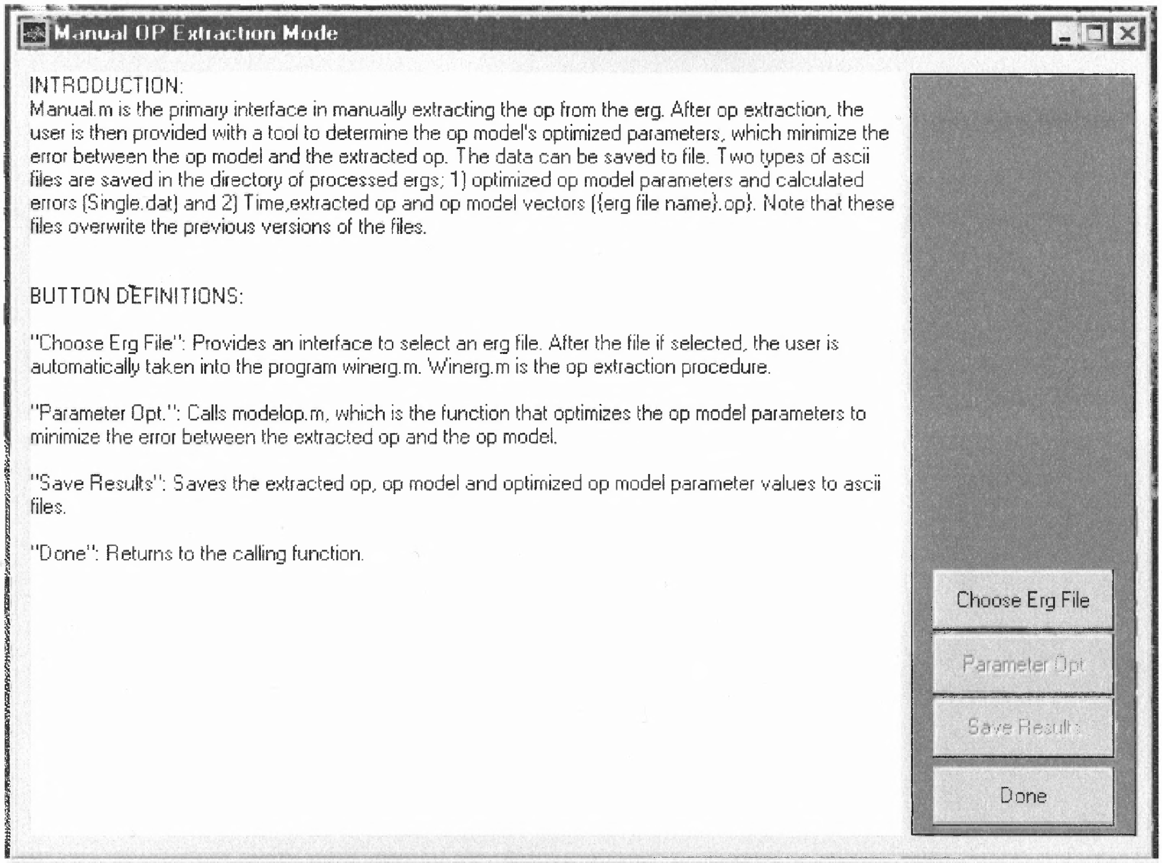


Figure 3.8 – After depressing the "Manual" button on the program's main GUI, figure 3.7, the "Manual OP Extraction Mode" GUI is called. The "Choose ERG File" and "Done" buttons are initially the only functions available. The "Choose ERG File" is depressed to proceed with the extraction. If the "Done" button would have been depressed, the program would have returned to the GUI in Figure 3.7.

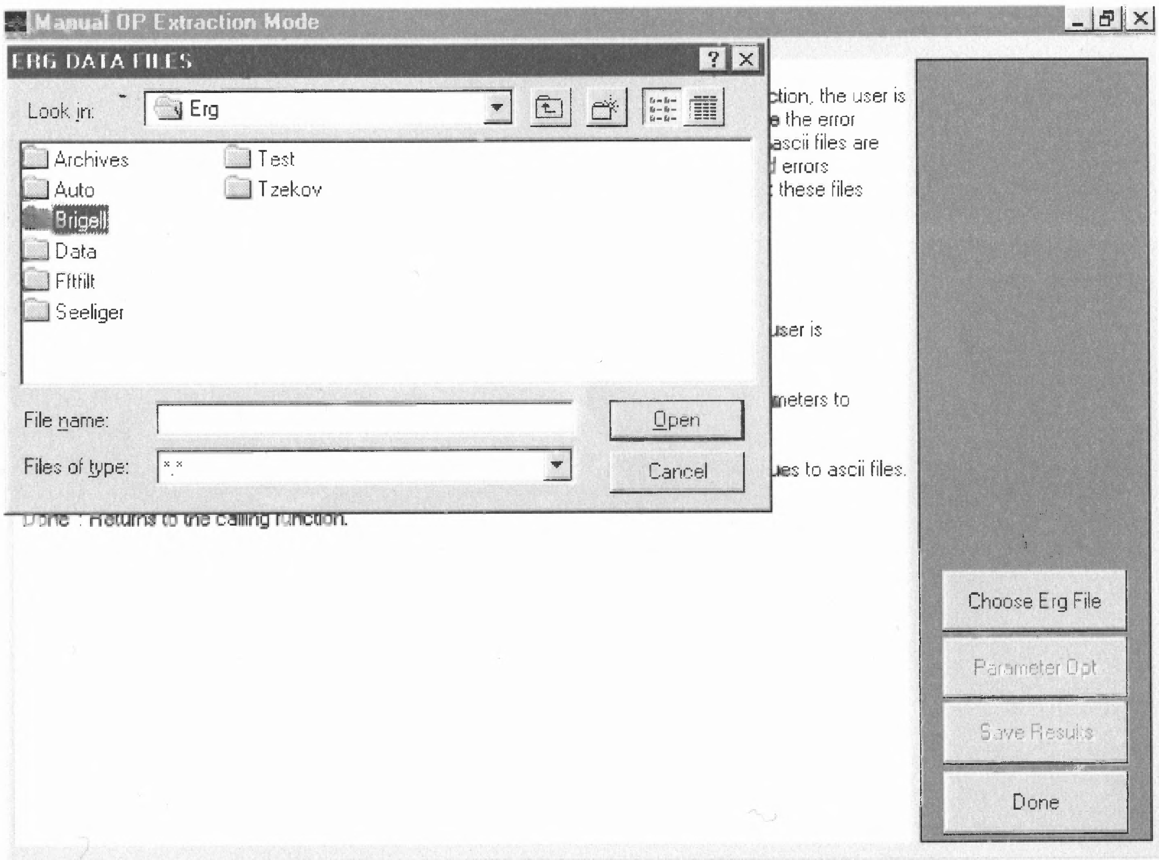


Figure 3.9 – The user must select the folder which contains the ERG data file to be processed. After extracting the OP, the file "Single.dat and the ascii file containing the matrix ["Time", "OP Model" and "Extracted OP"] will be saved to this folder.

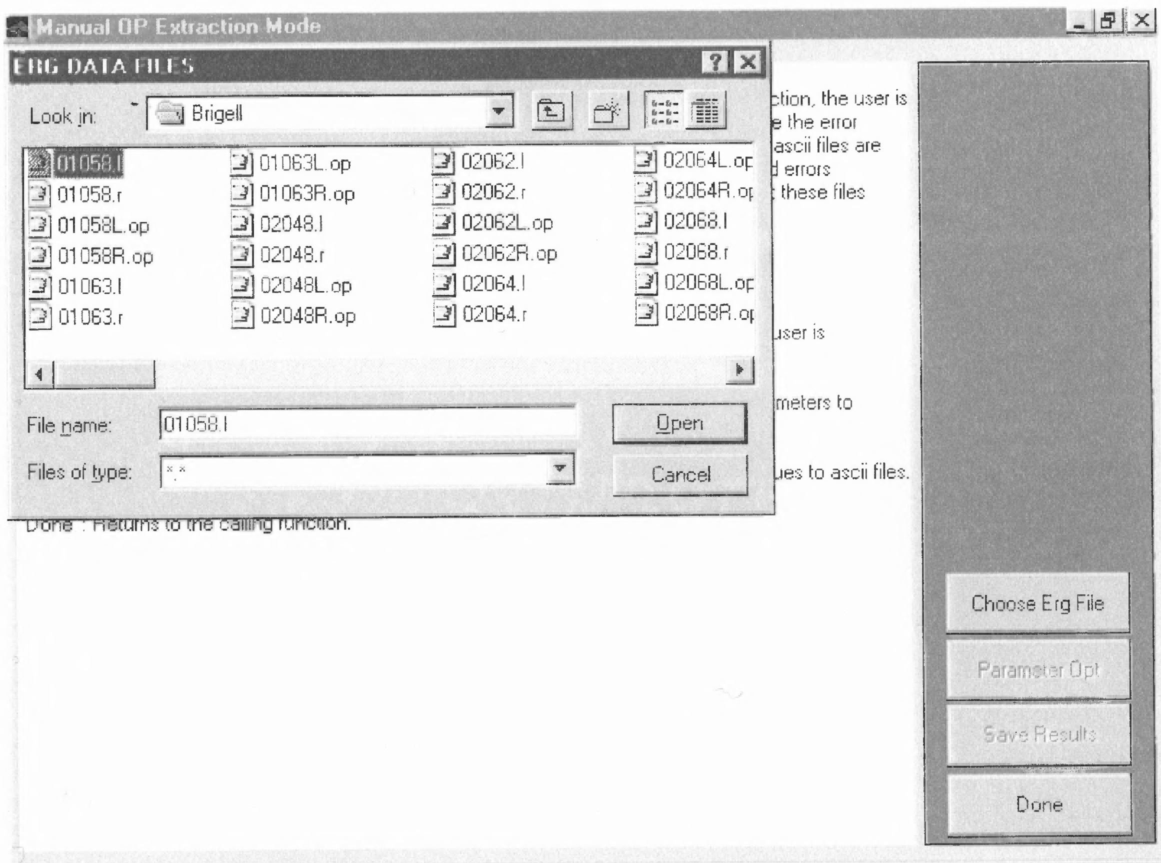


Figure 3.10 – The user must select the ERG file to be processed. The file must be in the format illustrated in Appendix A1.

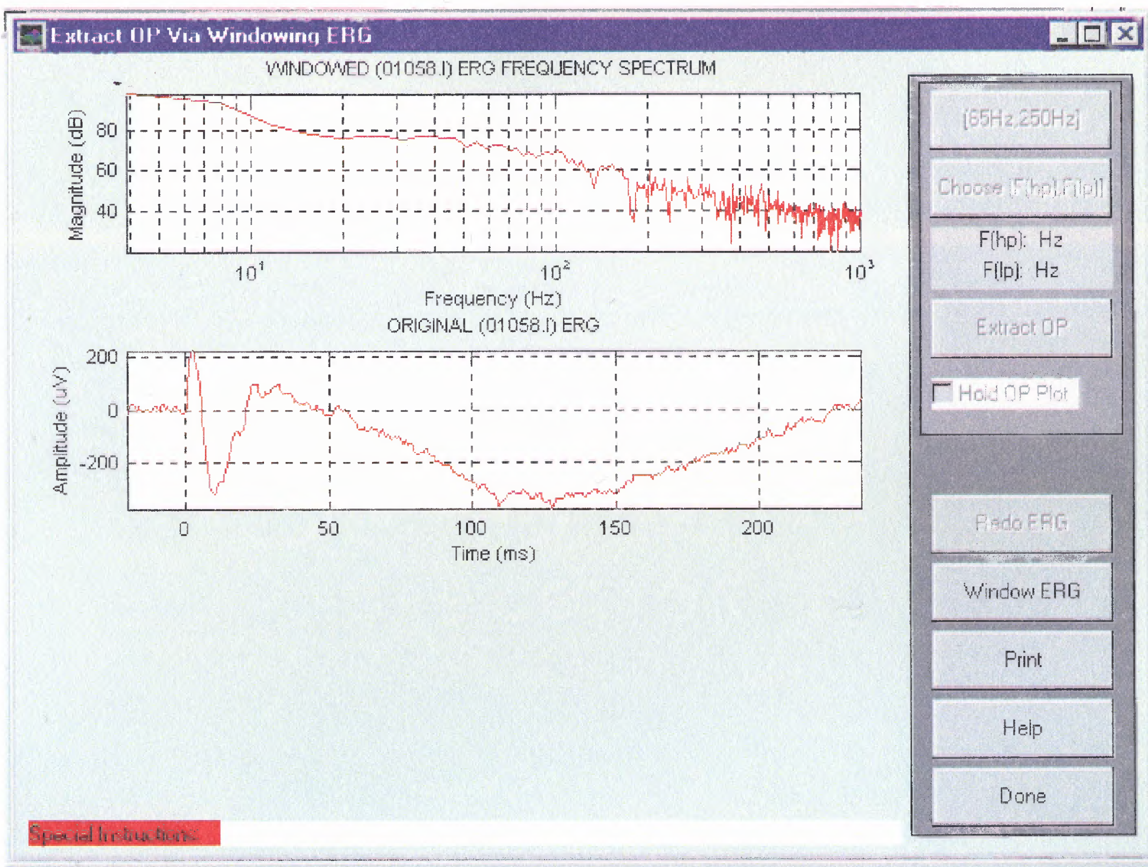


Figure 3.11 – The "Extract OP Via Windowing ERG" GUI is the engine for extracting the OP in the manual mode. The ERG with its frequency spectrum is initially displayed. The 'Window ERG", "Print", "Help" and "Done" buttons are active. The user can window the ERG with a uniform window. The displayed plots can be printed. A help window is available by depressing the "Help" button. The program can be returned to the GUI on figure 3.9 by depressing the "Done" button. The 'Windowed ERG" will be depressed. The user is prompted to select the left and right windowing locations. The selection is made by positioning the crosshairs over the left and right locations and clicking the left mouse button for each location, respectively.

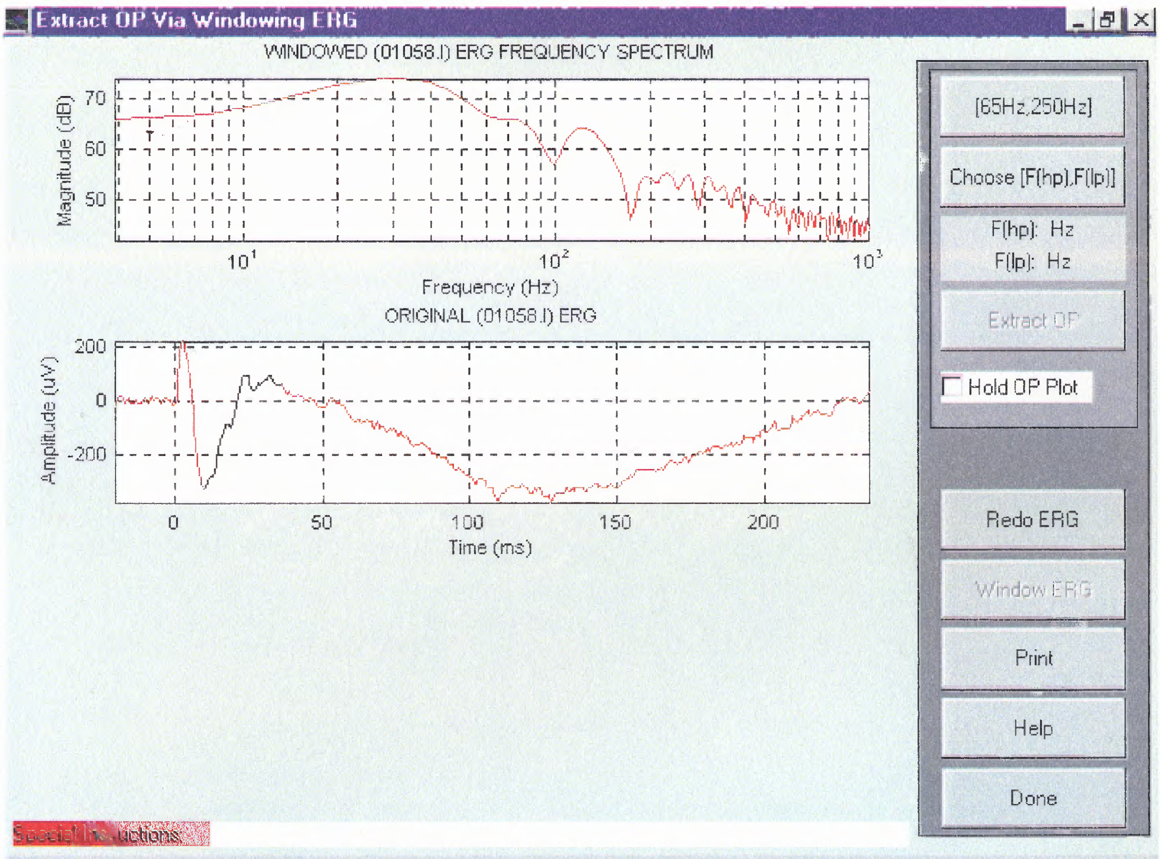


Figure 3.12 – After selecting the left and right windowing locations, the windowed ERG is highlighted and the frequency spectrum of the windowed ERG is plotted. The "[65Hz, 250Hz]", "Choose [F(hp), F(lp)]" and "Redo ERG" buttons are now active. The user can use the default bandpass filter of [65Hz, 250Hz] by depressing the "[65Hz, 250Hz]" button. A custom bandpass can be selected by depressing "Choose [F(hp), F(lp)]". The user will select the highpass and lowpass cutoff frequencies by positioning the crosshairs in the frequency spectrum plot and clicking the left mouse button, respectively. The default bandpass will be selected. After selecting the bandpass filter, the "Extract OP" button will be activated.

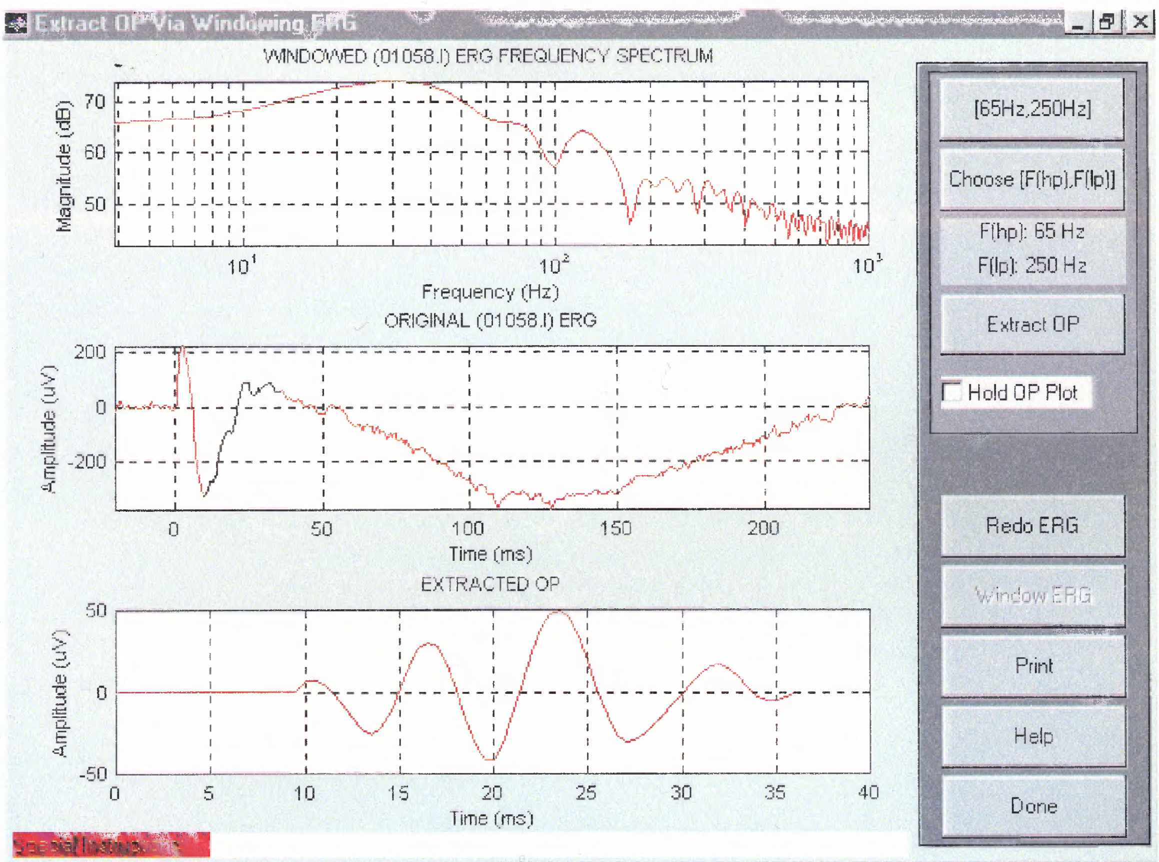


Figure 3.13 – The "Extract OP" button was depressed. The extracted OP is plotted below the ERG plot. If the user would like to choose different cutoff frequencies but retain the current OP plot for future comparison, The "Hold OP Plot" box can be checked. If the box is not checked the current OP will be overwritten. If the OP is satisfactory, the "Done" button can be depressed.

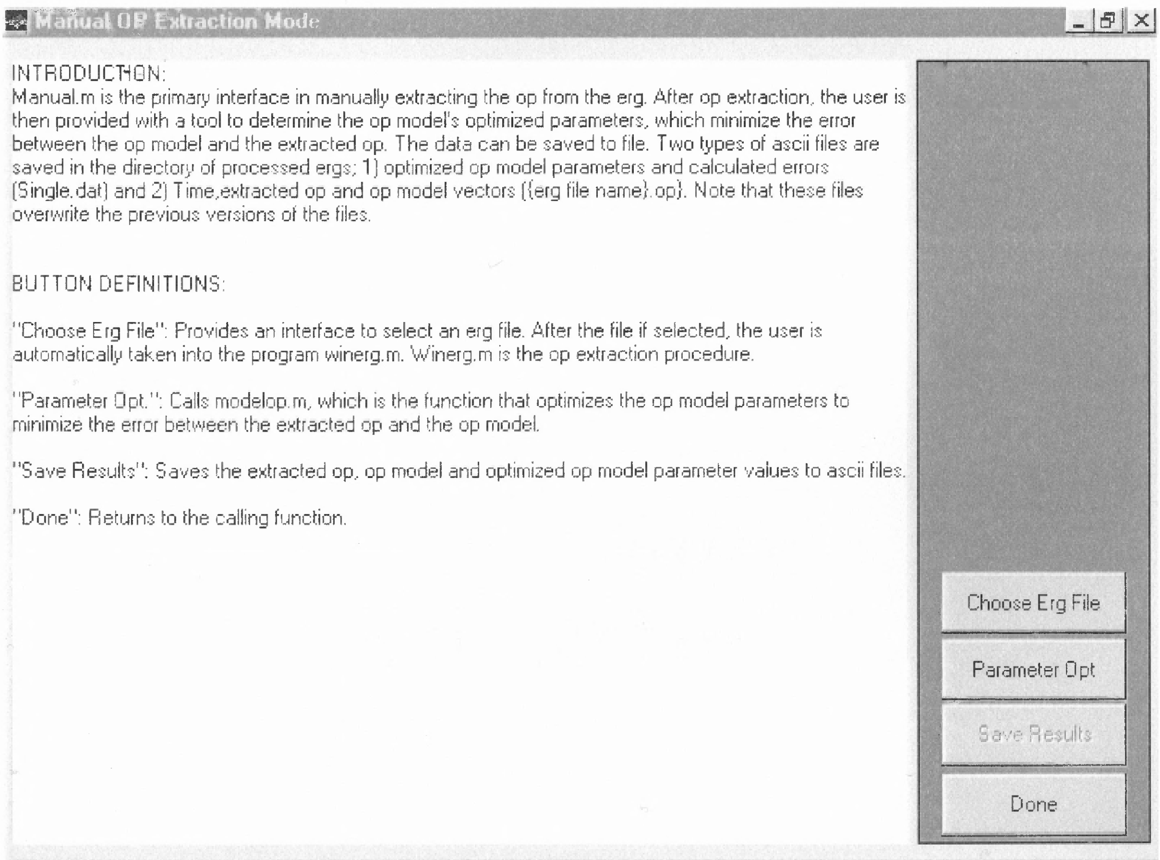


Figure 3.14 – After extracting the OP, the program returns to the "Manual OP Extraction Mode" GUI. The "Parameter Opt" button is active. The button is depressed to goto the OP model parameter optimization engine.

