



# Computer assisted electroretinogram data acquisition and analysis

---

**Nick Balge, Andrew Dias, Whitney Johnson, Dan Jonovic**

**Advisor: Professor Mitch Tyler**

**Client: Dr. Bikash Pattnaik**

**10/20/2008**

# Contents

- Table of Figures ..... 3
- 1. Background ..... 4
- 2. Applications/Motivation: ..... 5
- 3. Problem Statement: ..... 6
- 4. Lab layout: ..... 6
- 5. Design constraints: ..... 7
- 6. The current interface ..... 8
- 7. Design 1: Tweak Existing Interface ..... 9
  - A. Functionality ..... 9
  - B. GUI Modification ..... 10
- 8. Design 2: Complete GUI Overhaul ..... 11
  - A. GUI Plans ..... 11
  - B. GUI Design ..... 12
- 9. Design 3: Start over from scratch ..... 13
  - A. Existing Solutions ..... 13
  - B. Programming Languages ..... 14
  - C. The Design ..... 14
- 10. Design Matrix ..... 15
  - A. Definition of Criteria ..... 15
- 11. Potential Problems ..... 16
- 12. Future Work ..... 17
- 13. Conclusion ..... 17
- 14. References ..... 18
- 15. Appendix A: Product Design Specifications ..... 19

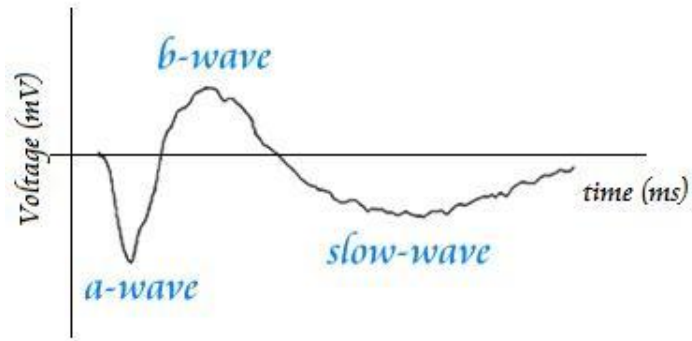
## Table of Figures

<b>Figure 1:</b> Example of ERG wave .....	4
<b>Figure 2:</b> Measuring the amplitudes and times of waves.....	5
<b>Figure 3:</b> Organization of our client’s lab.....	6
<b>Figure 4:</b> A solution valve opener .....	7
<b>Figure 5:</b> GUI for the original interface.....	8
<b>Figure 6:</b> The existing backend .....	9
<b>Figure 7:</b> Possible GUI Overhaul .....	11
<b>Figure 8:</b> The Design Matrix.....	16

## 1. Background

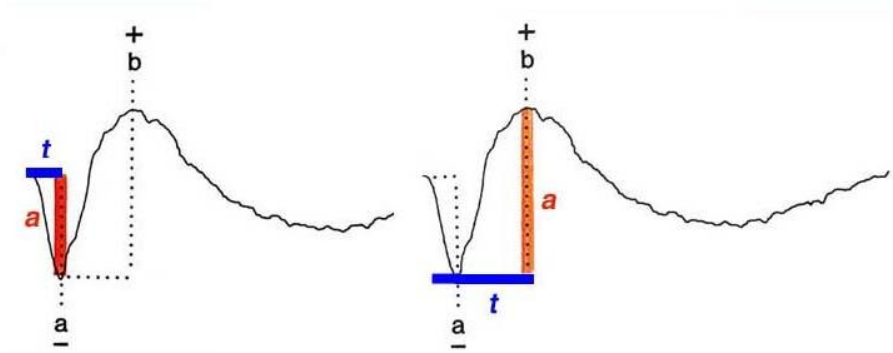
An electroretinogram (ERG) is used to measure the electrical responses of various cell layers in the retina (Medline, 2007). This response is gathered by placing an electrode either on the retina or the skin surrounding the eye, connecting the other electrode to ground, and stimulating the retina with light; the intensity and wavelength of the light stimulus is often varied.

The result of the light stimulus is a triphasic wave that consists of a, b, and slow-waves (Figure 1). These different components represent separate layers



**Figure 1:** Example of ERG wave [Creel, 2008]. The a-wave is negative, the b-wave is positive, and the slow-wave is negative.

called the “late receptor potential” and reflects the physiological health of photoreceptors in the outer retina. The b-wave reflects the health of the outer cell layers of the retina and the slow-wave provides information on the bipolar and ganglion cells (Creel, 2008; Ye and Goo, 2007). Measurements taken from the wave include the amplitude of the a-wave from time zero, the amplitude of the b-wave with respect to the a-wave, the time between the flash and trough of the a-wave, and the time between the flash and the peak of the b-wave (Figure 2). The measurement of the time from flash onset to wave peak is called the implicit time. The b/a wave ratio is a good indicator of ischaemia, or insufficient blood flow in a central retinal vein obstruction (Matsui, Y *et al*, 1994).



**Figure 2:** *Measuring the amplitudes and times of waves*

There are a few ways to record the ERG. The most common way