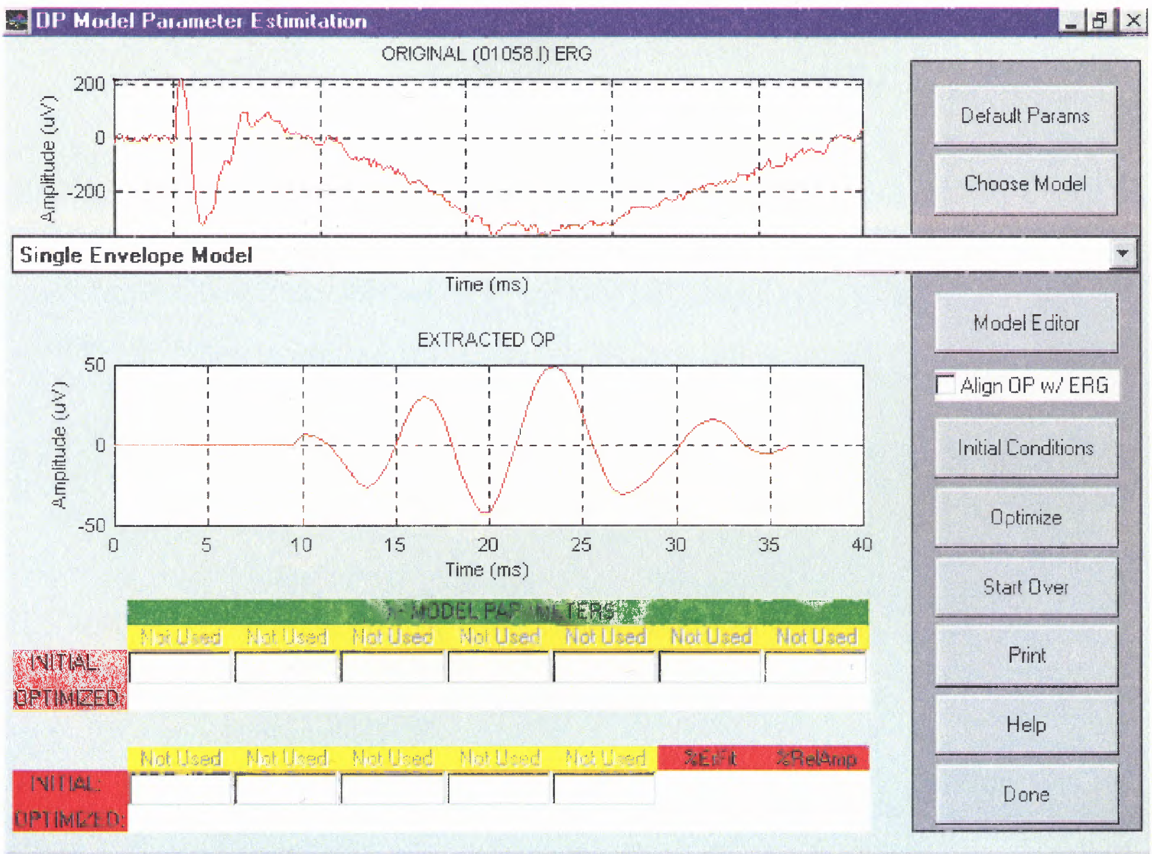


Figure 3-15 – The "Model Parameter Estimation" GUI is the engine that provides optimization of the selected OP model's parameters to the extracted OP. The ERG and the OP are initially the plotted. The user must select the desired OP model. For this study the Single Envelope model was used.

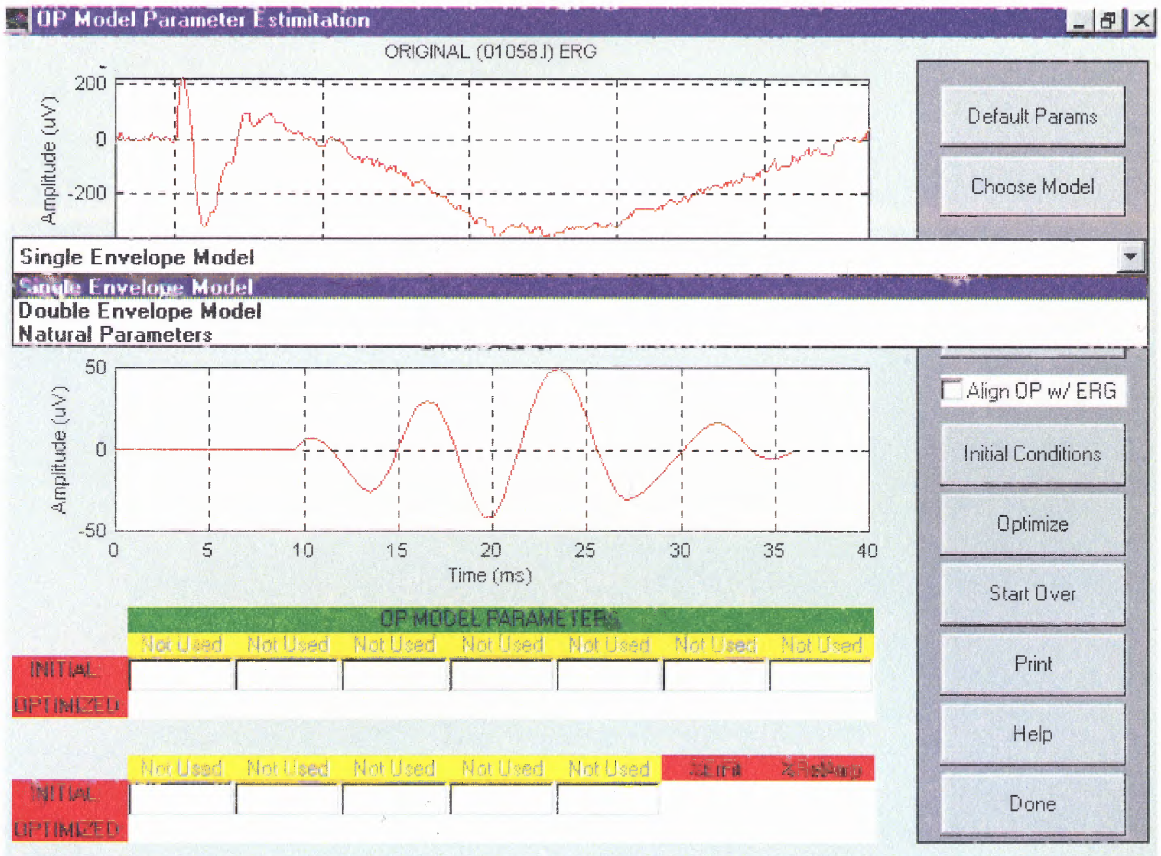


Figure 3.16 - The Single Envelope Model was used for this study. The user could select any model displayed in the pull down menu or create a new model by depressing the "Model Editor" button.

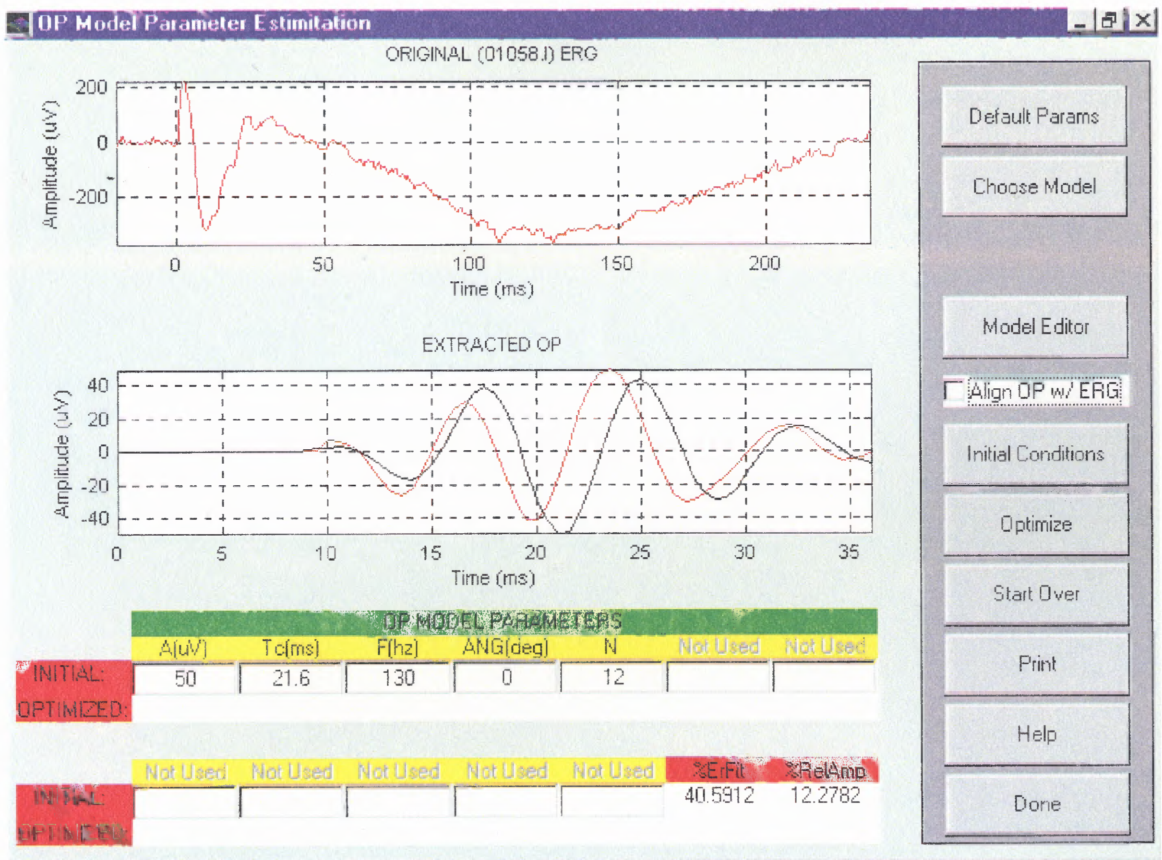


Figure 3.17 – After the OP model is selected, the OP model's parameters initial conditions are displayed; along with the %ErFit [(Fit Error / A)*100] and %RelAmp [(A / Atp)100]. By depressing the "Initial Condition" button, the OP model is plotted on the Extracted OP plot.

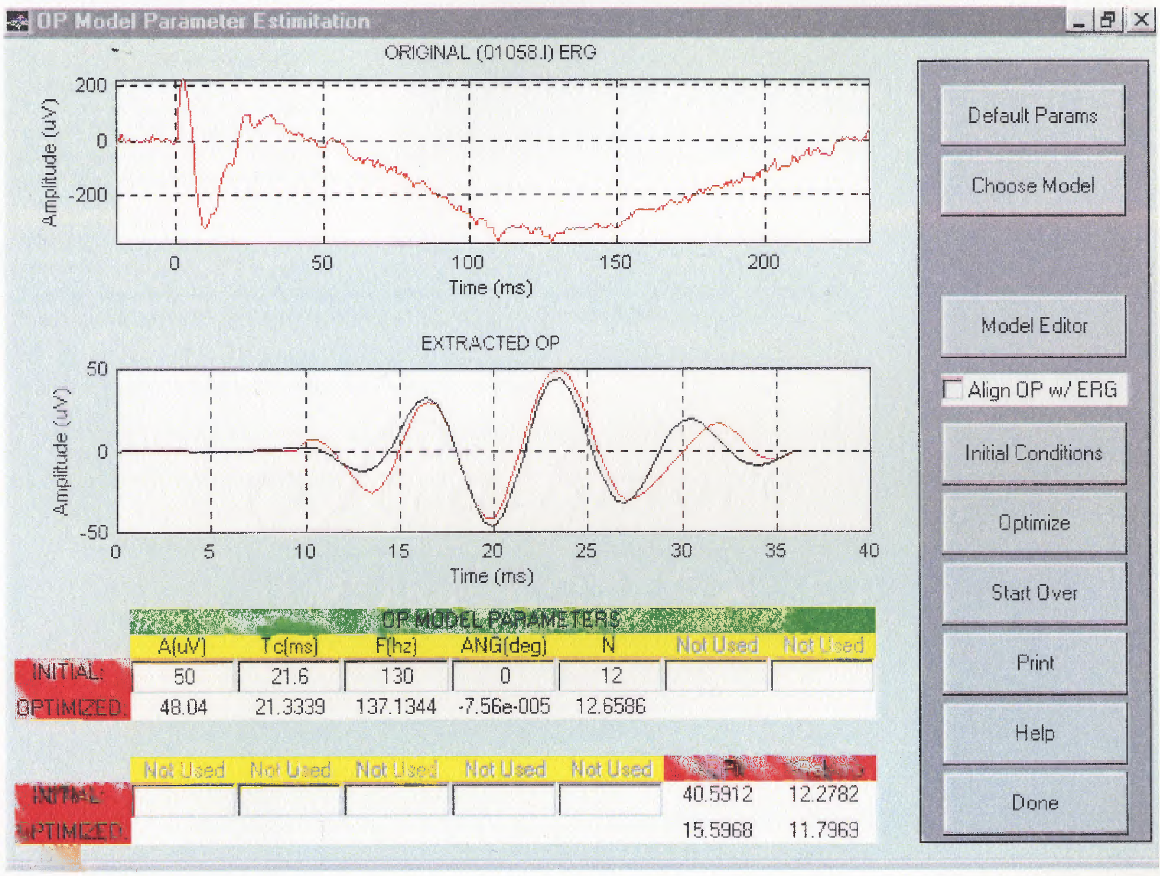


Figure 3.18 – After the depressing the "Optimize" button, the engine uses MATLAB's fmins simplex algorithm to optimize the OP model's parameters to the extracted OP. If by visual inspection and evaluation of the values %Erfit and %RelAmp, the fit is deemed unsatisfactory, the user can supply another set of initial conditions and rerun the optimization. When the optimization is completed, the "Done" button is depressed.

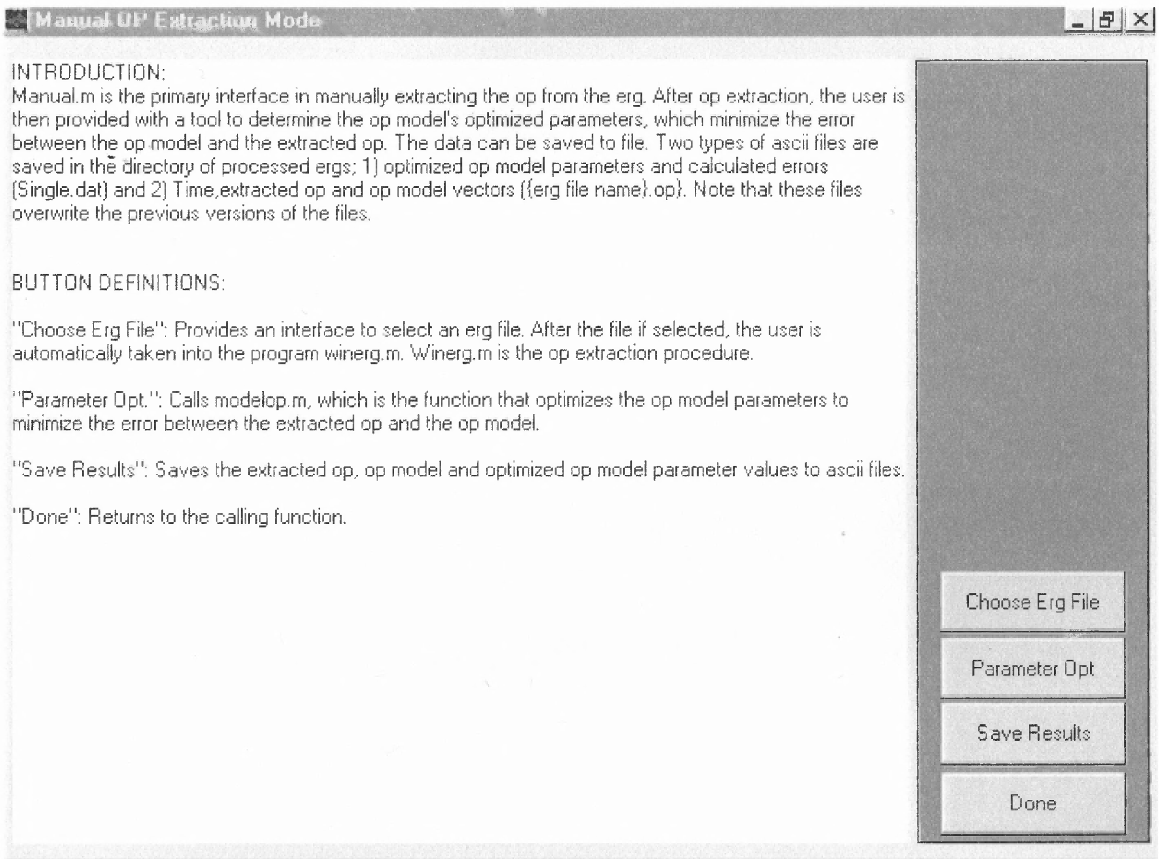


Figure 3.19 – After completing the OP model parameter optimization. The program returns to the "Manual OP Extraction Mode" GUI. The "Save Results" button is active. By depressing the "Save Results" button the ERG header information; bandpass filter's cutoff frequencies; sample frequency; b-wave's trough to peak amplitude; optimized OP model parameters; fit error; %ErFit and %RelAmp are saved in the ascii file "Single.dat". The file is stored in the folder containing the ERG file. The user can process another ERG or quit the Manual mode. By depressing the "Done" button will return the user to GUI illustrated in figure 3.7.

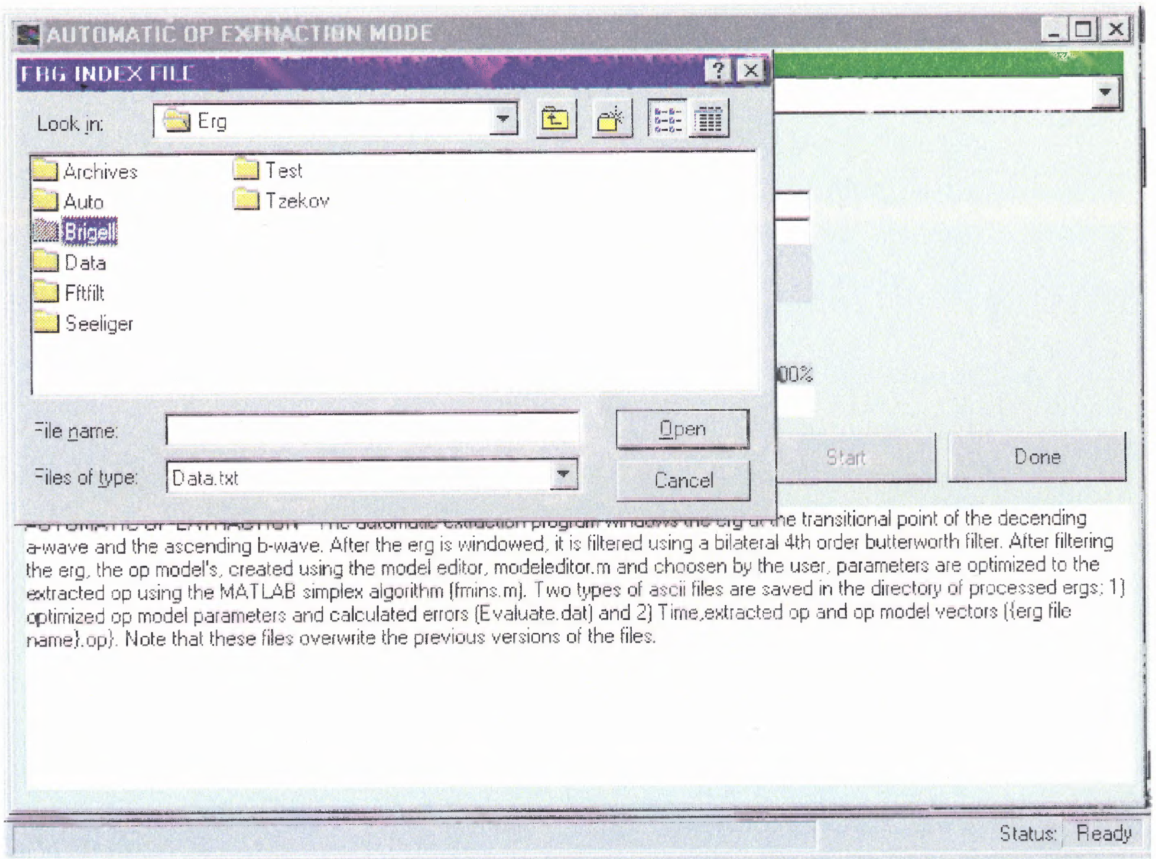


Figure 3.20 – If the "Automatic" button on the "Oscillatory Potential Extraction Program" GUI, see figure 3.7, is depressed, the user must select the folder containg the batch of ERG files to be processed.

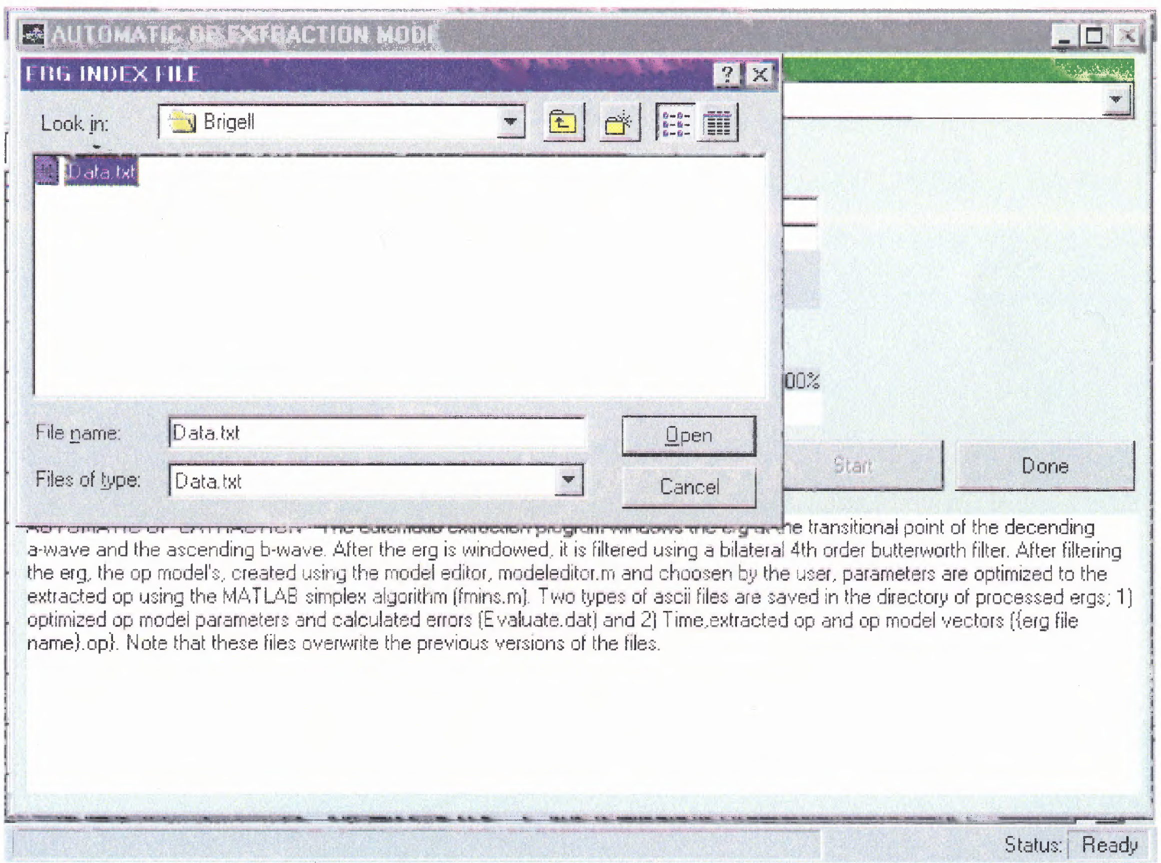


Figure 3.21 – In the Automatic mode, the user must supply an ERG index file. The index file contains a list of the ERG files, including the file extention, to be processed. For convience the index file should be named “data.txt”.

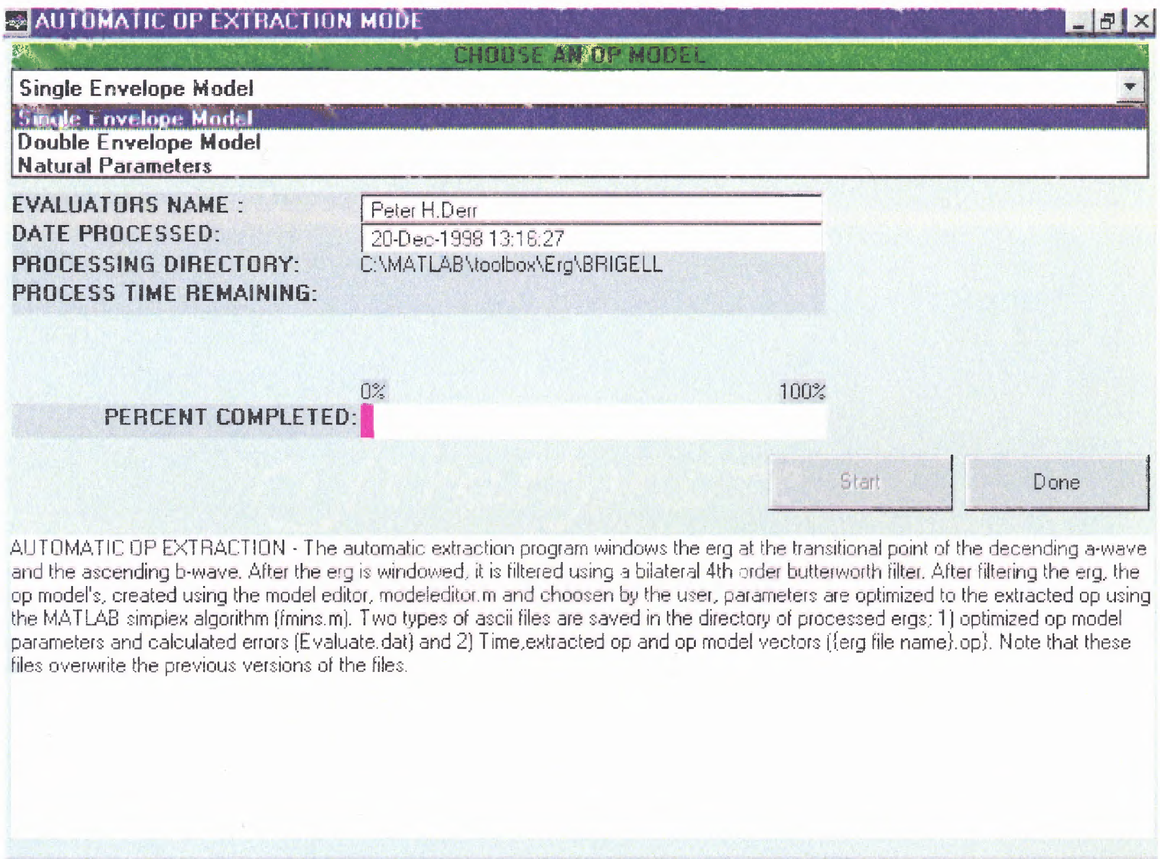


Figure 3.22 – The user must select the OP model to have its parameters optimized to the extracted OP. The user can input the evaluator's name. The automatic extraction process is date/time stamped.

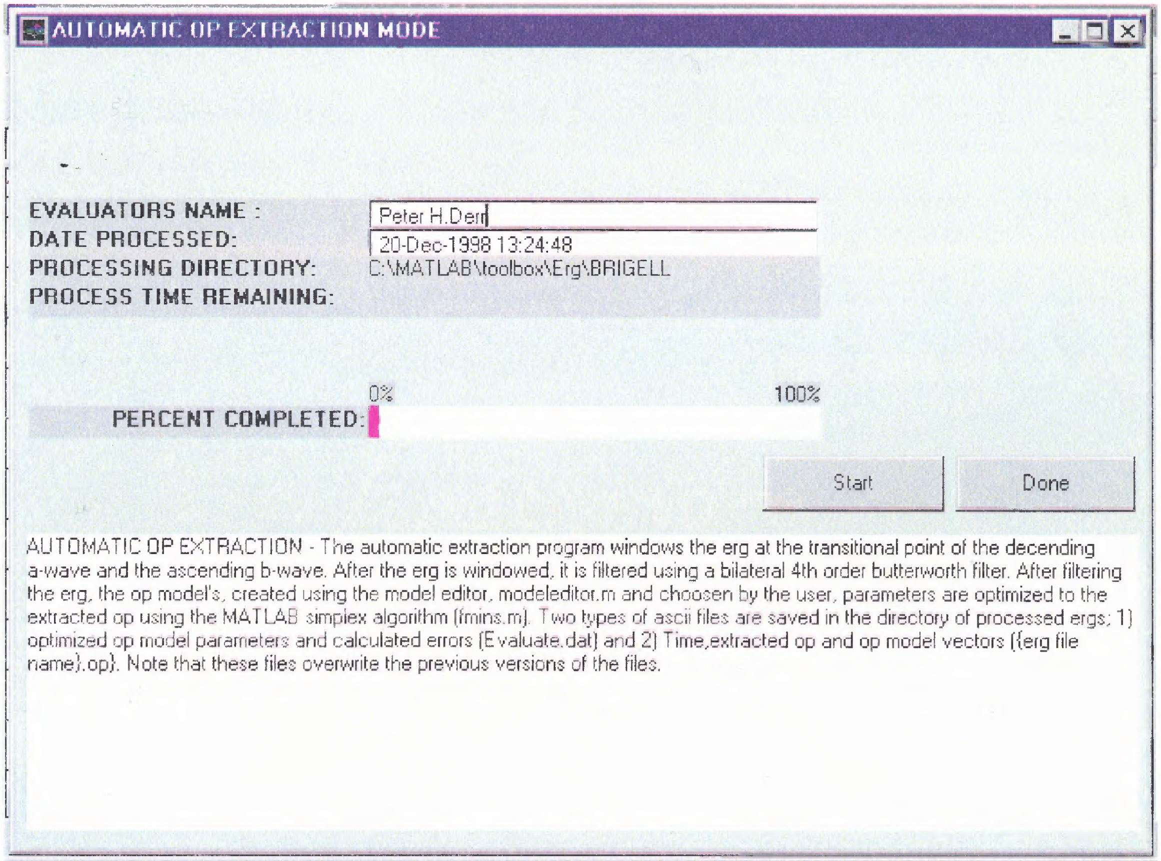


Figure 3.23 – After the OP model is selected, the "Start" button is active. After depressing the "Start" button the GUI provides real time process status. The process time remaining is posted along with a percent completed bar. After the automatic extraction program is completed the ERG header information; bandpass filter's cutoff frequencies; sample frequency; b-wave trough to peak amplitude; optimized OP model parameters; fit error; %ErFit and %RelAmp are saved in the ascii file "Evaluate.dat". The file is stored in the folder containing the ERG file. Depressing the "Done" button will return the user to GUI illustrated in figure 3.7.

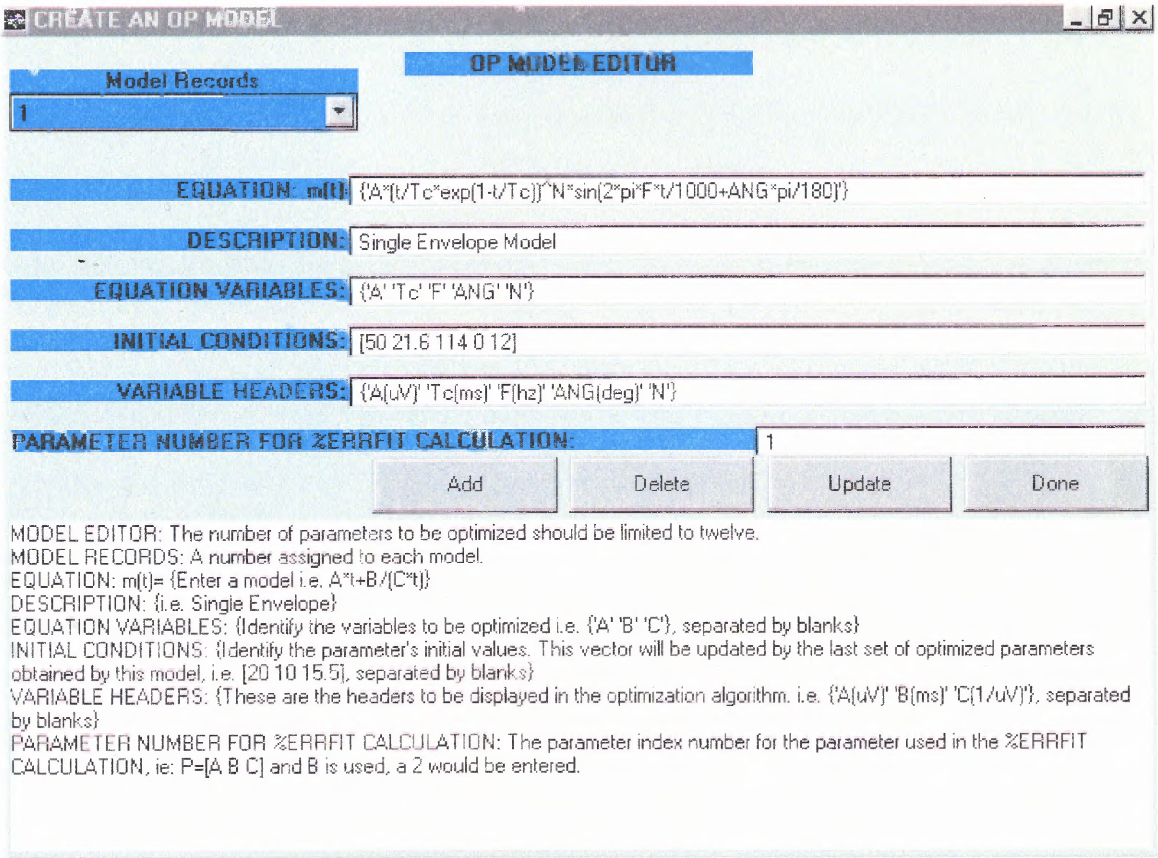


Figure 3.24 – The "Create An OP Model" GUI provides a means to create new OP models. The model information must follow a specific syntax. It is suggested to follow an existing model's format for proper syntax.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Verification of Method

The proposed windowing and signal conditioning technique was tested to verify that the electroretinogram (ERG) signal could be successfully bandpass filtered to extract a bona fide oscillatory potential (OP). A test signal, which resembles an ERG, was constructed. The signal contained characteristics of the ERG, which may produce the natural response of a filter, if not conditioned prior to filtering. The characteristics were the impulse-like flash artifact, the transition to the positive cornea b-wave and a simulated OP. Figure 4.0 illustrates the a) simulated OP (solid trace) and the composite signal (dotted trace) without the simulated OP; b) the composite signal with the superimposed simulated OP; and c) the extracted OP (solid trace) compared to the original OP (dotted trace).

The simulated OP signal was generated using the single envelope OP model, equation 4.0, developed by Dr. Andrew Meyer of New Jersey Institute of Technology. The parameter vector was set equal to [A=40 uV, t_c=20 ms, f=120 Hz, ϕ=0, N=15]*. The

$$m(t) = A\left(\frac{t}{t_c}\right)^N e^{\left(1-\frac{t}{t_c}\right)^N} \sin\left(2\pi ft + \frac{\phi\pi}{180}\right) \qquad (4.0)$$

flash artifact was modeled using the signal constructed by equation 4.1. The a + b wave

$$imp(t) = \begin{cases} 100, 200, 300, 275, 250, 225, 200, 175, 150, 125, 100, 75, 50, 25 & t = 0, 0.5, \ldots, 6.5ms \\ 0 & 7.0 \le t \le 100ms \end{cases} \qquad (4.1)$$

* The initial condition values for the parameter vector was chosen based on the results of a previous investigation [13]. Special attention in selection of initial conditions is needed to guarantee curve fitting to a global minima error.

signal was constructed based on equation 4.2. The composite signal, illustrated in equation 4.3 is the summation of equations 4.0, 4.1 and 4.2.

$$x(t) = 50\sin(2\pi 40t + \pi) + 100\sin(2\pi 10t + \frac{3\pi}{2}) \tag{4.2}$$

$$composite(t) = m(t) + imp(t) + x(t) \tag{4.3}$$

Using the OP extraction program, which implements the windowing and pre filter signal conditioning technique, the simulated OP was successfully extracted. The root mean square (rms) error between the extracted OP and the simulated OP was calculated and was equal to 2.1240. Also visual inspection of the extracted OP shows that there was no significant difference between the two signals. The differences that are present can be accounted by spectral overflow from other components of the windowed composite signal which has frequency components within the OP bandwidth. Figure 4.1 a) shows the frequency spectrum of the simulated OP (solid trace) and the windowed composite signal without the OP (dotted trace). Figure 4.1 b) compares the simulated OP with component of the windowed composite signal that did not contain an OP, which has the same frequency bandwidth of the simulated OP.

To give some perspective to the calculated rms error, a signal of normally distributed random noise with zero mean and a variance of one, which was amplified by a gain of forty, was compared to the simulated OP. The comparison was based on the resulting rms error of the two signals. After curve fitting to the OP model, the noise's rms error was equal to 41.2 and the envelopes amplitude was equal to 40. To provide a more meaningful rms error value, the error should be scaled by the size of the OP.

Therefore, the rms error of the two OP becomes (2.2140/40)*100 or 5.53% and the comparison of the random noise to the OP becomes (41.2/40)*100=103%. This error percent is termed %ErFit. The low %ErFit for the extracted simulated OP compared to the original simulated OP verifies that the extracted OP is in fact a slightly distorted.

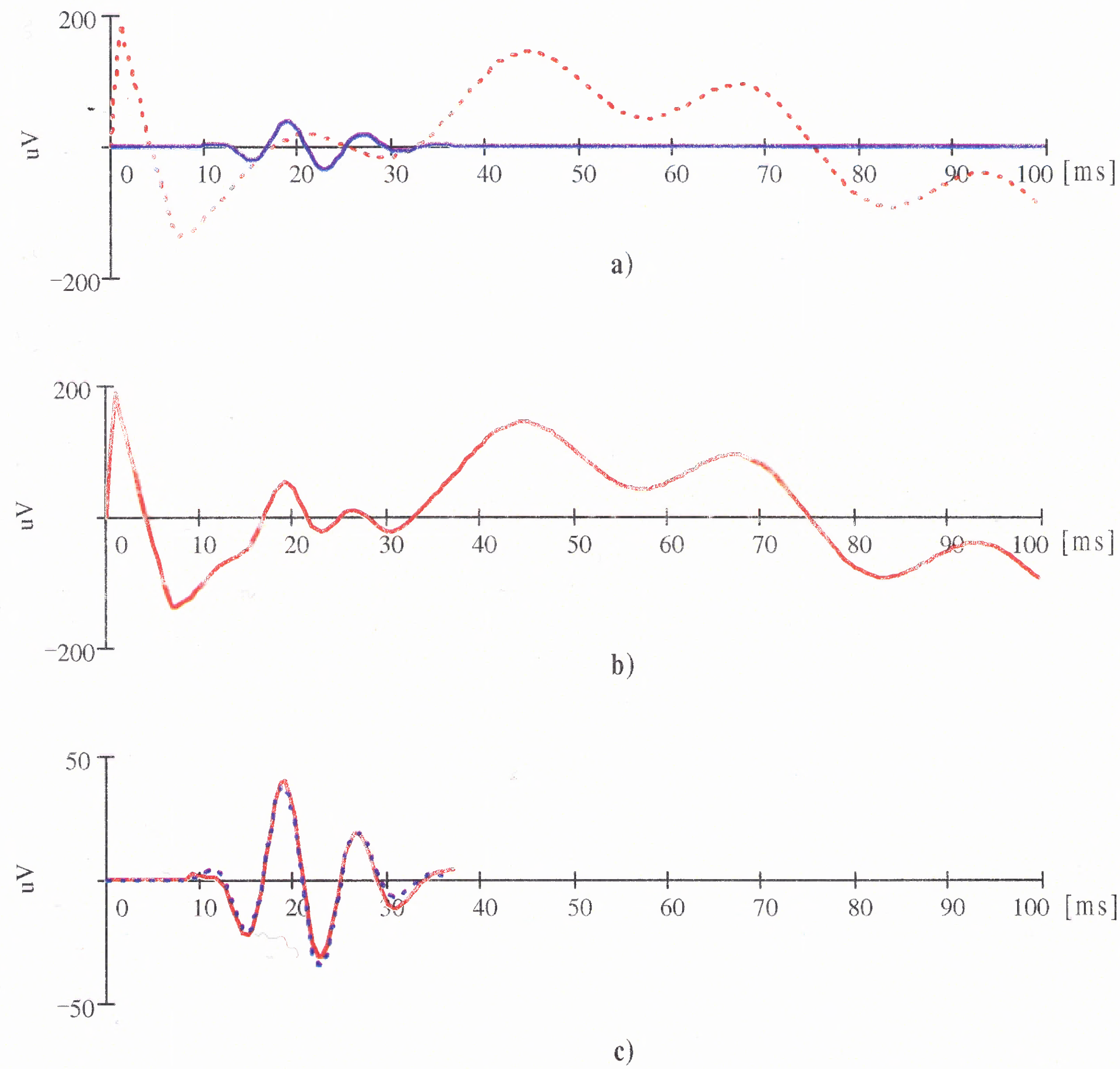


Figure 4.0 - a) simulated OP (solid trace) and the composite signal (dotted trace) without the simulated OP; **b)** the composite signal with the superimposed simulated OP ; and **c)** the extracted OP (solid trace) compared to the simulated OP (dotted trace).

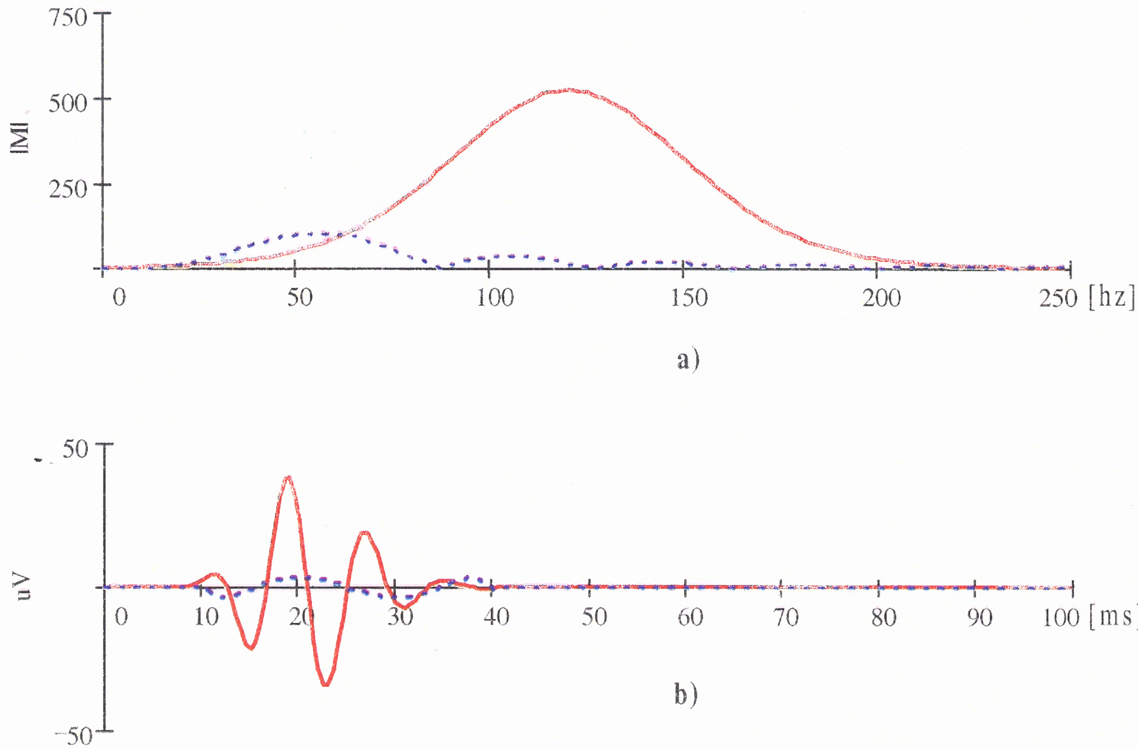


Figure 4.1 - a) shows the frequency spectrum of the simulated OP (solid trace) and the windowed composite signal without the OP (dotted trace). **b)** compares the simulated OP with component of the windowed composite signal that did not contain an OP, which has the same frequency bandwidth of the simulated OP.

4.2 Extracted Oscillatory Potential

Using the proposed signal conditioning technique , Dr. Brigell's fifty electroretinograms (ERG) were windowed, signal conditioned and filtered in an effort to extract the oscillatory potential (OP). Using the "Oscillatory Potential Extraction Program", Dr. Andrew Meyer's five parameter single envelope OP model [12], [13], equation 4.4 was optimized to the extracted OP. Figure 4.2 illustrates a possible OP model.

$$M(t) = A(t/Tc)^N e^{(1-t/Tc)^N} \sin(2\pi ft + ANG * \frac{\pi}{180}) \tag{4.4}$$

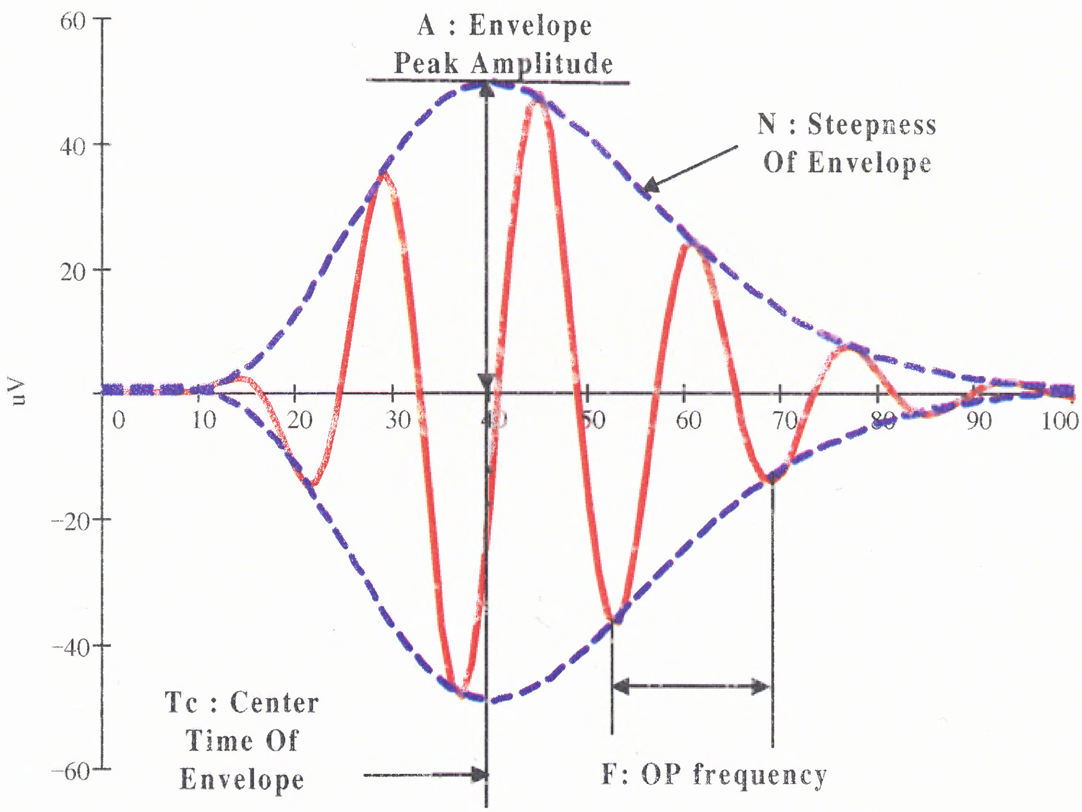


Figure 4.2 – The solid line plot is an OP model based on equation 4.4. The model parameters were chosen to be A=50 uV, Tc=40 ms, F=121 Hertz , ANG=0 and N=7. The OP model envelope is outlined by the dashed line plots.

The single OP model's parameters are defined as follows: A is the maximum amplitude of the OP model envelope; Tc is the time in which A occurs; F is the frequency of the OP signal; ANG is the phase angle of the OP signal; and N is the steepness factor of the OP model envelope. The parameters were optimized to minimize the rms error of the OP model and the extracted OP, reference equation 4.5, where M is the OP model and OP is the extracted signal.

$$Error = \sqrt{\frac{1}{n}\sum (M_i - OP_i)^2} \tag{4.5}$$

The proposed technique was implemented using the OP extraction program. The ERGs were processed with the program set to automatic mode. Cases with questionable parameters, such as the model's central time occurring prior to τ_1, or the model's frequency exceeding the filter's bandpass frequencies, were reprocessed with the program's manual mode. The bandpass filter was kept constant for all fifty ERG cases, with the corner frequencies set at 65 and 250 hertz. The fifty sets of optimized OP parameters are tabulated in Table 4.0. The cases in the table are grouped by diagnosis with each group containing two sub groups; the eyes that went to neovascularization and those which did not go to completion. The arithmetic mean is calculated for each group and subgroup.

Comparing each diagnosed group, including the subgroups which went to neo vascularization completion, the calculated mean of each group suggests that the OP parameters do change dependent on the condition of the eye. The calculations which show the most obvious separation of the four diagnosis groups are the %ErFit and %RelAmp. %ErFit and %RelAmp are determined by equations 4.6 and 4.7.

$$\%ErFit = (Error / A) * 100 \tag{4.6}$$

$$\%RelAmp = (A / Atp) * 100 \tag{4.7}$$

The Error is calculated using equation 4.5. A is the OP model envelope's maximum amplitude and Atp is the filtered positive cornea b-wave's amplitude. %ErFit provides a relative interpretation to the parameter optimization error. The Error is scaled by the amplitude of the OP model envelope's amplitude. If the optimization error is not scaled, the resulting value from equation 4.5 will be misleading. An extracted OP with a large amplitude will most probably produce a large error in comparison to a small amplitude signal producing a smaller error. The difference in error has nothing to do with the quality of the optimization, but is directly correlated to the shear magnitude of the OP signal and the difference squared component of equation 4.5. %RelAmp provides a relative amplitude of the extracted OP in comparison to the filtered positive cornea b-wave. The calculations show that the fellow eye ERGs generally contain OPs with both the largest envelope amplitude and %RelAmp, were eyes diagnosed as non profuse eyes had the largest % ErFit and smallest %RelAmp. Figure 4.3 is a scatter plot of %RelAmp versus % ErFit. The fellow eyes are clustered along a %ErFit vertical line of 18.84, with a distribution of %RelAmp along that line. The diagnosed eyes, with the exception of the least profuse cases, have a large % ErFit distribution to the right of the fellow eye's mean %ErFit. All the diagnosed eyes have a smaller %RelAmp. Ten of the twelve diagnosed eyes which went to neovascularization are clearly detectable from the non diagnosed eyes. Using multi-variant analysis, the diagnosed eyes can be scored based on the diagnosed eyes contained within the 2sd (standard deviation) oval (see figure 4.3).

Table 4.0 – Optimized OP model parameters for Dr. Brigell's fellow eyes.

OP_Model(t)= A*(t/Tc*exp(1-t/Tc))^N*sin(2*pi*F*t/1000+ANG*pi/180)
A = Amplitude of OP Model at peak of envelope
Tc = Center time of the OP Model envelope
F = Frequency of the modulation of the OP Model envelope
ANG = Phase shift of the modulation of the OP Model envelope
N = steepness of OP Model envelope
Atp = The amplitude of the low pass filtered positive cornea b-wave
Arel = A/Atp
Error = RMS Error of the difference of the extracted OP and OP model
%ErFit = (Error/A)*100
%RelAmp = (A/Atp)*100

Diagnosis Legend:

Eye	
N = Non Perfuse	P = Perfuse
I = Indeterminate Hemorrhage	F = Fellow Eye

NV	
0 = No neovascularization	X = neovascularization

CASE	Eye	NV	Atp	A	Arel	Tc	F	ANG	N	Error	%ErFit	%RelAmp
01058 L	F	0	406.8	56.9	0.14	20.84	139.02	0	12.5	8.97	15.75	13.99
01063 R	F	0	548.7	50.1	0.09	22.19	131.62	0	15.0	9.01	17.98	9.14
02048 R	F	0	488.9	51.7	0.11	22.22	130.80	0	21.4	7.21	13.94	10.57
02062 L	F	0	523.3	29.9	0.06	17.95	130.61	0	30.8	2.43	8.14	5.71
02064 L	F	0	248.9	19.5	0.08	17.47	133.25	0	34.1	1.85	9.51	7.83
02068 L	F	0	359.8	20.1	0.06	23.63	125.74	0	16.2	3.34	16.60	5.58
02081 R	F	0	327.3	17.6	0.05	23.40	121.99	0	17.9	2.39	13.55	5.38
03-044 L	F	0	448.3	50.0	0.11	23.49	124.03	0	31.1	5.03	10.06	11.15
03051 L	F	0	763.0	51.0	0.07	21.20	129.97	0	12.5	7.43	14.58	6.68
04090 L	F	0	386.6	35.6	0.09	21.37	102.46	0	16.0	5.70	16.01	9.21
050482 R	F	0	380.4	46.7	0.12	18.29	147.72	0	11.2	7.13	15.28	12.27
05059 L	F	0	320.0	51.4	0.16	22.02	133.41	0	9.1	8.20	15.97	16.05
05079 R	F	0	523.7	52.0	0.10	23.40	135.98	0	8.5	11.84	22.76	9.93
05081 R	F	0	214.8	21.7	0.10	31.41	154.12	0	9.4	7.42	34.20	10.11
050881 L	F	0	629.0	50.4	0.08	21.33	134.47	0	10.5	16.36	32.49	8.00
07054 L	F	0	302.9	11.8	0.04	22.60	136.50	0	6.5	3.63	30.84	3.89
07056 L	F	0	556.2	52.4	0.09	22.52	128.22	46	13.2	3.69	7.06	9.41
07082 L	F	0	634.4	81.9	0.13	20.31	132.32	0	16.5	10.01	12.23	12.91
80432 R	F	0	684.7	45.3	0.07	23.42	123.44	0	14.1	7.68	16.94	6.62
08056 R	F	0	568.9	48.1	0.08	20.11	127.60	0	12.4	13.34	27.73	8.45
08061 R	F	0	286.5	25.0	0.09	22.19	134.44	0	12.9	5.85	23.42	8.72
09060 R	F	0	100.0	12.0	0.12	21.09	96.20	252	16.4	3.02	25.29	11.95
09068 R	F	0	535.0	46.3	0.09	21.19	139.66	0	10.8	12.96	27.97	8.66
09112 L	F	0	408.7	44.8	0.11	19.64	133.63	0	11.8	9.78	21.82	10.97
09114 R	F	0	734.1	70.1	0.10	20.00	139.25	0	10.0	14.69	20.96	9.55
Mean:			**455.2**	**41.7**	**0.09**	**21.73**	**130.66**		**15.2**	**7.56**	**18.84**	**9.31**
Stdev:			**167.5**	**17.9**	**0.0**	**2.7**	**11.9**		**7.1**	**4.0**	**7.6**	**2.9**
2*stdev:			**334.9**	**35.7**	**0.1**	**5.3**	**23.7**		**14.3**	**8.1**	**15.3**	**5.7**

Table 4.0 (Continued) – Optimized OP model parameters for Dr. Brigell's eyes diagnosed indeterminate hemorrhage, non perfuse and perfuse. The arithmetic mean is calculated for each group of diagnosed eyes that went to neovascularization and those that did not go to completion. The overall mean of each group is also calculated.

CASE	Eye	NV	Atp	A	Arel	Tc	F	ANG	N	Error	%ErFit	%RelAmp
02064 R	I	0	415.7	6.9	0.02	17.85	118.77	0	38.0	1.75	25.41	1.66
03-044 R	I	0	205.0	18.3	0.09	30.62	133.72	0	41.0	4.65	25.46	8.92
05059 R	I	0	470.2	15.6	0.03	23.94	125.71	0	16.1	5.01	32.02	3.32
050881 R	I	0	470.0	47.6	0.10	25.68	84.63	0	52.5	7.20	15.12	10.13
08061 L	I	0	245.0	11.5	0.05	27.66	91.81	0	40.8	4.76	41.21	4.71
09068 L	I	0	349.1	20.2	0.06	19.30	106.65	0	36.7	5.36	26.59	5.77
Sub-Mean:			**359.1**	**20.0**	**0.06**	**24.18**	**110.22**		**37.5**	**4.79**	**27.64**	**5.75**
01063 L	I	1	350.6	33.4	0.10	24.25	89.25	0	16.5	12.83	38.43	9.53
02048 L	I	1	279.5	4.99	0.02	26.35	115.23	0	18.8	1.02	20.44	1.78
05079 L	I	1	167.6	4.86	0.03	23.11	98.11	0	31.3	0.83	17.14	2.9
07054 R	I	1	338.7	14.1	0.04	21.49	107.18	0	15.1	3.19	22.57	4.17
Sub-Mean:			**284.1**	**14.3**	**0.05**	**23.80**	**102.44**		**20.4**	**4.47**	**24.65**	**4.60**
Mean:			**329.1**	**17.7**	**0.05**	**24.03**	**107.11**		**30.7**	**4.66**	**26.44**	**5.29**
07056 R	N	0	510.8	17.2	0.03	23.81	115.65	0	15.2	3.31	19.3	3.36
01058 R	N	1	216.7	16.1	0.07	23.99	117.44	0.00	16	5.3	32.84	7.43
04090 R	N	1	166.6	16.8	0.10	20.95	99.78	0	18	3.21	19.12	10.09
050482 L	N	1	365.2	6.3	0.02	24.73	135.52	180	16.5	5.51	88.18	1.71
08056 L	N	1	306.8	4.4	0.01	23.34	175.44	180	15.6	2.62	59.48	1.43
09060 L	N	1	324.6	9.2	0.03	27.15	121.92	0	12.9	8.97	97.91	2.82
Sub-Mean:			**276.0**	**10.5**	**0.05**	**24.0**	**130.0**		**15.8**	**5.1**	**59.5**	**4.7**
Mean:			**315.1**	**11.6**	**0.04**	**24.0**	**127.6**		**15.7**	**4.8**	**52.8**	**4.5**
02081 L	P	0	363.2	7.63	0.02	18.24	127.7	195	85.6	1.22	16.05	2.1
07082 R	P	0	508.7	21.4	0.04	17.98	114.02	0	17.5	2.52	11.79	4.21
80432 L	P	0	795.1	36.4	0.05	28.14	117.52	0	14.2	6.49	17.83	4.58
09112 R	P	0	296.8	16.6	0.06	20.83	114.02	0	16.8	3.37	20.31	5.58
Sub-Mean:			**490.9**	**20.5**	**0.04**	**21.30**	**118.32**		**33.5**	**3.40**	**16.50**	**4.12**
02062 R	P	1	615.5	23.8	0.04	17.47	126.04	0	13.5	3.48	14.60	3.87
02068 R	P	1	527.3	17.0	0.03	24.02	121.97	0	20.1	10.85	63.81	3.22
03051 R	P	1	828.6	31.1	0.04	19.05	114.16	0	15.0	6.11	19.65	3.75
05081 L	P	1	426.5	38.5	0.09	16.46	135.42	0	14.9	15.14	39.29	9.03
09114 L	P	1	627.8	47.2	0.08	21.22	124.82	0	12.6	7.57	16.04	7.52
Sub-Mean:			**605.1**	**31.5**	**0.05**	**19.64**	**124.48**		**15.2**	**8.63**	**30.68**	**5.48**
Mean:			**554.4**	**26.6**	**0.05**	**20.38**	**121.74**		**23.3**	**6.31**	**24.37**	**4.87**

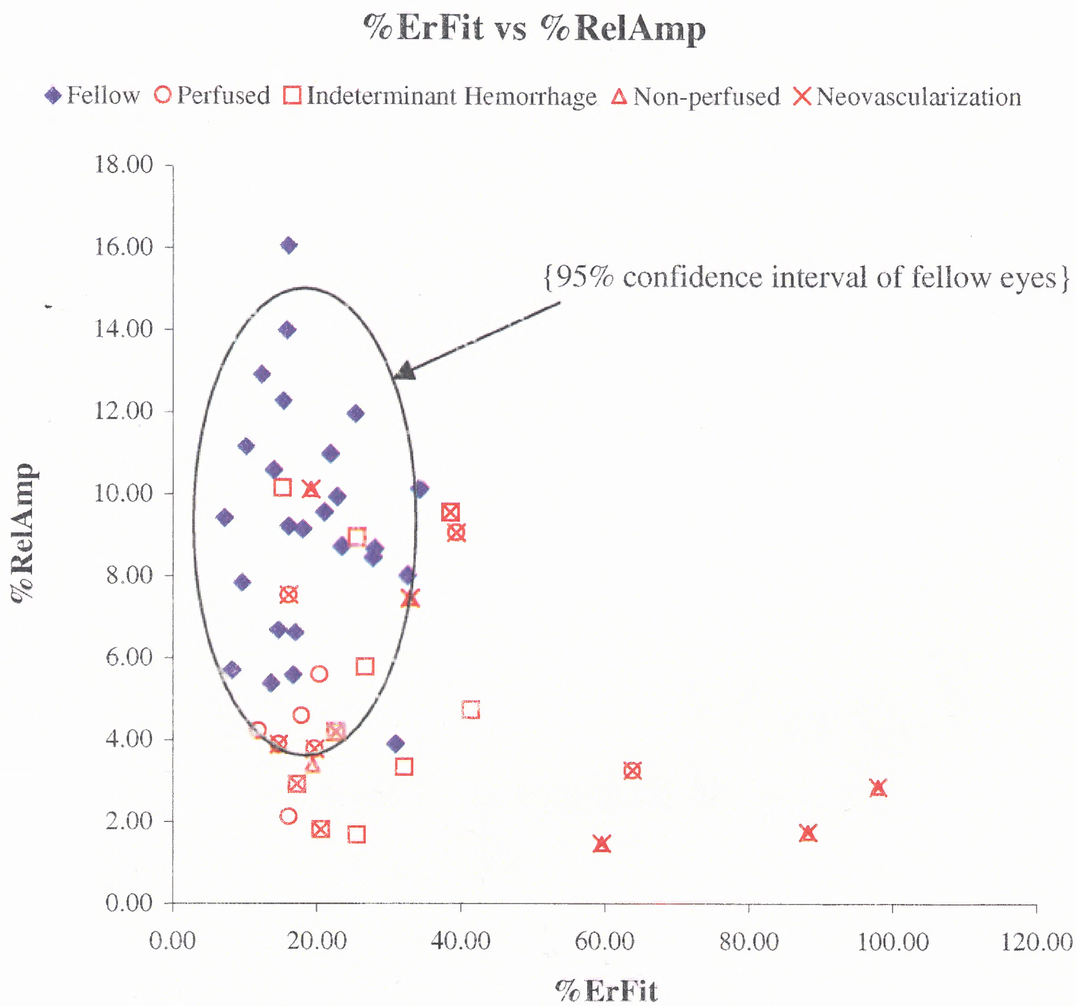


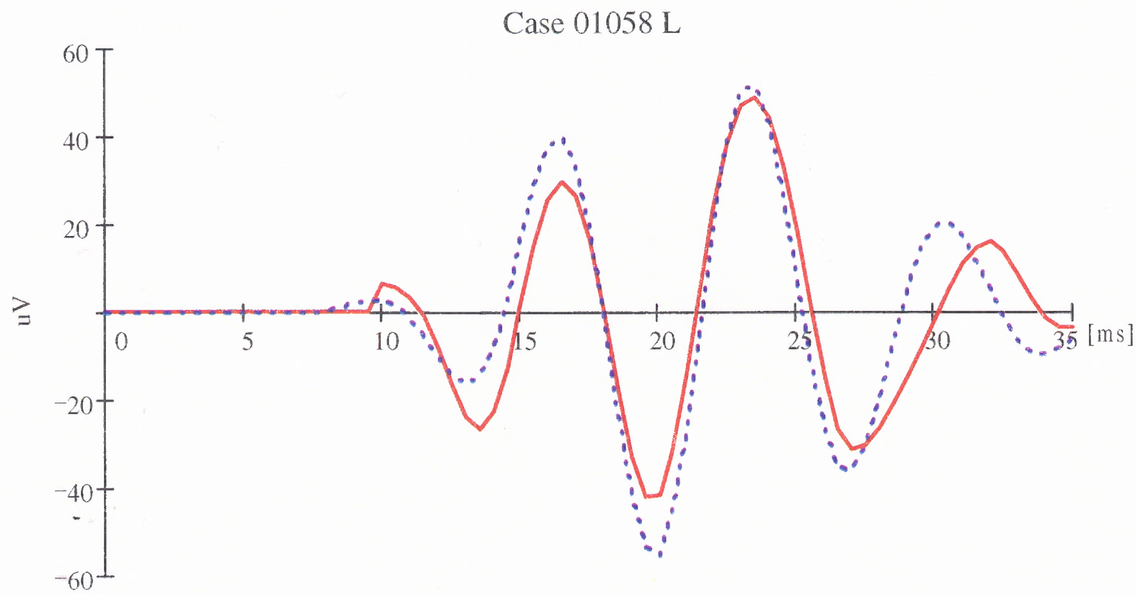
Figure 4.3 - Scatter plot of %ErFit vs %RelAmp. The OP Model's parameter optimization error (see equation 4.5), scaled by the OP's amplitude, is compared to the OP amplitude, relative to the positive cornea b-wave's peak to trough amplitude. Evaluating the plot, the 13 out of the 25 diagnosed eyes have a smaller OP amplitude compared to the fellow eyes, relative to the positive cornea b-wave's peak to trough amplitude.

CHAPTER 5

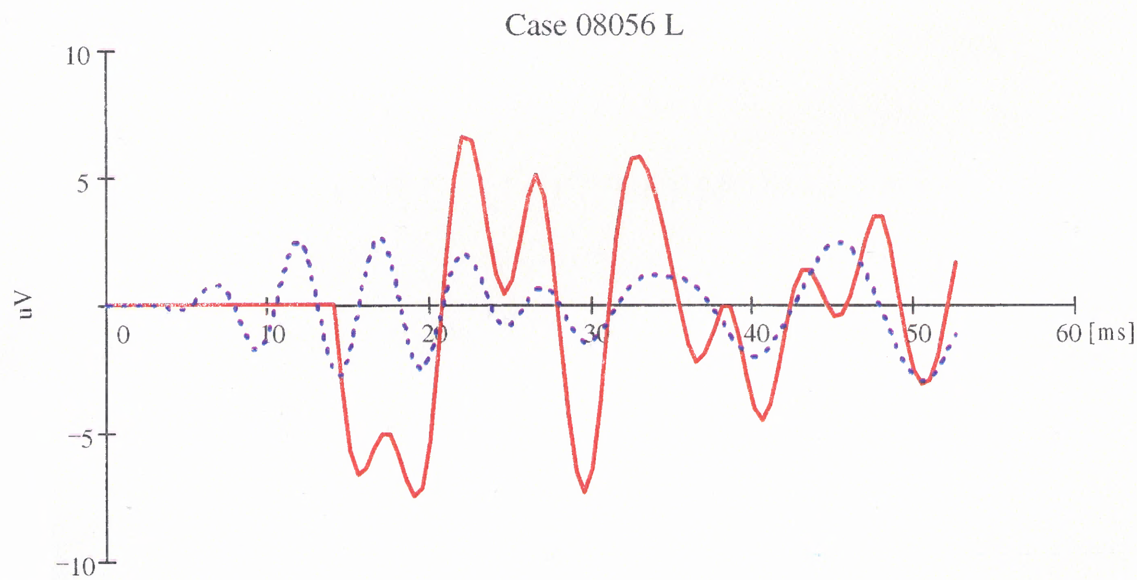
CONCLUSION

The proposed pre filtering windowing and signal conditioning technique, utilizing the "Oscillatory Potential Extraction Program", successfully extracted non distorted signals from Dr. Brigell's 50 electroretinograms. Some of the extracted signals did not have the appearance of an OP. Figure 5.0 illustrates two extremes of extracted signals. The signal (fellow eye) plotted in figure 5.0 a) has the characteristics of an OP, the signal is contained within a distinguishable OP model envelope. The signal (diagnosed eye which went to neovascularization completion) plotted in figure 5.0 b), which by inspection is clearly not an OP. When fitted to the OP model, the extracted signal produced a small amplitude, a long center time latency and a high frequency parameter.

Haupt et al [16] compared the proposed technique with extracted results from this research to a previous study completed by Haupt et al [13]. The current results produced larger envelope amplitudes and lower frequencies (lengthened half-wavelengths) compared to previous study. The signals with increased half-wavelengths appear more frequently in the diagnosed eyes which went to neovascularization completion. This result suggests that the proposed technique can extract both a non distorted OP and a pathological significant signal, which may be a severely degraded OP or not an OP at all.



a)



b)

Figure 5.0 - Two extremes of extracted signals. The signal (fellow) in **a)** has the characteristics of an OP, the signal is contained within a distinguishable OP model envelope. The diagnosed eye, which went to neovascularization completion, in b), when fitted to the OP model, produced a small amplitude, a long center time latency and a high frequency parameter.

CHAPTER 6

SUGGESTIONS FOR FUTURE RESEARCH

Prognosis and early diagnosis of circulatory disorders of the inner retina, such as diabetic retinopathy and central retina vein occlusion, may minimize the impact of such diseases on the pathological state of the retina. Since the oscillatory potential (OP) may be the indicator for these diseases, any method which improves non evasive extraction of the OP or provides further insight into the source of the OP warrants further research.

The results obtained from the proposed pre filtering signal conditioning technique used in conjunction with the single envelope OP model provides parameter separation between non diagnosed and diagnosed eyes. However, since separation of the pathological eyes was not obtained, the results are not conclusive. Reasons that pathological separation was not obtained may be attributed to the possible truncation of the OP during the windowing process or the OP model is not fitting the extracted OP's endpoints accurately. Since the method was tested on a single source database of fifty ERGs, it is suggested that further ERGs which are obtained from multiple sources be processed. Extensive ERG processing will aid in the determination of optimal initial conditions for the parameter optimization algorithm. Also, it is suggested that variations of the OP model be evaluated. Since the proposed technique is applicable to stored ERG data, it may also prove beneficial to research improvements to real time data processing of the ERG and the OP.

APPENDIX A

A.1 Electroretinogram Data Format

Table A.0 illustrates the format of the ERG data file. The first line contains header information about the case. Successive lines are row vectors containing sampling time and the measured ERG voltage. The Dr. Brigell's ERG's, usually contains pre-stimuli measurements of about 20 milliseconds for the determination of sensor offset. The data file contains 235.5 milliseconds of post stimuli data. Typically, each ERG data file has 512 data points

Table A.0 - Electroretinograph data file format. The first line contains patient case information. The following rows comprise of the data matrix. The data is formatted with the time vector in the first column and the signal vector in the second column.

```
01058        $ gais        iv        _ L
-20.000000,  0.0
-19.500000,  25.609756
-19.000000,  25.609756
-18.500000,  34.146341
   .
   .
   .
234.500000,  34.146341
235.000000,  42.682927
235.500000,  59.756098
```

A.2 State Space Model for Direct Form Type II Bandpass Digital Filter

Reference figure 3.0 for the realization. The state equations are defined as followed:

$$x_1(k+1) = -a2 * x_1(k) + x_2(k) + (-a2 * b1 + b2) * u(k)$$
$$x_2(k+1) = -a3 * x_1(k) + x_3(k) + (-a3 * b1 + b3) * u(k)$$
$$x_3(k+1) = -a4 * x_1(k) + x_4(k) + (-a4 * b1 + b4) * u(k)$$
$$x_4(k+1) = -a5 * x_1(k) + (-a5 * b1 + b5) * u(k)$$

$$y(k) = x_1(k) + b1 * u(k)$$

Table A.1 – Direct form type II bandpass digital filter's state space variable values after being stimulated by a unit impulse. Reference figure 3.1 for a plot of the state variables

4th Order Butterworth Digital Filter

Values of State Variables After Filter is Stimulated by an Unit Impulse

k	u(k)	$x_4(k+1)$	$x_4(k)$	$x_3(k+1)$	$x_3(k)$	$x_2(k+1)$	$x_2(k)$	$x_1(k+1)$	$x_1(k)$	y(k)
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	1	0.0331	0	0.1132	0	-0.3214	0	0.1739	0	0.0591
6	0	-0.0766	0.0331	0.3662	0.1132	-0.4843	-0.3214	0.1904	0.1739	0.1739
7	0	-0.0839	-0.0766	0.2881	0.3662	-0.288	-0.4843	0.0761	0.1904	0.1904
8	0	-0.0335	-0.0839	0.0619	0.2881	0.0266	-0.288	-0.064	0.0761	0.0761
9	0	0.0282	0.0335	-0.1561	0.0619	0.2818	0.0266	-0.1618	-0.064	-0.064
10	0	0.0713	0.0282	-0.2817	-0.1561	0.3997	0.2818	-0.1943	-0.1618	-0.1618
11	0	0.0856	0.0713	-0.3009	-0.2817	0.3859	0.3997	-0.1722	-0.1943	-0.1943
12	0	0.0759	0.0856	-0.2443	-0.3009	0.2907	0.3859	-0.1209	-0.1722	-0.1722
13	0	0.0533	0.0759	-0.1557	-0.2443	0.1711	0.2907	-0.0651	-0.1209	-0.1209
14	0	0.0287	0.0533	-0.0715	-0.1557	0.0658	0.1711	-0.0205	-0.0651	-0.0651
15	0	0.009	0.0287	-0.0106	-0.0715	0.001	0.0658	0.0054	-0.0205	-0.0205
16	0	-0.0024	0.009	0.0194	-0.0106	-0.0293	0.001	0.017	0.0054	0.0054
17	0	-0.0075	-0.0024	0.0302	0.0194	-0.039	-0.0293	0.0207	0.017	0.017
18	0	-0.0091	-0.0075	0.0322	0.0302	-0.0411	-0.039	0.022	0.0207	0.0207
19	0	-0.0097	-0.0091	0.0331	0.0322	-0.0435	-0.0411	0.0238	0.022	0.022
20	0	-0.0105	-0.0097	0.0358	0.0331	-0.0486	-0.0435	0.0265	0.0238	0.0238
21	0	-0.0117	-0.0105	0.0402	0.0358	-0.0551	-0.0486	0.0293	0.0265	0.0265
22	0	-0.0129	-0.0117	0.0444	0.0402	-0.0604	-0.0551	0.0311	0.0293	0.0293
23	0	-0.0137	-0.0129	0.0467	0.0444	-0.0624	-0.0604	0.0311	0.0311	0.0311
24	0	-0.0137	-0.0137	0.0459	0.0467	-0.0602	-0.0624	0.0291	0.0311	0.0311
25	0	-0.0128	-0.0137	0.0421	0.0459	-0.0542	-0.0602	0.0256	0.0291	0.0291
26	0	-0.0113	-0.0128	0.0362	0.0421	-0.0459	-0.0542	0.0211	0.0256	0.0256
27	0	-0.0093	-0.0113	0.0292	0.0362	-0.0365	-0.0459	0.0163	0.0211	0.0211
28	0	-0.0072	-0.0093	0.022	0.0292	-0.027	-0.0365	0.0116	0.0163	0.0163
29	0	-0.0051	-0.0072	0.015	0.022	-0.0179	-0.027	0.0072	0.0116	0.0116
30	0	-0.0032	-0.0051	0.0087	0.015	-0.0097	-0.0179	0.0032	0.0072	0.0072
31	0	-0.0014	-0.0032	0.003	0.0087	-0.0024	-0.0097	-0.0002	0.0032	0.0032
32	0	0.0001	-0.0013	-0.0018	0.003	0.0037	-0.0024	-0.003	-0.0002	-0.0002
33	0	0.0013	0.0001	-0.0056	-0.0018	0.0085	0.0037	-0.0057	-0.003	-0.003

APPENDIX B

The following modules are required for the execution of the MATLAB program erg.m. The program must be ran in a version of MATLAB that supports object oriented programming and advanced data structures such as cells, structures and multidimensional arrays.

B.1 erg.m

```
% Main program for extracting an oscillatory potential (op) from
% an electroretinogram (erg). The op can be extracted automatically
% or manually.
%
% AUTOMATIC MODE:
% In the automatic mode, the user is prompted to input
% the directory of erg files to have potential ops extracted.
% In the directory, there must be an index file termed "Data.txt".
% "Data.txt" contains a list of the erg file names. The user
% must select an op model. {Currently the program is limited to a
% twelve parameter model}
%
% MANUAL MODEL
% In the manual mode, the user is walked through the steps of extracting
% the single op with the ability to modify windowing, filter's bandpass
% frequencies and initial conditions of the op model.
%
% Erg.m handles menuing and program flow. Whether in automatic or
% manual processing, the op extraction program performs parameter
% estimation of a five parameter op model. The parameters are
% determined by minimizing the error between the op and op model. The
% parameter estimation error is the squareroot of the average of the
% sum of the squared difference of the op model and the extracted op.
%
% PARAMETER ESTIMATION ERROR:
%       e(t)={(1/n)*sum[m(t)-OP(t)]^2}^(1/2)  t=0...n-1
%
% Global variables generated in Erg.m:
%   METHOD:  Sets extraction method chosen 1)=Manual and 2) Automatic
%
% Peter Derr, Dr Andrew Meyer and Dr Edward Haupt 11/18/97
% Rev. A 12/3/97
```

```
% Rev. B 07/13/98
% Rev. C 08/19/98
% Required M-Files: auto.m and manual.m
%==========================================================

% GLOBAL VARIABLES - Since the program uses extensive recursive windowing,
% global variable for each m-file that creates a multiple occurrences of
% its window must have its global variables defined in the calling
% m-file.
%------------------------------------------------------------------------
%erg.m generated global variables:
global METHOD
%------------------------------------------------------------------------
main_win=figure( ...
  'Name','Oscllatory Potential Extraction Program', ...
  'NumberTitle', 'off', ...
  'Units', 'normalized', ...
  'MenuBar', 'none', ...
  'BackingStore','off', ...
  'Position',[.01 .065 .98 .88]);
%------------------------------------------------------------------------
control_panel=uicontrol(...
  'Style','frame', ...
  'Units','normalized', ...
  'Position',[.79 .02 .20 .95], ...
  'BackgroundColor', [0.50 0.50 0.50]);
%------------------------------------------------------------------------
manual_str=[ ...
     'METHOD=1;' ...
     'manual(''initialize'');'];
manual_win=uicontrol(...
  'String','Manual', ...
  'Style','pushbutton', ...
  'Units','normalized', ...
  'Position',[.81 .2 .16 .075], ...
  'CallBack','Manual(''initialize'')', ...
  'Visible', 'on', ...
  'Enable','on', ...
  'BackgroundColor', [0.50 0.50 0.50], ...
  'CallBack', manual_str);
%------------------------------------------------------------------------
auto_str=[ ...
    'METHOD=2;' ...
    'auto;'];
auto_win=uicontrol(...
  'String','Automatic', ...
```

```
  'Style','pushbutton', ...
  'Units','normalized', ...
  'Position',[.81 .115 .16 .075], ...
  'Visible', 'on', ...
  'Enable','on', ...
  'BackgroundColor', [0.50 0.50 0.50], ...
  'CallBack', auto_str);
%----------------------------------------------------------------------
quit_str=[ ...
    'clear all;' ...
    'close(gcf);'];
quit_win=uicontrol(...
  'String','Quit', ...
  'Style','pushbutton', ...
  'Units','normalized', ...
  'Position',[.81 .03 .16 .075], ...
  'Visible', 'on', ...
  'Enable','on', ...
  'BackgroundColor', [0.50 0.50 0.50], ...
  'CallBack',quit_str);
%----------------------------------------------------------------------
title_message=uicontrol( ...
  'Style','text', ...
  'Units','normalized', ...
  'Position', [.008 .89 .772 .08], ...
  'horizontalAlignment','center', ...
  'FontSize', 12, ...
  'FontWeight','Bold', ...
  'BackgroundColor', [1 1 1], ...
  'Enable','on');
%----------------------------------------------------------------------
title_error=[ ...
    'Program could open ergtitle.txt, If the path is incorrect, ' ...
    'modify line 121 of erg.m to point to the correct path'];
title_fid=fopen('c:\matlab\toolbox\erg\auto\ergtitle.txt','rt');
if title_fid == -1
  set(title_message,'String',title_error);
else
  title_text=fscanf(title_fid,'%c',inf);
  fclose(title_fid);
  set(title_message,'String',title_text);
end
%----------------------------------------------------------------------
author_message=uicontrol( ...
  'Style','text', ...
  'Units','normalized', ...
```

```
  'Position', [.008 .62 .772 .27], ...
  'horizontalAlignment','center', ...
  'FontSize', 10, ...
  'FontWeight','Bold', ...
  'BackgroundColor', [1 1 1], ...
  'Enable','on');
%-------------------------------------------------------------------------
author_error=[ ...
    'Program could open ergauthor.txt, If the path is incorrect, ' ...
    'modify line 134 of erg.m to point to the correct path'];
author_fid=fopen('c:\matlab\toolbox\erg\auto\ergauthor.txt','rt');
if author_fid == -1
  set(author_message,'String',author_error);
else
  author_text=fscanf(author_fid,'%c',inf);
  fclose(author_fid);
  set(author_message,'String',author_text);
end
%-------------------------------------------------------------------------
text_message=uicontrol( ...
  'Style','text', ...
  'Units','normalized', ...
  'Position', [.008 .02 .772 .60], ...
  'horizontalAlignment','left', ...
  'FontSize', 7, ...
  'FontWeight','normal', ...
  'BackgroundColor', [1 1 1], ...
  'Enable','on');
%-------------------------------------------------------------------------
text_error=[ ...
    'Program could open ergtext.txt, If the path is incorrect, ' ...
    'modify line 156 of erg.m to point to the correct path'];
text_fid=fopen('c:\matlab\toolbox\erg\auto\ergtext.txt','rt');
if text_fid == -1
  set(text_message,'String',text_error);
else
  text_text=fscanf(text_fid,'%c',inf);
  fclose(text_fid);
  set(text_message,'String',text_text);
end
```

B.2 auto.m

```
function auto(action)
% The procedure automatically extracts the oscillatory potential (op)
% from the electroretinogram (erg). Using the simplex method (fmins.m),
% The procedure determines an optimum parameter vector that minimizes the
% error between the extracted op and the selected op model. The procedure
% is called by erg.m. Auto.m has a graphical user interface. The user
% selects a desired model and then presses the "Start" button. When the
% process is completed, the user can press the "Done". The program saves
% two types of files in the currently selected erg directory:
% 1) An ascii file named "Evaluate.dat" that contains:
%       A) evaluator's name.
%       B) date/time of evaluation.
%       C) op model used for evaluation.
%       D) erg header information.
%       E) filter's corner frequencies.
%       F) sampling frequency of the erg data.
%       G) optimized op model's parameter vector.
%       H) error between extracted op and the op model.
%       I) %ERFIT = The relation of the op model fitting error
%                   and the size of the b-wave.
%       J) %RELFIT = The relation of the op model envelope size
%                   and the size of the b-wave.
% 2) An ascii files named "{erg file name}.op" that contains:
%       A) evaluator's name.
%       B) date/time of evaluation.
%       C) [TIME OP OP MODEL] matrix.
%
% NOTE - Every time auto.m is executed the previous versions of the .dat
% and .op files will be overwritten. Be sure to move them or rename them
% before execution. Both files can be imported into a spreadsheet.
%
% Peter Derr, Dr Andrew Meyer and Dr Edward Haupt 11/18/97
% Rev.A 12/3/97
% Rev.B 9/21/98
% Required M-Files: rfile.m, fergferg.m and optparam.m
%===================================================

% Global variables generated in Manual.m:
%     AUTO_WIN:      Main figure object.
%     DIRECTORY_WIN: Directory text object.
%     DIRECTORY_TXT: Directory object.
%     TIME_WIN:      Process time text object.
%     TIME_TXT:      Process object.
%     PERCENT_WIN:   Percent text object.
```

```
%   PERCENT_TXT:   Percent object.
%   STATUS_TXT:    Percent active object.
%   MINPERC_TXT:   Minimum percent text object
%   MAX_PERC_TXT:  Maximum percent text object
%   START_WIN:     Start button object.
%   ERG_DATA:      Erg data matrix.
%   ERG_SIGNAL:    Erg signal vector.
%   ERG_TIME:      Erg time vector.
%   SAMPLE_FREQ:   Sampling frequency of erg signal.
%   TIME_ZERO:     Index of time vector for t = 0.
%   LENGTH_SIGNAL: Length of erg signal.
%   FH:            Filter's highpass frequency.
%   FL:            Filter's lowpass frequency.
%   P_INIT:        Initial starting vector for optimiztion.
%   PATHNAME:      Path name of directory of ergs to process.
%   TXT_NAME:      erg filename.
%   INDEX_FILE:    Concatenation of PATHNAME and TXT_NAME.
%   PROC_TIME:     Process time variable.
%   MODELDES_WIN:  Model popup object.
%   MODELDES_TXT:  Model popup text object.
%   RECORD:        Model's record number.
%   TEMP_MODEL:    OP model.
%   TEMP_VAR:      OP model variables
%   TEMP_ERR:      OP model parameter used to calculate %ERFIT and %RELAMP.
%   MODEL:         OP model database.
%   MAXPARAM:      Maximum parameters in OP model.
%   HEADER:        Erg file header information
%   TEMP_TITLE:    OP model variable names.
%   PRINT_MODEL:   OP model for printing to file.
%   NAME_WIN:      Name of person processing ergs object.
%   NAME_TXT:      Text Name of person processing ergs object text.
%   DATE_WIN:      Date object.
%   DATE_TXT:      Editable Date object.
%==========================================================
global ERG_DATA ERG_SIGNAL ERG_TIME SAMPLE_FREQ P_INIT TIME_ZERO
global METHOD AUTO_WIN DIRECTORY_WIN DIRECTORY_TXT TIME_WIN
global TIME_TXT PERCENT_WIN PERCENT_TXT MINPERC_TXT
global MAX_PERC_TXT STATUS_TXT TXT_NAME INDEX_FILE
global PATHNAME PROC_TIME MODELDES_WIN RECORD TEMP_MODEL
global TEMP_VAR B A MODELDES_TXT MODEL START_WIN MAXPARAM
global HEADER LENGTH_SIGNAL FH FL TEMP_ERR TEMP_TITLE
global PRINT_MODEL NAME_WIN NAME_TXT DATE_WIN DATE_TXT
%---------------------------------------------------------------------
% global variables generated by fergferg.m
global APP OP
%---------------------------------------------------------------------
```

```
% global variables generated by optparam.m
global M_OUT E_OPT
%------------------------------------------------------------------------
if nargin < 1
  action='initialize';
end
if strcmp(action,'initialize')
  AUTO_WIN=figure( ...
    'Name','AUTOMATIC OP EXTRACTION MODE', ...
    'Units', 'normalized', ...
    'Position',[.01 .065 .98 .88], ...
    'NumberTitle','off', ...
    'Pointer','arrow', ...
    'MenuBar','none', ...
    'Visible','on', ...
    'BackingStore', 'on');
  %------------------------------------------------------------------------
  MODELDES_TXT=uicontrol(...
    'Style','text', ...
    'Units','normalized', ...
    'FontSize',6, ...
    'FontWeight','Bold', ...
    'Position',[.01 .962 .98 .03], ...
    'BackgroundColor', [0 1 0], ...
    'String','CHOOSE AN OP MODEL', ...
    'Visible','on');
  %------------------------------------------------------------------------
  MODELDES_WIN=uicontrol(...
    'Style','popup', ...
    'Units','normalized', ...
    'FontSize',6, ...
    'FontWeight','Bold', ...
    'Position',[.01 .932 .98 .03], ...
    'BackgroundColor', [1 1 1], ...
    'String','INITIALIZING', ...
    'Visible','on', ...
    'CallBack','auto("getmodel")');
  %------------------------------------------------------------------------
  NAME_WIN=uicontrol(...
    'Style','text', ...
    'Units','normalized', ...
    'FontSize',8, ...
    'FontWeight','Bold', ...
    'horizontalAlignment','left', ...
    'Position',[.01 .768 .30 .042], ...
    'String', 'EVALUATORS NAME :');
```

```
%----------------------------------------------------------------------
NAME_TXT=uicontrol(...
  'Style','edit', ...
  'Units','normalized', ...
  'FontSize',10, ...
  'FontWeight','Normal', ...
  'BackgroundColor', [1 1 1], ...
  'horizontalAlignment','left', ...
  'Position',[.31 .768 .40 .042]);
%----------------------------------------------------------------------
DATE_WIN=uicontrol(...
  'Style','text', ...
  'Units','normalized', ...
  'FontSize',8, ...
  'FontWeight','Bold', ...
  'horizontalAlignment','left', ...
  'Position',[.01 .732 .30 .042], ...
  'String', 'DATE PROCESSED:');
%----------------------------------------------------------------------
DATE_TXT=uicontrol(...
  'Style','edit', ...
  'Units','normalized', ...
  'FontSize',10, ...
  'FontWeight','Normal', ...
  'BackgroundColor', [1 1 1], ...
  'horizontalAlignment','left', ...
  'Position',[.31 .732 .40 .042]);
%----------------------------------------------------------------------
DIRECTORY_WIN=uicontrol(...
  'Style','text', ...
  'Units','normalized', ...
  'FontSize',8, ...
  'FontWeight','Bold', ...
  'horizontalAlignment','left', ...
  'Position',[.01 .696 .30 .042], ...
  'String', 'PROCESSING DIRECTORY:');
%----------------------------------------------------------------------
DIRECTORY_TXT=uicontrol(...
  'Style','text', ...
  'Units','normalized', ...
  'FontSize',10, ...
  'FontWeight','Normal', ...
  'horizontalAlignment','left', ...
  'Position',[.31 .696 .40 .042]);
%----------------------------------------------------------------------
TIME_WIN=uicontrol(...
```

```
  'Style','text', ...
  'Units','normalized', ...
  'FontSize',8, ...
  'FontWeight','Bold', ...
  'horizontalAlignment','left', ...
  'Position',[.01 .660 .30 .042], ...
  'String', 'PROCESS TIME REMAINING:');
%-----------------------------------------------------------------------
TIME_TXT=uicontrol(...
  'Style','text', ...
  'Units','normalized', ...
  'FontSize',10, ...
  'FontWeight','Normal', ...
  'horizontalAlignment','left', ...
  'Position',[.31 .660 .40 .042]);
%-----------------------------------------------------------------------
PERCENT_WIN=uicontrol(...
  'Style','text', ...
  'Units','normalized', ...
  'FontSize',8, ...
  'FontWeight','Bold', ...
  'horizontalAlignment','right', ...
  'Position',[.01 .508 .30 .042], ...
  'String', 'PERCENT COMPLETED:');
%-----------------------------------------------------------------------
MINPERC_TXT=uicontrol(...
  'Style','text', ...
  'Units','normalized', ...
  'FontSize',10, ...
  'FontWeight','Normal', ...
  'horizontalAlignment','left', ...
  'Position',[.31 .551 .024 .03], ...
  'String','0%');
%-----------------------------------------------------------------------
MAXPERC_TXT=uicontrol(...
  'Style','text', ...
  'Units','normalized', ...
  'FontSize',10, ...
  'FontWeight','Normal', ...
  'horizontalAlignment','right', ...
  'Position',[.67 .551 .04 .03], ...
  'String','100%');
%-----------------------------------------------------------------------
PERCENT_TXT=uicontrol(...
  'Style','text', ...
  'Units','normalized', ...
```

```
  'FontSize',10, ...
  'FontWeight','Normal', ...
  'horizontalAlignment','left', ...
  'Position',[.31 .508 .40 .042], ...
  'BackgroundColor', [1 1 1]);
%----------------------------------------------------------------------
STATUS_TXT=uicontrol(...
  'Style','text', ...
  'Units','normalized', ...
  'FontSize',10, ...
  'FontWeight','Normal', ...
  'horizontalAlignment','left', ...
  'Position',[.31 .508 .01 .042], ...
  'BackgroundColor', [1 0 1]);
%----------------------------------------------------------------------
START_WIN=uicontrol( ...
  'String','Start', ...
  'Style','pushbutton', ...
  'Units','normalized', ...
  'Position',[.66 .42 .16 .068], ...
  'Enable','off', ...
  'Callback', 'auto("start");');
%----------------------------------------------------------------------
STOP_WIN=uicontrol( ...
  'String','Done', ...
  'Style','pushbutton', ...
  'Units','normalized', ...
  'Position',[.83 .42 .16 .068], ...
  'Enable','on', ...
  'Callback','auto("housekeeping");');
%----------------------------------------------------------------------
text_message=uicontrol( ...
  'Style','text', ...
  'Units','normalized', ...
  'Position', [.008 .02 .984 .37], ...
  'horizontalAlignment','left', ...
  'FontSize', 7, ...
  'FontWeight','normal', ...
  'BackgroundColor', [1 1 1], ...
  'Enable','on');
%----------------------------------------------------------------------
text_error=[ ...
    'Program could open aut.txt, If the path is incorrect, ' ...
    'modify line 271 of auto.m to point to the correct path'];
text_fid=fopen('c:\matlab\toolbox\erg\auto\auto.txt','rt');
if text_fid == -1
```

```
    set(text_message,'String',text_error);
  else
    text_text=fscanf(text_fid,'%c',inf);
    fclose(text_fid);
    set(text_message,'String',text_text);
  end
  now_date=now;
  convert_date=datestr(now_date,0);
  set(DATE_TXT,'String',convert_date);
  drawnow;
  auto('getindex');
  %-----------------------------------------------------------------------
elseif strcmp(action,'getindex')
  [TXT_NAME,PATHNAME]=uigetfile('C:\Matlab\Toolbox\Erg\Data.txt','ERG INDEX
                                               FILE');
  INDEX_FILE=[PATHNAME,TXT_NAME];
  temp_path=PATHNAME;
  n=length(PATHNAME);
  temp_path(n)=[];
  set(DIRECTORY_TXT,'String',temp_path);
  del_file=[PATHNAME,'Evaluate.dat'];
  delete(del_file);
  clear n temp_path del_file
  auto('modelmenu')
elseif strcmp(action,'modelmenu')
  menu_str='';
  model_ct=getnumod;
  if model_ct > 0
    load 'c:\matlab\toolbox\erg\auto\model.txt' MODEL -mat;
    for index=1:model_ct-1
      menu_str=[menu_str,MODEL(index).des,'|'];
    end
    menu_str=[menu_str,MODEL(model_ct).des];
  else
    menu_str='CREATE MODELS';
  end
  set(MODELDES_WIN,'Visible','on','String',menu_str);
  clear index model_ct menu_str getnumodel
elseif strcmp(action,'getmodel')
  RECORD=get(MODELDES_WIN,'Value');
  set(MODELDES_WIN,'Visible','off');
  set(MODELDES_TXT,'Visible','off');
  TEMP_MODEL=char(eval(MODEL(RECORD).equ));
  PRINT_MODEL=TEMP_MODEL;
  TEMP_VAR=eval(MODEL(RECORD).var);
  TEMP_ERR=eval(char(MODEL(RECORD).err));
```

```
  temp_ini=eval(char(MODEL(RECORD).ini));
  TEMP_TITLE=eval(MODEL(RECORD).title);
  format short
  MAXPARAM=max(size(TEMP_VAR));
  for index=1:MAXPARAM
    tmp_param=['pm(',num2str(index),')'];
    TEMP_MODEL=subs(TEMP_MODEL,TEMP_VAR{index},tmp_param);
    P_INIT(index)=temp_ini(index);
  end
  clear index
  set(START_WIN,'Enable','on');
elseif strcmp(action,'start')
  fclose('all');
  store_file=[PATHNAME,'Evaluate.dat'];
  paramid=fopen(store_file,'wt');
  fprintf(paramid,'%30s \n\n',get(NAME_TXT,'String'));
  fprintf(paramid,'%30s \n\n',get(DATE_TXT,'String'));
  fprintf(paramid,'%7s %50s\n\n','M(t)=',char(PRINT_MODEL));
  fclose(paramid);
  paramid=fopen(store_file,'At');
  fprintf(paramid,'%35s %7s %7s %9s %9s','FILE
          TITLE','F(high)','F(low)','F(samp)','Atp');
  clear index
  for index=1:MAXPARAM
    fprintf(paramid,'%11s',char(TEMP_TITLE(index)));
  end
  fprintf(paramid,'%10s %9s %9s \n','Error','%ERFIT','%RELAMP');
  fclose(paramid);
  indexid=fopen(INDEX_FILE,'rt');
  filect=0;
  fseek(indexid,0,'bof');
  while feof(indexid)~=1
    fgetl(indexid);
    filect=filect+1;
  end
  load 'C:\matlab\toolbox\erg\auto\avgtime' -mat;
  PROC_TIME=filect*avg;
  gettime;
  count=0;
  fseek(indexid,0,'bof');
  while feof(indexid)~=1
    tic;
    count=count+1;
    clear ergfile ergfilename fileid;
    ergfile=fscanf(indexid,'%s',1);
    ergfilename=[PATHNAME,ergfile];
```

```
fileid=fopen(ergfilename,'rt');
[HEADER,ERG_DATA]=rfile(fileid);
fclose(fileid);
ERG_TIME=ERG_DATA(:,1);
ERG_SIGNAL=ERG_DATA(:,2);
LENGTH_SIGNAL=length(ERG_SIGNAL);
for index=1:LENGTH_SIGNAL
  if ERG_TIME(index) >= 0
    TIME_ZERO=index;
    break;
  end
end
if TIME_ZERO >= 3
  temp_signal=ERG_SIGNAL-mean(ERG_SIGNAL(2:TIME_ZERO-1));
  ERG_SIGNAL=temp_signal;
end
clear index temp_signal;
sample_time=(ERG_TIME(TIME_ZERO+1)-ERG_TIME(TIME_ZERO))/1000;
SAMPLE_FREQ=(sample_time)^(-1);
FH=65;
FL=250;
[B,A]=butter(2,[FH FL]*2/SAMPLE_FREQ);
fergferg;
[P_OPT]=optparam(P_INIT);
length_ergname=length(ergfilename);
for index=1:length_ergname
  if strcmp(ergfilename(index),'.')
    dot_locate=index;
    break;
  end
end
clear index
ergfilename(dot_locate)=[];
new_op_name=[ergfilename,'.op'];
clear index length_ergname ergfilename
erfit=(E_OPT/P_OPT(TEMP_ERR))*100;
relamp=(P_OPT(TEMP_ERR)/APP)*100;
fopid=fopen(new_op_name,'wt');
fprintf(fopid,'%30s \n\n',get(NAME_TXT,'String'));
fprintf(fopid,'%30s \n\n',get(DATE_TXT,'String'));
fprintf(fopid,'%8s %8s %8s \n','Time(ms)','OP(uV)','MODEL(uV)');
fclose(fopid);
M_OUT=M_OUT(:);
opsize=max(size(OP));
opmod=[OP(1,1) OP(1,2) M_OUT(1)];
fopid=fopen(new_op_name,'At');
```

```
    fprintf(fopid,'%8.2f %8.2f %8.2f \n',opmod);
    fclose(fopid);
    for index=2:opsize
      opmod=[OP(index,1) OP(index,2) M_OUT(index)];
      fopid=fopen(new_op_name,'At');
      fprintf(fopid,'%8.2f %8.2f %8.2f \n',opmod);
      fclose(fopid);
    end
    paramid=fopen(store_file,'At');
    format_popt=['%35s %7.2f %7.2f %9.2f %9.2f'];
    for index=1:MAXPARAM
      format_popt=[format_popt,' ','%10.2f'];
    end
    format_popt=[format_popt,' ','%9.2f %9.2f %9.2f\n'];
    fprintf(paramid,format_popt,HEADER,FH,FL,SAMPLE_FREQ,APP,P_OPT,
            E_OPT, erfit,relamp);
    fclose(paramid);
    timer=toc;
    load 'C:\matlab\toolbox\erg\auto\avgtime' -mat;
    if numsamp==0
      avg=timer;
    else
      avg=((avg*numsamp)+timer)/(numsamp+1);
    end
    numsamp=numsamp+1;
    PROC_TIME=(filect-count)*avg;
    gettime;
    save 'C:\matlab\toolbox\erg\auto\avgtime' avg numsamp -mat;
    total_time=filect*avg;
    done_perc=((total_time-PROC_TIME)/total_time)*.40;
    set(STATUS_TXT,'Position',[.31 .508 done_perc .042]);
    drawnow;
  end
elseif strcmp(action,'housekeeping')
  fclose('all');
  clear ERG_DATA ERG_SIGNAL ERG_TIME SAMPLE_FREQ P_INIT ...
  TIME_ZERO METHOD AUTO_WIN DIRECTORY_WIN DIRECTORY_TXT ...
  TIME_WIN TIME_TXT PERCENT_WIN PERCENT_TXT MINPERC_TXT ...
  MAX_PERC_TXT STATUS_TXT TXT_NAME INDEX_FILE ...
  PATHNAME PROC_TIME MODELDES_WIN RECORD TEMP_MODEL
  TEMP_VAR MODELDES_TXT MODEL START_WIN MAXPARAM ...
  HEADER LENGTH_SIGNAL FH FL TEMP_ERR TEMP_TITLE
...PRINT_MODEL;
  close(gcf);
end
```

```
function gettime
%==========================================================
global PROC_TIME TIME_TXT
%----------------------------------------------------------------------
format short;
hours=(PROC_TIME-mod(PROC_TIME,3600))/3600;
minutes=(PROC_TIME-(hours*3600)-mod(PROC_TIME-(hours*3600),60))/60;
seconds=PROC_TIME-(hours*3600)-(minutes*60);
time=[num2str(hours),':',num2str(minutes),':',num2str(seconds)];
set(TIME_TXT,'String',time);
return

function NUMODEL=getnumod
%==========================================================
filexist=fopen('c:\matlab\toolbox\erg\auto\numodel.txt','r');
if filexist == -1
  NUMODEL=0;
  save 'c:\matlab\toolbox\erg\auto\numodel.txt' NUMODEL -mat;
else
  fclose(filexist);
  load 'c:\matlab\toolbox\erg\auto\numodel.txt' NUMODEL -mat;
end
```

B.3 manual.m

```
function manual(action)
% Main program for manual extraction of the oscillatory potential (op). The program
% provides an interface for launching functions that a) opens and reads the Erg
% data file (Rfile.m); b) windows and extracts the op (Winerg.m); c) performs parameter
% estimation for the extracted op (Modelop.m); and d) saves the extracted op and op
% model
% The program is called by Erg.m.
% When the process is completed, the user can press the "Save". The program saves
% two types of files in the currently selected erg directory:
% 1) An ascii file named "Single.dat" that contains:
%      A) date/time of evaluation.
%      B) op model used for evaluation.
%      C) erg header information.
%      D) filter's corner frequencies.
%      E) sampling frequency of the erg data.
%      F) optimized op model's parameter vector.
%      G) error between extracted op and the op model.
%      H) %ERFIT = The relation of the op model fitting error
%              and the size of the b-wave.
%      I) %RELFIT = The relation of the op model envelope size
```

```
%              and the size of the b-wave.
% 2) An ascii files named "{erg file name}.op" that contains:
%      A) date/time of evaluation.
%      B) [TIME OP OP MODEL] matrix.
%
% NOTE - Every time the save option is executed the previous versions of the .dat
% and .op files will be overwritten. Be sure to move them or rename them
% before execution. Both files can be imported into a spreadsheet.
%
% Peter Derr, Dr Andrew Meyer and Dr Edward Haupt 11/18/97
% Rev. A 12/3/97
% Rev. B 8/20/98
% Rev. C 9/23/98
% Required M-Files: rfile.m, winerg.m and modelop.m
%=====================================================================

% Global variables generated in Manual.m:
%    MANUAL_WIN:   Main figure object.
%    DONE_MAN_WIN: Done pushbutton object.
%    SAVE_WIN:     Save Results pushbutton object.
%    MODEL_WIN:    Parameter Estimation pushbutton object.
%    CHOOSE_WIN:   Choose Erg File pushbutton object.
%    ERG_NAME:     Name of Erg file, without extension.
%    ERG_DATA:     Two column erg data matrix [time,signal].
%=====================================================================
%erg.m generated global variables:
global METHOD
%------------------------------------------------------------------------
%manual.m generated global variables:
global DONE_MAN_WIN SAVE_WIN MODEL_WIN CHOOSE_WIN ERG_NAME
FILE
global ERG_DATA CONTROL_PANEL WINDOW_MESSAGE WINDOW_ERROR
PATHNAME HEADER
%------------------------------------------------------------------------
%fergferg.m generated global variables:
global OP APP
%------------------------------------------------------------------------
%winerg.m generated global variables:
global ERG_TIME ERG_SIGNAL LENGTH_SIGNAL SAMPLE_FREQ FH FL
%------------------------------------------------------------------------
% modelop.m generated global variables
global E_OPT P_OPT M_OUT MAXPARAM TEMP_TITLE TEMP_ERR
%------------------------------------------------------------------------
if strcmp(action,'initialize')
  MANUAL_WIN=figure( ...
    'Name','Manual OP Extraction Mode', ...
```

```
  'NumberTitle', 'off', ...
  'Units', 'normalized', ...
  'MenuBar', 'none', ...
  'BackingStore','on', ...
  'Position',[.01 .065 .98 .88]);
%----------------------------------------------------------------------
CONTROL_PANEL=uicontrol(...
  'Style','frame', ...
  'Units','normalized', ...
  'Position',[.79 .02 .20 .95], ...
  'BackgroundColor', [0.50 0.50 0.50]);
%----------------------------------------------------------------------
CHOOSE_WIN=uicontrol( ...
  'String', 'Choose Erg File', ...
  'Style', 'pushbutton', ...
  'Units', 'normalized', ...
  'Position', [.81 0.275 0.16 .075], ...
  'Enable', 'on', ...
  'CallBack', 'manual("geterg")');
%----------------------------------------------------------------------
MODEL_WIN=uicontrol( ...
  'String', 'Parameter Opt', ...
  'Style', 'pushbutton', ...
  'Units', 'normalized', ...
  'Position', [.81 0.195 0.16 .075], ...
  'Enable', 'off', ...
  'CallBack', 'manual("model")');
%----------------------------------------------------------------------
SAVE_WIN=uicontrol( ...
  'String', 'Save Results', ...
  'Style', 'pushbutton', ...
  'Units', 'normalized', ...
  'Position', [.81 .115 .16 .075], ...
  'Enable', 'off', ...
  'CallBack','manual("savedata")');
%----------------------------------------------------------------------
DONE_MAN_WIN=uicontrol( ...
  'String', 'Done', ...
  'Style', 'pushbutton', ...
  'Units', 'normalized', ...
  'Position', [.81 .03 .16 .075], ...
  'Enable', 'on', ...
  'CallBack','manual("housekeeping")');
%----------------------------------------------------------------------
WINDOW_MESSAGE=uicontrol( ...
  'Style','text', ...
```

```
    'Units','normalized', ...
    'Position', [.01 .02 .77 .95], ...
    'horizontalAlignment','left', ...
    'BackgroundColor', [1 1 1], ...
    'Enable','on');
  %----------------------------------------------------------------------
  WINDOW_ERROR=[ ...
      'Program could open manual.txt, If the path is incorrect, ' ...
      'modify line 119 of manual.m to point to the correct path'];
  window_fid=fopen('c:\matlab\toolbox\Erg\auto\manual.txt','rt');
  if window_fid == -1
    set(WINDOW_MESSAGE,'String',WINDOW_ERROR);
  else
    WINDOW_TEXT=fscanf(window_fid,'%c',inf);
    fclose(window_fid);
    set(WINDOW_MESSAGE,'String',WINDOW_TEXT);
  end
  %----------------------------------------------------------------------
elseif strcmp(action,'geterg')
  set(SAVE_WIN,'Enable','off');
  [ERG_NAME,PATHNAME]=uigetfile('C:\Matlab\Toolbox\Erg\*.*','ERG DATA
                                              FILES');
  FILE=[PATHNAME,ERG_NAME];
  fileid=fopen(FILE,'rt');
  [HEADER,ERG_DATA]=rfile(fileid);
  fclose(fileid);
  clear fileid;
  clear global OP APP ERG_TIME ERG_SIGNAL LENGTH_SIGNAL TIME_ZERO
  SAMPLE_FREQ FH FL E_OPT P_OPT M_OUT;
  winerg('initialize');
  set(MODEL_WIN,'Enable','on');
elseif strcmp(action,'model')
  modelop('initialize');
  set(SAVE_WIN,'Enable','on');
elseif strcmp(action,'savedata')
  now_date=now;
  convert_date=datestr(now_date,0);
  erfit=(E_OPT/P_OPT(TEMP_ERR))*100;
  relamp=(P_OPT(TEMP_ERR)/APP)*100;
  store_file=[PATHNAME,'Single.dat'];
  paramid=fopen(store_file,'Wt');
  fprintf(paramid,'%30s \n\n',convert_date);
  fprintf(paramid,'%35s %7s %7s %9s %9s','FILE TITLE', ...
    'F(high)','F(low)','F(samp)','Atp');
  for index=1:MAXPARAM
    fprintf(paramid,'%11s',char(TEMP_TITLE(index)));
```

```
  end
  fprintf(paramid,'%10s %9s %9s \n','Error','%ERFIT','%RELAMP');
  clear index
  format_popt=['%35s %7.2f %7.2f %9.2f %9.2f'];
  for index=1:MAXPARAM
    format_popt=[format_popt,' ','%10.2f'];
  end
  format_popt=[format_popt,' ','%9.2f %9.2f %9.2f\n'];
  fprintf(paramid,format_popt,HEADER,FH,FL,SAMPLE_FREQ,
         APP,P_OPT,E_OPT,erfit,relamp);
  fclose(paramid);
  length_ergname=length(FILE);
  for index=1:length_ergname
    if strcmp(FILE(index),'.')
      dot_locate=index;
      break;
    end
  end
  clear index
  FILE(dot_locate)=[];
  new_op_name=[FILE,'.op'];
  fopid=fopen(new_op_name,'Wt');
  fprintf(fopid,'%8s %8s %8s \n','Time(ms)','OP(uV)','MODEL(uV)');
  fclose(fopid);
  M_OUT=M_OUT(:);
  opsize=max(size(OP));
  opmod=[OP(1,1) OP(1,2) M_OUT(1)];
  fopid=fopen(new_op_name,'At');
  fprintf(fopid,'%8.2f %8.2f %8.2f \n',opmod);
  fclose(fopid);
  for index=2:opsize
    opmod=[OP(index,1) OP(index,2) M_OUT(index)];
    fopid=fopen(new_op_name,'At');
    fprintf(fopid,'%8.2f %8.2f %8.2f \n',opmod);
    fclose(fopid);
  end
elseif strcmp(action,'housekeeping')
  clear DONE_MAN_WIN SAVE_WIN MODEL_WIN CHOOSE_WIN ...
       ERG_NAME ERG_DATA CONTROL_PANEL WINDOW_MESSAGE ...
       WINDOW_ERROR PATHNAME HEADER MAXPARAM TEMP_TITLE...
       TIME_ERR
  close(gcf);
end
```

B.4 rfile.m

```
% rfile.m
% Function reads in the Erg data file and partitions the header and
% Erg information. The header information is put into vector head. The
% Erg information is placed into the two column matrix process. The
% first column is the time vector while the second column is the Erg
% signal. Either Auto.m or Manual.m calls the program. The subroutine
% does pass variables.
%
% Variables:
%  Input:
%       filen: The id number assigned to the Erg Data file when opened.
%
%  Output:
%       head:   Erg's header information.
%       process: Erg signal matrix. First column is time vector. Second
%       column is the Erg signal.
%
% Peter Derr, Dr Andrew Meyer and Dr Edward Haupt 11/18/97
%--------------------------------------------------------------------------
function [head,process]=rfile(filen)
fseek(filen,0,'bof');
head=fgetl(filen);
locate=max(size(head));
fseek(filen,locate+1,'bof');
process=[];
while ~feof(filen)
  A=fscanf(filen,'%schar',1);
  maxA=max(size(A));
  if strcmp(A(maxA),',')
    A(maxA)=[];
  end
  A=str2num(A);
  B=fscanf(filen,'%schar',1);
  B=str2num(B);
  process=[process;A B];
end
```

B.5 modelop.m

```
function modelop(action)
% Optimize multi parameter model to fit extracted op. The program also
% displays the parameter fitting against the extracted op.
%
% Peter Derr, Dr Andrew Meyer and Dr. Edward Haupt 5/7/98 Rev 0

% Global variables generated in modelop.m
%    OP_AX:       OP plot axes object.
%    ERG_AX:      ERG plot axes object.
%    MODELDES_WIN: Popup menu for choosing desired model.
%
%    P1_WIN, P2_WIN, P3_WIN, P4_WIN, P5_WIN, P6_WIN, P7_WIN, P8_WIN,
%    P9_WIN, P10_WIN, P11_WIN, P12_WIN: Op model parameter title objects.
%
%    P1_INIT, P2_INIT, P3_INIT, P4_INIT, P5_INIT, P6_INIT, P7_INIT, P8_INIT,
%    P9_INIT, P10_INIT, P11_INIT, P12_INIT: Op model initial parameter objects.
%
%    P1_OPT P2_OPT P3_OPT P4_OPT P5_OPT P6_OPT P7_OPT P8_OPT P9_OPT
%    P10_OPT P11_OPT P12_OPT: Op model optimized parameter objects.
%
%    ERFIT_INIT:  Initial relative error object.
%    RELAMP_INIT: Initial relative amplitude object.
%    ERFIT_OPT:   Optimized relative error object.
%    RELAMP_OPT:  Optimized relative amplitude object.
%    EXPAND_OP:   Expand OP plot object.
%    TITLE:       Op Model Description variable string.
%    INIVAL:      Initial parameter variable string.
%    OPTVAL:      Optimized parameter variable string.
%    FLAG:        Flag for determining plotting op model.(Initial or Optimized).
%    PLOTOP:      OP plotting information vector.
%    P_INIT:      Initial parameter vector.
%    MAXPARAM:    Maximum number of parameters in selected op model.
%    RECORD:      OP model number variable.
%    MODEL:       OP model.
%    TEMP_VAR:    Temporary op model variables.
%    TEMP_MODEL:  Temporary op model.
%    TEMP_ERR:    Parameter number used for calculating %ERRFIT
%    E_OPT:       Optimized op model error.
%    P_OPT:       Optimized parameter vector.
%    M_OUT:       OP model matrix.
%    LAST_OP:     OP model matrix, used for plotting the optimized op model.
%------------------------------------------------------------------------
% manual.m generated global variables:
global ERG_NAME
```

```
%---------------------------------------------------------------------
% fergferg.m generated global variables:
global OP APP
%---------------------------------------------------------------------
% winerg.m generated global variables:
global ERG_TIME ERG_SIGNAL
%---------------------------------------------------------------------
% modelop.m generated global variables
global OP_AX ERG_AX MODELDES_WIN P1_WIN P2_WIN P3_WIN P4_WIN
global P5_WIN P6_WIN P7_WIN P8_WIN P9_WIN P10_WIN P11_WIN P12_WIN
global P1_INIT P2_INIT P3_INIT P4_INIT P5_INIT P6_INIT P7_INIT P8_INIT
global P9_INIT P10_INIT P11_INIT P12_INIT P1_OPT P2_OPT P3_OPT P4_OPT
global P5_OPT P6_OPT P7_OPT P8_OPT P9_OPT P10_OPT P11_OPT P12_OPT
global ERFIT_INIT RELAMP_INIT ERFIT_OPT RELAMP_OPT EXPAND_OP
global TITLE INIVAL OPTVAL FLAG PLOTOP P_INIT MAXPARAM
global RECORD MODEL TEMP_VAR TEMP_MODEL E_OPT P_OPT M_OUT
global LAST_OP TEMP_ERR TEMP_TITLE USEDEF_WIN TEMP_INI LAST_OPT
%---------------------------------------------------------------------
if nargin < 1
  action='initialize';
end
if strcmp(action,'initialize')
  MODEL_OP=figure( ...
    'Name','OP Model Parameter Estimitation', ...
    'Units', 'normalized', ...
    'Position',[.01 .065 .98 .88], ...
    'NumberTitle', 'off', ...
    'Pointer','arrow', ...
    'MenuBar', 'none', ...
    'BackingStore','on');
  %-------------------------------------------------------------------
  OP_AX=axes( ...
    'Units','normalized', ...
    'Color', [1 1 1], ...
    'XColor',[0 0 0], ...
    'YColor',[0 0 0], ...
    'Position',[0.1 0.40 0.65 0.20], ...
    'Drawmode','fast', ...
    'Visible', 'on');
  %-------------------------------------------------------------------
  ERG_AX=axes( ...
    'Units','normalized', ...
    'Color', [1 1 1], ...
    'XColor',[0 0 0], ...
    'YColor',[0 0 0], ...
    'Position',[0.1 0.75 0.65 0.20], ...
```

```
 'Drawmode','fast', ...
 'Visible','on');
%-----------------------------------------------------------------
CONSOLE_WIN=uicontrol( ...
 'Style','frame', ...
 'Units','normalized', ...
 'Position',[.79 .02 .20 .95], ...
 'BackGroundColor',[0.60 0.60 0.60]);
%-----------------------------------------------------------------
USEDEF_WIN=uicontrol( ...
 'String', 'Default Params', ...
 'Style', 'pushbutton', ...
 'Units', 'normalized', ...
 'Position', [.81 .865 .16 .075], ...
 'Enable', 'on', ...
 'CallBack','modelop("usedefault")');
%-----------------------------------------------------------------
PICKMODEL_WIN=uicontrol( ...
 'String', 'Choose Model', ...
 'Style', 'pushbutton', ...
 'Units', 'normalized', ...
 'Position', [.81 .78 .16 .075], ...
 'Enable', 'on', ...
 'CallBack','modelop("modelmenu")');
%-----------------------------------------------------------------
MODELDES_WIN=uicontrol(...
 'Style','popup', ...
 'Units','normalized', ...
 'FontSize',6, ...
 'FontWeight','Bold', ...
 'Position',[.01 .705 .98 .05], ...
 'BackgroundColor', [1 1 1], ...
 'String','INITIALIZING', ...
 'Visible','on', ...
 'CallBack','modelop("getrecord");');
%-----------------------------------------------------------------
ADDMODEL_WIN=uicontrol( ...
 'String', 'Model Editor', ...
 'Style', 'pushbutton', ...
 'Units', 'normalized', ...
 'Position', [.81 .61 .16 .075], ...
 'Enable', 'on', ...
 'CallBack','modeleditor("initialize")');
%-----------------------------------------------------------------
EXPAND_OP=uicontrol( ...
 'String','Align OP w/ ERG', ...
```

```
 'Style','checkbox', ...
 'BackGroundColor',[1 1 1]', ...
 'Units','normalized', ...
 'Position', [.81 .55 .16 .04], ...
 'Value', 0, ...
 'Enable','on', ...
 'Callback','modelop("expand_op")');
%-----------------------------------------------------------------
COMPARE_WIN=uicontrol( ...
 'String', 'Initial Conditions', ...
 'Style', 'pushbutton', ...
 'Units', 'normalized', ...
 'Position', [.81 .455 .16 .075], ...
 'Enable', 'on', ...
 'CallBack','modelop("useinit")');
%-----------------------------------------------------------------
OPTIMIZE_WIN=uicontrol( ...
 'String', 'Optimize', ...
 'Style', 'pushbutton', ...`
 'Units', 'normalized', ...
 'Position', [.81 .37 .16 .075], ...
 'Enable', 'on', ...
 'CallBack','modelop("optimize")');
%-----------------------------------------------------------------
START_WIN=uicontrol( ...
 'String', 'Start Over', ...
 'Style', 'pushbutton', ...
 'Units', 'normalized', ...
 'Position', [.81 .285 .16 .075], ...
 'Enable', 'on', ...
 'CallBack','modelop("start")');
%-----------------------------------------------------------------
PRINT_WIN=uicontrol( ...
 'String', 'Print', ...
 'Style', 'pushbutton', ...
 'Units', 'normalized', ...
 'Position', [.81 .2 .16 .075], ...
 'Enable', 'on', ...
 'CallBack','print -dwin');
%-----------------------------------------------------------------
HELP_WIN=uicontrol( ...
 'String', 'Help', ...
 'Style', 'pushbutton', ...
 'Units', 'normalized', ...
 'Position', [.81 .115 .16 .075], ...
 'Enable', 'on', ...
```

```
 'CallBack','modelop("help")');
%-----------------------------------------------------------------
DONE_WIN=uicontrol( ...
 'String', 'Done', ...
 'Style', 'pushbutton', ...
 'Units', 'normalized', ...
 'Position', [.81 .03 .16 .075], ...
 'Enable', 'on', ...
 'CallBack', 'modelop("housekeeping")');
%-----------------------------------------------------------------
TITLE_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.11 .276 .6419 .031], ...
 'String', 'OP MODEL PARAMETERS', ...
 'BackgroundColor', 'green');
%-----------------------------------------------------------------
INITIAL1_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.01 .201 .1 .044], ...
 'String', 'INITIAL: ', ...
 'BackgroundColor', 'red');
%-----------------------------------------------------------------
OPTIMIZED1_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.01 .170 .1 .031], ...
 'String', 'OPTIMIZED: ', ...
 'BackgroundColor', 'red');
%-----------------------------------------------------------------
INITIAL2_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.01 .051 .1 .044], ...
 'String', 'INITIAL: ', ...
 'BackgroundColor', 'red');
%-----------------------------------------------------------------
OPTIMIZED2_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.01 .02 .1 .031], ...
 'String', 'OPTIMIZED: ', ...
 'BackgroundColor', 'red');
%-----------------------------------------------------------------
P1_WIN=uicontrol(...
```

```
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.11 .245 .0917 .031], ...
 'String','Not Used', ...
 'Enable','off', ...
 'BackgroundColor', 'yellow');
%-----------------------------------------------------------------
P2_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.2017 .245 .0917 .031], ...
 'String','Not Used', ...
 'Enable','off', ...
 'BackgroundColor', 'yellow');
%-----------------------------------------------------------------
P3_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.2934 .245 .0917 .031], ...
 'String','Not Used', ...
 'Enable','off', ...
 'BackgroundColor', 'yellow');
%-----------------------------------------------------------------
P4_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.3851 .245 .0917 .031], ...
 'String','Not Used', ...
 'Enable','off', ...
 'BackgroundColor', 'yellow');
%-----------------------------------------------------------------
P5_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.4768 .245 .0917 .031], ...
 'String','Not Used', ...
 'Enable','off', ...
 'BackgroundColor', 'yellow');
%-----------------------------------------------------------------
P6_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.5685 .245 .0917 .031], ...
 'String','Not Used', ...
 'Enable','off', ...
 'BackgroundColor', 'yellow');
```

```
%----------------------------------------------------------------
P7_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.6602 .245 .0917 .031], ...
 'String','Not Used', ...
 'Enable','off', ...
 'BackgroundColor', 'yellow');
%----------------------------------------------------------------
P8_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.11 .095 .0917 .031], ...
 'String','Not Used', ...
 'Enable','off', ...
 'BackgroundColor', 'yellow');
%----------------------------------------------------------------
P9_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.2017 .095 .0917 .031], ...
 'String','Not Used', ...
 'Enable','off', ...
 'BackgroundColor', 'yellow');
%----------------------------------------------------------------
P10_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.2934 .095 .0917 .031], ...
 'String','Not Used', ...
 'Enable','off', ...
 'BackgroundColor', 'yellow');
%----------------------------------------------------------------
P11_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.3851 .095 .0917 .031], ...
 'String','Not Used', ...
 'Enable','off', ...
 'BackgroundColor', 'yellow');
%----------------------------------------------------------------
P12_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.4768 .095 .0917 .031], ...
 'String','Not Used', ...
```

```
 'Enable','off', ...
 'BackgroundColor', 'yellow');
%-----------------------------------------------------------------
ERFIT_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.5685 .095 .0917 .031], ...
 'String', '%ErFit', ...
 'BackgroundColor', 'red');
%-----------------------------------------------------------------
RELAMP_WIN=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.6602 .095 .0917 .031], ...
 'String', '%RelAmp', ...
 'BackgroundColor', 'red');
%-----------------------------------------------------------------
P1_INIT=uicontrol(...
 'Style','edit', ...
 'Units','normalized', ...
 'Position',[.11 .201 .0917 .044], ...
 'BackgroundColor', 'white');
%-----------------------------------------------------------------
P2_INIT=uicontrol(...
 'Style','edit', ...
 'Units','normalized', ...
 'Position',[.2017 .201 .0917 .044], ...
 'BackgroundColor','white');
%-----------------------------------------------------------------
P3_INIT=uicontrol(...
 'Style','edit', ...
 'Units','normalized', ...
 'Position',[.2934 .201 .0917 .044], ...
 'BackgroundColor', 'white');
%-----------------------------------------------------------------
P4_INIT=uicontrol(...
 'Style','edit', ...
 'Units','normalized', ...
 'Position',[.3851 .201 .0917 .044], ...
 'BackgroundColor', 'white');
%-----------------------------------------------------------------
P5_INIT=uicontrol(...
 'Style','edit', ...
 'Units','normalized', ...
 'Position',[.4768 .201 .0917 .044], ...
 'BackgroundColor', 'white');
```

```
%-----------------------------------------------------------------
P6_INIT=uicontrol(...
 'Style','edit', ...
 'Units','normalized', ...
 'Position',[.5685 .201 .0917 .044], ...
 'BackgroundColor', 'white');
%-----------------------------------------------------------------
P7_INIT=uicontrol(...
 'Style','edit', ...
 'Units','normalized', ...
 'Position',[.6602 .201 .0917 .044], ...
 'BackgroundColor', 'white');
%-----------------------------------------------------------------
P8_INIT=uicontrol(...
 'Style','edit', ...
 'Units','normalized', ...
 'Position',[.11 .051 .0917 .044], ...
 'BackgroundColor', 'white');
%-----------------------------------------------------------------
P9_INIT=uicontrol(...
 'Style','edit', ...
 'Units','normalized', ...
 'Position',[.2017 .051 .0917 .044], ...
 'BackgroundColor','white');
%-----------------------------------------------------------------
P10_INIT=uicontrol(...
 'Style','edit', ...
 'Units','normalized', ...
 'Position',[.2934 .051 .0917 .044], ...
 'BackgroundColor', 'white');
%-----------------------------------------------------------------
P11_INIT=uicontrol(...
 'Style','edit', ...
 'Units','normalized', ...
 'Position',[.3851 .051 .0917 .044], ...
 'BackgroundColor', 'white');
%-----------------------------------------------------------------
P12_INIT=uicontrol(...
 'Style','edit', ...
 'Units','normalized', ...
 'Position',[.4768 .051 .0917 .044], ...
 'BackgroundColor', 'white');
%-----------------------------------------------------------------
ERFIT_INIT=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
```

```
 'Position',[.5685 .051 .0917 .044], ...
 'BackgroundColor', 'white');
%-------------------------------------------------------------------
RELAMP_INIT=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.6602 .051 .0917 .044], ...
 'BackgroundColor', 'white');
%-------------------------------------------------------------------
P1_OPT=uicontrol(...
 'Style','text', ...
 'horizontalAlignment','center', ...
 'Units','normalized', ...
 'Position',[.11 .170 .0917 .031], ...
 'BackgroundColor', 'white');
%-------------------------------------------------------------------
P2_OPT=uicontrol(...
 'Style','text', ...
 'horizontalAlignment','center', ...
 'Units','normalized', ...
 'Position',[.2017 .170 .0917 .031], ...
 'BackgroundColor','white');
%-------------------------------------------------------------------
P3_OPT=uicontrol(...
 'Style','text', ...
 'horizontalAlignment','center', ...
 'Units','normalized', ...
 'Position',[.2934 .170 .0917 .031], ...
 'BackgroundColor', 'white');
%-------------------------------------------------------------------
P4_OPT=uicontrol(...
 'Style','text', ...
 'horizontalAlignment','center', ...
 'Units','normalized', ...
 'Position',[.3851 .170 .0917 .031], ...
 'BackgroundColor', 'white');
%-------------------------------------------------------------------
P5_OPT=uicontrol(...
 'Style','text', ...
 'horizontalAlignment','center', ...
 'Units','normalized', ...
 'Position',[.4768 .170 .0917 .031], ...
 'BackgroundColor', 'white');
%-------------------------------------------------------------------
P6_OPT=uicontrol(...
 'Style','text', ...
```

```
 'horizontalAlignment','center', ...
 'Units','normalized', ...
 'Position',[.5685 .170 .0917 .031], ...
 'BackgroundColor', 'white');
%-------------------------------------------------------------------
P7_OPT=uicontrol(...
 'Style','text', ...
 'horizontalAlignment','center', ...
 'Units','normalized', ...
 'Position',[.6602 .170 .0917 .031], ...
 'BackgroundColor', 'white');
%-------------------------------------------------------------------
P8_OPT=uicontrol(...
 'Style','text', ...
 'horizontalAlignment','center', ...
 'Units','normalized', ...
 'Position',[.11 .02 .0917 .031], ...
 'BackgroundColor', 'white');
%-------------------------------------------------------------------
P9_OPT=uicontrol(...
 'Style','text', ...
 'horizontalAlignment','center', ...
 'Units','normalized', ...
 'Position',[.2017 .02 .0917 .031], ...
 'BackgroundColor','white');
%-------------------------------------------------------------------
P10_OPT=uicontrol(...
 'Style','text', ...
 'horizontalAlignment','center', ...
 'Units','normalized', ...
 'Position',[.2934 .02 .0917 .031], ...
 'BackgroundColor', 'white');
%-------------------------------------------------------------------
P11_OPT=uicontrol(...
 'Style','text', ...
 'horizontalAlignment','center', ...
 'Units','normalized', ...
 'Position',[.3851 .02 .0917 .031], ...
 'BackgroundColor', 'white');
%-------------------------------------------------------------------
P12_OPT=uicontrol(...
 'Style','text', ...
 'horizontalAlignment','center', ...
 'Units','normalized', ...
 'Position',[.4768 .02 .0917 .031], ...
 'BackgroundColor', 'white');
```

```
%---------------------------------------------------------------------
ERFIT_OPT=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.5685 .02 .0917 .031], ...
 'BackgroundColor', 'white');
%---------------------------------------------------------------------
RELAMP_OPT=uicontrol(...
 'Style','text', ...
 'Units','normalized', ...
 'Position',[.6602 .02 .0917 .031], ...
 'BackgroundColor', 'white');
%---------------------------------------------------------------------
modelop('start')
elseif strcmp(action,'start')
 for index=1:12
   TITLE(index)=eval(['P',num2str(index),'_WIN']);
   INIVAL(index)=eval(['P',num2str(index),'_INIT']);
   OPTVAL(index)=eval(['P',num2str(index),'_OPT']);
   set(OPTVAL(index),'String','');
 end
 clear index
 axes(ERG_AX);
 plot(ERG_TIME,ERG_SIGNAL,'r');
 setergax;
 axes(OP_AX);
 PLOTOP=plot(OP(:,1),OP(:,2),'r','Erasemode','none');
 setopax;
 modelop('modelmenu')
elseif strcmp(action,'modelmenu')
 menu_str='';
 model_ct=getnumod;
 if model_ct > 0
   load 'c:\matlab\toolbox\erg\auto\model.txt' MODEL -mat;
   for index=1:model_ct-1
     menu_str=[menu_str,MODEL(index).des,'|'];
   end
   menu_str=[menu_str,MODEL(model_ct).des];
 else
   menu_str='CREATE MODELS';
 end
 set(MODELDES_WIN,'Visible','on','String',menu_str);
 clear index model_ct menu_str getnumodel
elseif strcmp(action,'getrecord')
 RECORD=get(MODELDES_WIN,'Value');
 set(MODELDES_WIN,'Visible','off');
```

```
  TEMP_MODEL=char(eval(MODEL(RECORD).equ));
  TEMP_VAR=eval(MODEL(RECORD).var);
  TEMP_ERR=eval(char(MODEL(RECORD).err));
  TEMP_INI=eval(char(MODEL(RECORD).ini));
  TEMP_LAST=eval(char(MODEL(RECORD).last));
  TEMP_TITLE=eval(MODEL(RECORD).title);
  MAXPARAM=max(size(TEMP_VAR));
  for index=1:MAXPARAM
    set(TITLE(index),'Enable','on','String',char(TEMP_TITLE(index)));
    set(INIVAL(index),'Enable','on','String',num2str(TEMP_LAST(index)));
    set(OPTVAL(index),'Enable','on');
    tmp_param=['pm(',num2str(index),')'];
    TEMP_MODEL=subs(TEMP_MODEL,TEMP_VAR{index},tmp_param);
  end
  clear index;
  for index=(MAXPARAM+1):12
    set(TITLE(index),'Enable','off','String','Not Used');
    set(INIVAL(index),'Enable','off','String','');
    set(OPTVAL(index),'Enable','off','String','');
  end
elseif strcmp(action,'expand_op')
  if get(EXPAND_OP,'Value') == 0
    if isempty(M_OUT)
      xlim=[min(OP(:,1)) max(OP(:,1))];
      ylim=[min(OP(:,2)) max(OP(:,2))];
      set(OP_AX,'XLim',xlim,'YLim',ylim);
      setopax;
    else
      xlim=[min(OP(:,1)) max(OP(:,1))];
      ylim=[min(min(OP(:,2),M_OUT)) max(max(OP(:,2),M_OUT))];
      set(OP_AX,'XLim',xlim,'YLim',ylim);
      setopax;
    end
  elseif get(EXPAND_OP,'Value') == 1
    xlim=[min(ERG_TIME) max(ERG_TIME)];
    ylim=[min(ERG_SIGNAL) max(ERG_SIGNAL)];
    if isempty(M_OUT)
      set(OP_AX,'XLim',xlim,'YLim',ylim);
      setopax;
    else
      set(OP_AX,'XLim',xlim,'YLim',ylim);
      setopax;
    end
  end
elseif strcmp(action,'usedefault')
  load 'c:\matlab\toolbox\erg\auto\model.txt' MODEL -mat;
```

```
  TEMP_INI=eval(char(MODEL(RECORD).ini));
  for index=1:MAXPARAM
    set(INIVAL(index),'Enable','on','String',num2str(TEMP_INI(index)));
  end
  for index=1:12
    TITLE(index)=eval(['P',num2str(index),'_WIN']);
    INIVAL(index)=eval(['P',num2str(index),'_INIT']);
    OPTVAL(index)=eval(['P',num2str(index),'_OPT']);
    set(OPTVAL(index),'String','');
  end
elseif strcmp(action,'useinit')
  clear P_INIT
  for index=1:MAXPARAM
    P_INIT(index)=str2num(get(INIVAL(index),'String'));
  end
  clear index
  FLAG=0;
  paramcmp(P_INIT);
  PLOTOP=plot(OP(:,1),OP(:,2),'r',OP(:,1),M_OUT,'k','Erasemode','none');
  setopax;
elseif strcmp(action,'optimize')
  clear P_INIT P_OPT
  for index=1:MAXPARAM
    P_INIT(index)=str2num(get(INIVAL(index),'String'));
  end
  clear index
  for index=1:12
    set(OPTVAL(index),'String','');
  end
  FLAG=1;
  P_OPT=optparam(P_INIT);
  PLOTOP=plot(OP(:,1),OP(:,2),'r',OP(:,1),M_OUT,'k','Erasemode','none');
  setopax;
  for index=1:MAXPARAM
    set(OPTVAL(index),'String',num2str(P_OPT(index)));
  end
  set(ERFIT_OPT,'String',num2str((E_OPT/P_OPT(TEMP_ERR))*100));
  set(RELAMP_OPT,'String',num2str((P_OPT(TEMP_ERR)/APP)*100));
  MODEL(RECORD).last=mat2str(P_OPT,4);
  save 'c:\matlab\toolbox\erg\auto\model.txt' MODEL -mat;
elseif strcmp(action,'help')
  WINDOW_HELP=dialog( ...
    'Name','MODEL OP HELP WINDOW', ...
    'Units', 'normalized', ...
    'Position',[.01 .065 .77 .88], ...
    'NumberTitle','off', ...
```

```
    'Pointer','arrow', ...
    'MenuBar','none', ...
    'BackingStore', 'on');
  %-------------------------------------------------------------------
  CLOSE_HELP=uicontrol( ...
    'String','Close', ...
    'Style','pushbutton', ...
    'Units','normalized', ...
    'Position',[.83 .01 .16 .075], ...
    'Enable','on', ...
    'Callback', 'close(gcf)');
  %-------------------------------------------------------------------
  HELP_MESSAGE=uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position', [.008 .095 .984 .895], ...
    'horizontalAlignment','left', ...
    'BackgroundColor', [1 1 1], ...
    'Enable','on');
  %-------------------------------------------------------------------
  HELP_ERROR=[ ...
      'Program could not open modelop.txt, If the path is incorrect, ' ...
      'modify line 643 of modelop.m to point to the correct path.    '];
  help_fid=fopen('c:\matlab\toolbox\Erg\auto\modelop.txt','rt');
  if help_fid == -1
    set(HELP_MESSAGE,'String',HELP_ERROR);
  else
    help_message=fscanf(help_fid,'%c',inf);
    fclose(help_fid);
    set(HELP_MESSAGE,'String',help_message);
  end
elseif strcmp(action,'housekeeping')
  clear OP_AX ERG_AX MODELDES_WIN P1_WIN P2_WIN P3_WIN ...
  P4_WN P5_WIN P6_WIN P7_WIN P8_WIN P9_WIN P10_WIN P11_WIN ...
  P12_WIN P1_INIT P2_INIT P3_INIT P4_INIT P5_INIT P6_INIT P7_INIT ...
  P8_INIT P9_INIT P10_INIT P11_INIT P12_INIT P1_OPT P2_OPT P3_OPT ...
  P4_OPT P5_OT P6_OPT P7_OPT P8_OPT P9_OPT P10_OPT P11_OPT ...
  P12_OPT ERFIT_INIT RELAMP_INIT ERFIT_OPT RELAMP_OPT
  EXPAND_OP TITLE INIVAL OPTVAL FLAG PLOTOP P_INIT RECORD
  MODEL TEMP_VAR TEMP_MODEL LAST_OP;
  close(gcf);
end

function setergax
%===================================================================
global ERG_AX ERG_TIME ERG_SIGNAL ERG_NAME
```

```
xlim=[min(ERG_TIME) max(ERG_TIME)];
ylim=[min(ERG_SIGNAL) max(ERG_SIGNAL)];
set(ERG_AX,'FontSize',8,'Color',[1 1 1], ...
'XLim',xlim,'YLim',ylim,'XGrid','on','YGrid','on');
xlabel('Time (ms)','FontSize',8);
ylabel('Amplitude (uV)','FontSize',8);
title(['ORIGINAL (',ERG_NAME,') ERG'],'FontSize',8);
return;

function setopax
%============================================================
global OP_AX OP
set(OP_AX,'FontSize',8,'Color',[1 1 1],'XGrid','on','YGrid','on');
xlabel('Time (ms)','FontSize',8);
ylabel('Amplitude (uV)','FontSize',8);
title('EXTRACTED OP','FontSize',8);
return;

function NUMODEL=getnumod
%============================================================
filexist=fopen('c:\matlab\toolbox\erg\auto\numodel.txt','r');
if filexist == -1
  NUMODEL=0;
  save 'c:\matlab\toolbox\erg\auto\numodel.txt' NUMODEL -mat;
else
  fclose(filexist);
  load 'c:\matlab\toolbox\erg\auto\numodel.txt' NUMODEL -mat;
end
```

B.6 winerg.m

```
function winerg(action)
% The function provides a graphical user interface for windowing and filtering
% the erg signal for the extraction of the oscillatory potential (op). The function
% plots the erg frequency spectrum and the erg time domain signal and prompts the
% user to choose the window via the mouse on the erg time domain plot. After
% choosing the window, the windowed erg's frequency spectrum is displayed. The user
% can choose the bandwidth of the 2nd order bandpass butterworth filter by
% a) using the default bandwidth [65 hz 250 hz] or choose the bandwidth via the
% mouse on the frequency spectrum plot. The function is called by Manual.m and returns
% to the function Manual.m upon completion of op extraction.
%
% Peter Derr, Dr Andrew Meyer and Dr Edward Haupt 8/20/98
%============================================================
```

```
% Global variables generated in winerg.m:
%    MESSAGE_INFO:           Special windowing instructions.
%    WINDOW_ERG:             Main figure object.
%    MESSAGE_WIN:            Special instructions title object.
%    SPECIAL_MESSAGE:        Special instructions text object.
%    ERG_SPEC_AX:            Frequency spectrum axes object.
%    ERG_AX:                 Erg axes object.
%    OP_AX:                  OP axes object
%    CONSOLE_WIN:            Console window object.
%    DONE_WIN:               Done button object.
%    WINDOW_WIN:             Window erg button object.
%    RE_WINDOW_WIN:          Rewindow erg button object.
%    FILTER_CONSOLE_WIN:     Filter subconsole window object.
%    FILTER_WIN:             Extract OP button object.
%    DEFAULT_WIN:            Default bandpass button object.
%    FREQ_WIN:               Select bandpass frequencies button object.
%    HELP_WIN:               Help button object.
%    FREQ_HIGH_WIN:          High pass frequency text object.
%    FREQ_LOW_WIN:           Low pass frequency text object.
%    HOLD_OP:                Hold OP checkbox object.
%    ERG_TIME:               Erg time vector.
%    ERG_SIGNAL:             Erg signal vector.
%    LENGTH_SIGNAL:          Length of Erg matrix
%    TIME_ZERO:              Index of time=0 of the erg signal.
%    SAMPLE_FREQ:            Sampling frequency of the erg signal.
%    LEFT_WIN:               Index of left erg window.
%    RIGHT_WIN:              Index of right erg window.
%    FH:                     Highpass frequency.
%    FL:                     Lowpass frequency
%    B,A:                    Butterworth filter's coefficients.
%    OP_COLOR:               OP color vector, used with the hold op plot function.
%    OP_COLOR_CNT:           OP counter, used with the hold op plot function.
%    MAX_OP_AX:              Maximum OP value of the held OPs.
%    MIN_OP_AX:              Minimum OP value of the held OPs.
%    FREQ_VEC:               Frequency vector.
%    ERG_FREQ:               FFT of ERG signal
%
%==========================================================================
% erg.m generated global variables:
global METHOD
%-----------------------------------------------------------------------
% manual.m generated global variables:
global ERG_NAME ERG_DATA
%-----------------------------------------------------------------------
% fergferg.m generated global variables:
global OP APP
```

```
%---------------------------------------------------------------------
% winerg.m generated global variables:
global MESSAGE_INFO WINDOW_ERG MESSAGE_WIN SPECIAL_MESSAGE ...
global OP_AX ERG_AX ERG_SPEC_AX CONSOLE_WIN DONE_WIN ...
global WINDOW_WIN RE_WINDOW_WIN FILTER_CONSOLE_WIN ...
global FILTER_WIN DEFAULT_WIN FREQ_WIN HELP_WIN ...
global FREQ_HIGH_WIN FREQ_LOW_WIN HOLD_OP ERG_TIME ...
global ERG_SIGNAL LENGTH_SIGNAL TIME_ZERO SAMPLE_FREQ ...
global LEFT_WIN RIGHT_WIN FH FL B A OP_COLOR OP_COLOR_CNT ...
global MAX_OP_AX MIN_OP_AX FREQ_VEC ERG_FREQ
%---------------------------------------------------------------------
if nargin < 1
  action='initialize';
end
if strcmp(action,'initialize')
  MAX_OP_AX=0;
  MIN_OP_AX=0;
  OP_COLOR=['r','g','b','k','y','m','c'];
  OP_COLOR_CNT=0;
  MESSAGE_INFO=[ ...
  '                                                               ';
  'Position Vertical axis of crosshair over position of left ERG window     ';
  'Position Vertical axis of crosshair over position of right ERG window    ';
  'Position Vertical axis of crosshair over position of high pass frequency ';
  'Position Vertical axis of crosshair over position of low pass frequency  ';
  'Redo the ERG with a different window                                     '];
  %---------------------------------------------------------------------
  WINDOW_ERG=figure( ...
    'Name','Extract OP Via Windowing ERG', ...
    'Units', 'normalized', ...
    'Position',[.01 .065 .98 .88], ...
    'NumberTitle','off', ...
    'Pointer','arrow', ...
    'MenuBar','none', ...
    'BackingStore', 'on');
  %---------------------------------------------------------------------
  MESSAGE_WIN=uicontrol( ...
    'Style','text', ...
    'Units','normalized', ...
    'Position', [.01 .008 .17 .031], ...
    'String','Special Instructions: ', ...
    'horizontalAlignment','left', ...
    'BackgroundColor', [1 0 0], ...
    'Enable','on');
  %---------------------------------------------------------------------
  SPECIAL_MESSAGE=uicontrol( ...
```

```
  'Style','text', ...
  'Units','normalized', ...
  'Position', [.18 .008 .60 .031], ...
  'String', MESSAGE_INFO(1,:) , ...
  'horizontalAlignment','left', ...
  'BackgroundColor', [1 1 1], ...
  'Enable','on');
%--------------------------------------------------------------------
OP_AX=axes( ...
  'Units','normalized', ...
  'Color', [1 1 1], ...
  'XColor',[0 0 0], ...
  'YColor',[0 0 0], ...
  'Position',[0.1 0.11 0.65 0.20], ...
  'Drawmode','fast', ...
  'Visible', 'off');
%--------------------------------------------------------------------
ERG_AX=axes( ...
  'Units','normalized', ...
  'Color', [1 1 1], ...
  'XColor',[0 0 0], ...
  'YColor',[0 0 0], ...
  'Position',[0.1 0.43 0.65 0.20], ...
  'Drawmode','fast', ...
  'Visible','on');
%--------------------------------------------------------------------
ERG_SPEC_AX=axes( ...
  'Units','normalized', ...
  'Color', [1 1 1], ...
  'XColor',[0 0 0], ...
  'YColor',[0 0 0], ...
  'Position',[0.1 0.75 0.65 0.20], ...
  'Drawmode','fast', ...
  'Visible', 'on');
%--------------------------------------------------------------------
CONSOLE_WIN=uicontrol(...
  'Style','frame', ...
  'Units','normalized', ...
  'Position',[.79 .02 .20 .95], ...
  'BackgroundColor', [0.50 0.50 0.50]);
%--------------------------------------------------------------------
FILTER_CONSOLE_WIN=uicontrol( ...
  'Style','frame', ...
  'Units','normalized', ...
  'Position',[.80 .52 .18 .44], ...
  'BackgroundColor', [0.60 0.60 0.60]);
```

```
%----------------------------------------------------------------------
DEFAULT_WIN=uicontrol( ...
  'String','[65Hz,250Hz]', ...
  'Style','pushbutton', ...
  'Units','normalized', ...
  'Position', [.81 .875 .16 .075], ...
  'Enable','off', ...
  'Callback','winerg("default_filter")');
%----------------------------------------------------------------------
FREQ_WIN=uicontrol( ...
  'String','Choose [F(hp),F(lp)]', ...
  'Style','pushbutton', ...
  'Units','normalized', ...
  'Position', [.81 .79 .16 .075], ...
  'Enable','off', ...
  'Callback','winerg("freqspec")');
%----------------------------------------------------------------------
FREQ_HIGH_WIN=uicontrol( ...
  'Style','text', ...
  'Units','normalized', ...
  'Position', [.81 .74 .16 .04], ...
  'String',['F(hp): ',num2str(FH),' Hz'], ...
  'Enable','on');
%----------------------------------------------------------------------
FREQ_LOW_WIN=uicontrol( ...
  'Style','text', ...
  'Units','normalized', ...
  'Position', [.81 .70 .16 .04], ...
  'String',['F(lp): ',num2str(FL),' Hz'], ...
  'Enable','on');
%----------------------------------------------------------------------
FILTER_WIN=uicontrol( ...
  'String','Extract OP', ...
  'Style','pushbutton', ...
  'Units','normalized', ...
  'Position', [.81 .615 .16 .075], ...
  'Enable','off', ...
  'Callback','winerg("filt_erg")');
%----------------------------------------------------------------------
HOLD_OP=uicontrol( ...
  'String','Hold OP Plot', ...
  'Style','checkbox', ...
  'BackGroundColor',[1 1 1]', ...
  'Units','normalized', ...
  'Position', [.81 .55 .13 .04], ...
  'Value', 0, ...
```

```
    'Enable','off', ...
    'Callback','winerg("hold_op")');
  %-----------------------------------------------------------------------
  RE_WINDOW_WIN=uicontrol( ...
    'String','Redo ERG', ...
    'Style','pushbutton', ...
    'Units','normalized', ...
    'Position', [.81 .37 .16 .075], ...
    'Enable','off', ...
    'Callback','winerg("newerg")');
  %-----------------------------------------------------------------------
  WINDOW_WIN=uicontrol( ...
    'String','Window ERG', ...
    'Style','pushbutton', ...
    'Units','normalized', ...
    'Position', [.81 .285 .16 .075], ...
    'Enable','on', ...
    'Callback','winerg("window")');
  %-----------------------------------------------------------------------
  PRINT_WIN=uicontrol( ...
    'String', 'Print', ...
    'Style', 'pushbutton', ...
    'Units', 'normalized', ...
    'Position', [0.81 0.2 0.16 0.075], ...
    'Enable', 'on', ...
    'CallBack','print -dwin');
  %-----------------------------------------------------------------------
  HELP_WIN=uicontrol( ...
    'String','Help', ...
    'Style','pushbutton', ...
    'Units','normalized', ...
    'Position',[.81 .115 .16 .075], ...
    'Enable','on', ...
    'Callback','winerg("help")');
  %-----------------------------------------------------------------------
  DONE_WIN=uicontrol( ...
    'String','Done', ...
    'Style','pushbutton', ...
    'Units','normalized', ...
    'Position',[.81 .03 .16 .075], ...
    'Enable','on', ...
    'Callback', 'winerg("housekeeping")');
  %-----------------------------------------------------------------------
winerg('newerg');
elseif strcmp(action,'newerg')
  if get(HOLD_OP,'Value') == 0
```

```
    axes(OP_AX);
    cla;
    set(OP_AX,'Visible','off');
  end
  ERG_TIME=ERG_DATA(:,1);
  ERG_SIGNAL=ERG_DATA(:,2);
  LENGTH_SIGNAL=length(ERG_SIGNAL);
  for index=1:LENGTH_SIGNAL,
    if ERG_TIME(index) >= 0
      TIME_ZERO=index;
      break;
    end
  end
  if TIME_ZERO >= 3
    temp_signal=ERG_SIGNAL-mean(ERG_SIGNAL(2:TIME_ZERO-1));
    ERG_SIGNAL=temp_signal;
  end
  clear index temp_signal;
  sample_time=(ERG_TIME(TIME_ZERO+1)-ERG_TIME(TIME_ZERO))/1000;
  SAMPLE_FREQ=(sample_time)^(-1);
  axes(ERG_AX);
  plot(ERG_TIME,ERG_SIGNAL,'r');
  setergax;
  ERG_FREQ=fft(ERG_SIGNAL(TIME_ZERO:LENGTH_SIGNAL),512);
  FREQ_VEC=SAMPLE_FREQ*(1:256)/512;
  axes(ERG_SPEC_AX)
  semilogx(FREQ_VEC,20*log10(abs(ERG_FREQ(1:256))),'r');
  setfreqax;
  set(WINDOW_WIN,'Enable','on');
  set(RE_WINDOW_WIN,'Enable','off');
elseif strcmp(action,'hold_op');
  axes(OP_AX);
  hold;
  if ishold == 0
    COLOR_OP_CNT=1;
  end
elseif strcmp(action,'window')
  set(WINDOW_WIN,'Enable','off');
  set(SPECIAL_MESSAGE,'String',MESSAGE_INFO(2,:));
  [left_x, left_y]=ginput(1);
  left_x=round(left_x);
  set(SPECIAL_MESSAGE,'String',MESSAGE_INFO(3,:));
  [right_x, right_y]=ginput(1);
  right_x=round(right_x);
  set(SPECIAL_MESSAGE,'String',MESSAGE_INFO(1,:));
  for index=1:LENGTH_SIGNAL,
```

```
    if ERG_TIME(index) >= left_x
      LEFT_WIN=index;
      break;
    end
  end
  clear index;
  for index=LEFT_WIN:LENGTH_SIGNAL,
    if ERG_TIME(index) >= right_x
      RIGHT_WIN=index;
      break;
    end
  end
  if (LEFT_WIN >= TIME_ZERO) & (LEFT_WIN < RIGHT_WIN)
    set(SPECIAL_MESSAGE,'String',MESSAGE_INFO(1,:));
    set(ERG_AX,'NextPlot', 'add');
    set(ERG_AX,'FontSize',8,'Color',[1 1 1],'XColor',[0 0 0],'YColor',[0 0 0]);

plot(ERG_TIME(LEFT_WIN:RIGHT_WIN),ERG_SIGNAL(LEFT_WIN:RIGHT_WIN
),'k');
    set(ERG_AX,'NextPlot', 'remove');
    set(ERG_AX,'FontSize',8,'Color',[1 1 1],'XColor',[0 0 0],'YColor',[0 0 0]);
    ERG_FREQ=fft(ERG_SIGNAL(LEFT_WIN:RIGHT_WIN),512);
    axes(ERG_SPEC_AX);
    semilogx(FREQ_VEC,20*log10(abs(ERG_FREQ(1:256))),'r');
    setfreqax;
    set(RE_WINDOW_WIN,'Enable','on');
    set(DEFAULT_WIN,'Enable','on');
    set(FREQ_WIN,'Enable','on');
    set(HOLD_OP,'Enable','on');
  else
    set(SPECIAL_MESSAGE,'String',MESSAGE_INFO(6,:));
    set(WINDOW_WIN,'Enable','on');
  end
elseif strcmp(action,'default_filter')
  if get(HOLD_OP,'Value') == 0
    axes(OP_AX);
    cla;
    set(OP_AX,'Visible','off');
  end
  FH=65;
  FL=250;
  [B,A]=butter(2,[FH FL]*2/SAMPLE_FREQ);
  set(FREQ_HIGH_WIN,'String',['F(hp): ',num2str(FH),' Hz']);
  set(FREQ_LOW_WIN,'String',['F(lp): ',num2str(FL),' Hz']);
  set(FILTER_WIN,'Enable','on');
elseif strcmp(action,'freqspec')
```

```
 if get(HOLD_OP,'Value') == 0
   axes(OP_AX);
   cla;
   set(OP_AX,'Visible','off');
 end
 set(SPECIAL_MESSAGE,'String',MESSAGE_INFO(4,:));
 [FH,high_y]=ginput(1);
 FH=round(FH);
 set(SPECIAL_MESSAGE,'String',MESSAGE_INFO(5,:));
 [FL,low_y]=ginput(1);
 FL=round(FL);
 set(SPECIAL_MESSAGE,'String',MESSAGE_INFO(1,:));
 [B,A]=butter(2,[FH FL]*2/SAMPLE_FREQ);
 set(FREQ_HIGH_WIN,'String',['F(hp): ',num2str(FH),' Hz']);
 set(FREQ_LOW_WIN,'String',['F(lp): ',num2str(FL),' Hz']);
 set(FILTER_WIN,'Enable','on');
elseif strcmp(action,'filt_erg')
 fergferg;
 temp_max_op=max(OP(:,2));
 temp_min_op=min(OP(:,2));
 if temp_max_op >=MAX_OP_AX
   MAX_OP_AX=temp_max_op;
 end
 if temp_min_op <=MIN_OP_AX
   MIN_OP_AX=temp_min_op;
 end
 axes(OP_AX);
 if get(HOLD_OP,'Value') == 1
   if OP_COLOR_CNT == 7
     OP_COLOR_CNT=1;
   else
     OP_COLOR_CNT=OP_COLOR_CNT+1;
   end
 else
   OP_COLOR_CNT=1;
 end
 set(OP_AX,'Visible','on');
 plot(OP(:,1),OP(:,2),OP_COLOR(OP_COLOR_CNT));
 setopax;
elseif strcmp(action,'help')
 WINDOW_HELP=dialog( ...
   'Name','ERG WINDOWING HELP WINDOW', ...
   'Units', 'normalized', ...
   'Position',[.01 .065 .77 .88], ...
   'NumberTitle','off', ...
   'Pointer','arrow', ...
```

```
  'MenuBar','none', ...
  'BackingStore', 'on');
%-----------------------------------------------------------------------
CLOSE_HELP=uicontrol( ...
  'String','Close', ...
  'Style','pushbutton', ...
  'Units','normalized', ...
  'Position',[.83 .01 .16 .075], ...
  'Enable','on', ...
  'Callback', 'close(gcf)');
%-----------------------------------------------------------------------
HELP_MESSAGE=uicontrol( ...
  'Style','text', ...
  'Units','normalized', ...
  'Position', [.008 .095 .984 .895], ...
  'horizontalAlignment','left', ...
  'BackgroundColor', [1 1 1], ...
  'Enable','on');
%-----------------------------------------------------------------------
HELP_ERROR=[ ...
    'Program could open winerg.txt, If the path is incorrect, ' ...
    'modify line 421 of winerg.m to point to the correct path'];
help_fid=fopen('c:\matlab\toolbox\Erg\auto\winerg.txt','rt');
if help_fid == -1
  set(HELP_MESSAGE,'String',HELP_ERROR);
else
  help_message=fscanf(help_fid,'%c',inf);
  fclose(help_fid);
  set(HELP_MESSAGE,'String',help_message);
end
elseif strcmp(action,'housekeeping')
  clear MESSAGE_INFO WINDOW_ERG MESSAGE_WIN ...
  SPECIAL_MESSAGE OP_AX ERG_AX ERG_SPEC_AX ...
  CONSOLE_WIN DONE_WIN WINDOW_WIN RE_WINDOW_WIN ...
  FILTER_CONSOLE_WIN FILTER_WIN DEFAULT_WIN ...
  FREQ_WIN HELP_WIN FREQ_HIGH_WIN FREQ_LOW_WIN ...
  HOLD_OP TIME_ZERO LEFT_WIN RIGHT_WIN ...
  B A OP_COLOR OP_COLOR_CNT MAX_OP_AX MIN_OP_AX;
 close(gcf);
end

function setergax
%=======================================================================
global ERG_AX ERG_TIME ERG_SIGNAL ERG_NAME

xlim=[min(ERG_TIME) max(ERG_TIME)];
```

```
ylim=[min(ERG_SIGNAL) max(ERG_SIGNAL)];
set(ERG_AX,'FontSize',8,'Color',[1 1 1], ...
    'XLim',xlim,'YLim',ylim,'XGrid','on','YGrid','on');
xlabel('Time (ms)','FontSize',8);
ylabel('Amplitude (uV)','FontSize',8);
title(['ORIGINAL (',ERG_NAME,') ERG'],'FontSize',8);
return;

function setfreqax
%==============================================================
global ERG_SPEC_AX FREQ_VEC ERG_FREQ ERG_NAME
set(ERG_SPEC_AX,'FontSize',8,'Color',[1 1 1],'XColor',[0 0 0],'YColor',[0 0 0]);
  axis([min(FREQ_VEC) max(FREQ_VEC) ...
      min(20*log10(abs(ERG_FREQ(1:256))))
max(20*log10(abs(ERG_FREQ(1:256))))])
  grid;
  xlabel('Frequency (Hz)','FontSize',8);
  ylabel('Magnitude (dB)','FontSize',8);
  Fop_plot=['WINDOWED (',ERG_NAME,') ERG FREQUENCY SPECTRUM'];
  title(Fop_plot,'FontSize',8);
return;


function setopax
%==============================================================
global OP_AX OP
set(OP_AX,'FontSize',8,'Color',[1 1 1],'XGrid','on','YGrid','on');
xlabel('Time (ms)','FontSize',8);
ylabel('Amplitude (uV)','FontSize',8);
title('EXTRACTED OP','FontSize',8);
return;
```

B.7 fergferg.m

```
function fergferg()
% Conditions Erg signal by attaching padding signals to a windowed Erg
% signal. After conditioning the signal is filtered using a second order
% bilateral butterworth filter. The filter is bandpass with corner frequencies
% [FH, FL] hz. The program is called by auto.m or winerg.m. After filtering
% the padding signals are removed. The subroutine passes variables.
%
% Peter Derr, Dr Andrew Meyer and Dr Edward Haupt 11/13/97
%==============================================================
global ERG_DATA ERG_SIGNAL ERG_TIME LENGTH_SIGNAL METHOD
global SAMPLE_FREQ OP APP B A TIME_ZERO TC LEFT_WIN
```

```
global RIGHT_WIN FH FL
%----------------------------------------------------------------------
clear sig size_sig
time=ERG_TIME(TIME_ZERO:LENGTH_SIGNAL);
sig=ERG_SIGNAL(TIME_ZERO:LENGTH_SIGNAL);
size_sig=length(sig);
%----------------------------------------------------------------------
clear leftwindow rightwindow min_sig locatemin sig_win
if METHOD==1
  if LEFT_WIN-TIME_ZERO <= 0
    leftwindow=1;
  else
    leftwindow=(LEFT_WIN-TIME_ZERO)+1;
  end
  rightwindow=(RIGHT_WIN-TIME_ZERO)+1;
elseif METHOD==2
  for i=1:size_sig
    if time(i) >= 60
      rightwindow=i;
      break;
    end
  end
  min_sig=min(sig(1:(round(rightwindow/2))));
  for i=1:rightwindow
    if sig(i) == min_sig
      locatemin=i;
      break;
    end
  end
  if locatemin-2 >= 1
    min_sig=min(sig(locatemin-2:locatemin+2));
    for i=locatemin-2:locatemin+2
      if sig(i)==min_sig
        leftwindow=i;
        break;
      end
    end
  elseif (locatemin-2 < 1)
    leftwindow=locatemin+1;
  end
end
%----------------------------------------------------------------------
sig_win=sig(leftwindow:rightwindow);
%----------------------------------------------------------------------
%Lowpass filter windowed ERG signal to determine trough to peak amplitude
clear sigcond Amin Amax
```

```
[sig_l,psz_left]=padleft(sig_win);
sig_left=smooth(sig_l);
[sig_r,psz_right]=padright(sig_win);
sig_right=smooth(sig_r);
%-------------------------------------------------------------------
sigcond=[sig_left;sig_win;sig_right];
clear filt_forward filt_for_flip filt_reverse filt_rev_flip
[BLP,ALP]=butter(2,60*(2/SAMPLE_FREQ));
filt_forward=filter(BLP,ALP,sigcond);
filt_for_flip=filt_forward(length(filt_forward):-1:1);
filt_reverse=filter(BLP,ALP,filt_for_flip);
filt_reverse(1:psz_right)=[];
filt_rev_flip=filt_reverse(length(filt_reverse):-1:1);
filt_rev_flip(1:psz_left)=[];
Amin=min(filt_rev_flip);
Amax=max(filt_rev_flip);
APP=Amax-Amin;
%-------------------------------------------------------------------
clear filt_forward filt_for_flip filt_reverse filt_rev_flip
sigcond=sigcond-sigcond(1);
filt_forward=filter(B,A,sigcond);
filt_for_flip=filt_forward(length(filt_forward):-1:1);
filt_for_flip=filt_for_flip-filt_for_flip(1);
filt_reverse=filter(B,A,filt_for_flip);
filt_reverse(1:psz_right)=[];
filt_rev_flip=filt_reverse(length(filt_reverse):-1:1);
filt_rev_flip(1:psz_left)=[];
%-------------------------------------------------------------------
clear index pady YO TO
OP=[];
if leftwindow >= 2
  for index=1:leftwindow-1;
    pady(index)=0;
  end
  pady=pady';
  YO=[pady; filt_rev_flip];
  YO=YO(:);
else
  YO=[filt_rev_flip];
  YO=YO(:);
end
n=length(YO);
for i=1:n
  TO(i)=time(i);
end
TO=TO(:);
```

```
OP=[TO YO];
return;

function [padsig,padsz]=padleft(signal)
% Generates the left padding signal. The function is called by
% function fergferg.m. The subroutine passes variables.
% Variables:
%  Input:
%    sig: Windowed Erg signal.
%
%  Output:
%    padsig: Padding signal.
%    padsz: Size of padding signal.
%
% Peter Derr, Dr Andrew Meyer and Dr Edward Haupt 12/16/97
%=============================================================
clear padsig padsz
data_sz=length(signal);
for i=2:data_sz;
  padsig(i-1)=(2*signal(1))-signal(i);
end
padsig=padsig';
padsig=padsig(length(padsig):-1:1);
padsz=length(padsig);
clear signal data_sz
return;

function [padsig,padsz]=padright(signal)
% Generates the right padding signal. The function is called by
% function fergferg.m. The subroutine passes variables.
% Variables:
%  Input:
%    sig: Windowed Erg signal.
%
%  Output:
%    padsig: Padding signal.
%    padsz: Size of padding signal.
%
% Peter Derr, Dr Andrew Meyer and Dr Edward Haupt 12/16/97
%=============================================================
clear padsig padsz
signal=signal(length(signal):-1:1);
data_sz=length(signal);
for i=2:data_sz;
  padsig(i-1)=(2*signal(1))-signal(i);
end
```

```
padsig=padsig';
padsz=length(padsig);
clear signal data_sz
return;

function [filtpad]=smooth(filt_pad)
% Generates a filtered padding signal.
%
% Peter Derr, Dr Andrew Meyer and Dr Edward Haupt 12/16/97
%===========================================================
global SAMPLE_FREQ
[BLP,ALP]=butter(2,60*(2/SAMPLE_FREQ));
[filt_pad_l,psz_l]=padleft(filt_pad);
[filt_pad_r,psz_r]=padright(filt_pad);
filt_pad=[filt_pad_l;filt_pad;filt_pad_r];
forward=filter(BLP,ALP,filt_pad);
flip=forward(length(forward):-1:1);
reverse=filter(BLP,ALP,flip);
reverse(1:psz_r)=[];
filtpad=reverse(length(reverse):-1:1);
filtpad(1:psz_l)=[];
```

B.8 optparam.m

```
function p_out=optparam(p_in)
% Optimizes the parameter vector p_in to obtain a best fit
% of the OP model to the extracted OP. The simplex algorithm is
% utilized. The program is called by either Modelop.m or Auto.m.
% The subroutine passes varibles.
%
% Variables:
%  Input:
%    p_in: Initial OP model parameter vector.
%
%  Output:
%    p_out: Optimized OP model parameter vector.
%
% Peter Derr, Dr Andrew Meyer and Dr Edward Haupt 12/18/97
% Required M-Files: fmins.m
%-----------------------------------------------------------------
ptol=2.0;    % parameter-tolerance
stol=4.0;    % function tolerance
steps=2000;  % maximum number of steps.
display=0;   % steps are not displayed (faster). Otherwise, =1.
ops=[display ptol stol 0 0 0 0 0 0 0 0 0 0 steps];
```

```
p_out=fmins('paramcmp',p_in,ops);
return;
```

B.9 paramcmp.m

```
function err=paramcmp(pm)
% Calculates error between the optimized parameter model and the extracted
% OP. The error is calculated by the square root of the sum of the squares
% method. The program is referenced by optpm.m to be used by the simplex
% algorithm, fmins.m. The program passes variables.
%
% Variables:
%  Input:
%   pm: OP model pmeter vector from fmins.
%
%  Output:
%   err: error between the OP model and the extracted OP.
%
% Peter Derr, Dr. Andrew Meyer and Dr. Edward Haupt 12/18/97

% Global variables generated in modelop.m
%   OP_AX:      OP plot axes object.
%   P_OPT:      Optimized op parameter vector.
%   E_OPT:      Error between the extracted op and op model.
%   M_OUT:      Op model data matrix.
%   LAST_OP:    The previous op model data matrix (used for plotting)
%---------------------------------------------------------------------------
% modelop.m generated global variables
global P1_OPT P2_OPT P3_OPT P4_OPT P5_OPT ERFIT_OPT ...
global RELAMP_OPT PLOTOP P6_OPT P7_OPT P8_OPT P9_OPT ...
global P_OPT OPTVAL METHOD OP ERFIT_INIT RELAMP_INIT ...
global APP FLAG OP_AX TEMP_MODEL TEMP_ERR
%-----------------------------------------------------------------------
% paramcmp.m generated global variables
global P_OPT E_OPT M_OUT LAST_OP
%-----------------------------------------------------------------------
n=length(OP(:,1));
for index=1:n
  t=OP(index,1);
  model_data(index)=eval(char(TEMP_MODEL),t);
end
clear index t n
model_data=model_data(:);
ecmp=model_data-OP(:,2);
n=length(OP);
err=((1/n)*(ecmp'*ecmp)^(1/2));
```

```
E_OPT=err;
M_OUT=model_data;
if (METHOD==1 & FLAG==0)
  set(ERFIT_INIT,'String',num2str((E_OPT/pm(TEMP_ERR))*100));
  set(RELAMP_INIT,'String',num2str((pm(TEMP_ERR)/APP)*100));
elseif (METHOD==1 & FLAG==1)
  maxpm=max(size(pm));
  if ~isempty(LAST_OP)
    set(PLOTOP,'Ydata',LAST_OP,'Color','w');
    drawnow;
  end
  set(PLOTOP,'Ydata',OP(:,2),'Color','r');
  drawnow;
  set(PLOTOP,'Ydata',M_OUT,'Color','k');
  drawnow;
  for i=1:maxpm
    set(OPTVAL(i),'String',num2str(pm(i)));
  end
  set(ERFIT_OPT,'String',num2str((E_OPT/pm(TEMP_ERR))*100));
  set(RELAMP_OPT,'String',num2str((pm(TEMP_ERR)/APP)*100));
end
LAST_OP=M_OUT; P_OPT=pm;
return;
```

REFERENCES

[1] Jack Yellot, "Important Dates in Vision Science", http://www.socsci.uci.edu/cogsci/vision/yellot_dates.html, pp.1-9,October 7, 1998

[2] F.H. Adler, *Physiology of the Eye*, The C.V. Mosby Company, Saint Louis, Mo., 1975, pp. 453-499.

[3] E. Bruce Goldstein, *Sensation & Perception*, fifth edition, Brooks/Cole Publishing Company, New York, NY., 1999, pp. 29-70.

[4] Neal S. Peachey, Yoshinobu Goto, Muayyad R. Al-Ubaidi and Muna I. Naash, "Properties of the mouse cone-mediated electroretinogram during light adaptation", *Neuroscience Letters* 162, pp. 9-11, 1993.

[5] Hahn Lance, "The retina: gross anatomy", http://retina.anatomy.upen.edu/~lance/eye/retina_gross.html, September 14, 1998

[6] Michael F. Marmor, MD. and Eberhart Zrenner, "ISCEV Standards, Recommendations and Guidelines", http://www.ukl.uni-freiburg.de/aug/iscev/standards/er.html, July 9, 1998

[7] Perry Speros, Ph.D., and James Price, M.D., Ph.D., "Oscillatory Potentials. History, Techniques and Potential Use in the Evaluation of Disturbances of Retinal Circulation", *Survey of Ophthalmology*, Volume 25, Number 4, pp. 237-252, January-February 1981.

[8] Henry Heyen, Lillenmor Wachtmeister and Dirk Van Norren, "Origin of the Oscillatory Potentials in the Primate Retina", *Vision Research*, Volume 25, No. 10, pp. 1365-1373, 1985.

[9] Calvin K. Adams and William W. Dawson, "Fast Retinal Potential Luminosity Functions", *Vision Research*, Volume 11, pp. 1135-1146, 1971.

REFERENCES
(Continued)

[10] K. Van der Torren, G. Groeneweg and G. Van Lith, "Measuring oscillatory potentials: Fourier analysis", *Documenta Ophthalmologica*, Volume 69, pp. 153-159, 1981.

[11] K. Van der Torren, G. Groeneweg and G. Van Lith, "Measuring oscillatory potentials during the course of arterial occlusion: a method based on Fourier analysis", *Documenta Ophthalmologica*, Volume 70, pp. 199-203, 1988.

[12] Andrew U. Meyer, Edward J. Haupt, Zhongquan Li Ken-Gen Lu and Howard Louis, "Clinical Electrophysiology of the Eye: Physiological Modeling and Parameter Estimation", in *Proc.* IEEE International Conference on Systems, Man and Cybernetics, Volume II, November 14-17, 1989

[13] E.J. Haupt, A.U. Meyer and M.G. Brigell, "Retrospective comparison of CRVO diagnosed eyes which did and did not later show neovascularization by an OP signal model", The Association for Research in Vision and Ophthalmology (ARVO) Annual Meeting, Fort Lauderdale, Florida, May 1997.

[14] Jeffrey R.M. Kunz, M.D. and Asher J. Finkel, M.D., *The American Medical Association: Family Medical Guide*, The Readers Digest Association by permission of Random House Inc., New York, 1987, pp. 329-331

[15] "Eye Anatomy", http://www.eyeclinicfbks.com/ENFANAT1.html, June 5, 1998

[16] E.J. Haupt, P. Derr, A.U. Meyer and M.G. Brigell, "Model fitting for a frequency parameter in signal conditioned oscillatory potential from electroretinograms of CRVO eyes", The Association for Research in Vision and Ophthalmology (ARVO) Annual Meeting, Fort Lauderdale, Florida, May 1998.

[17] Gregory S. McGillem, Clyde Guidry and Ramon F. Dacheux, "Antigenic Changes of Rabbit Retinal Müller Cells in Culture", *IOVS*, vol. 39, no. 2, pp. 1453-1461, 1998.

[18] Stephen C. Massey, Dianna A. Redburn and M.L.J. Crawford, "The Effects of 2-Amino-4-Phosphonobutyric Acid (APB) on the ERG and ganglion Cell Discharge of Rabbit Retina", *Vision Research.*, vol. 23, no. 12, pp. 1607-1613, 1983

REFERENCES
(Continued)

[19] Rad Tzekov, "luminance",ttzekov@retinafoundation.org,,January 11,1999

[20] Lillemor Wachtmeister, "Basic research and clinical aspects of the oscillatory potentials of the electroretinogram", *Documenta Opthalmologica*, 66, pp. 187-194, 1987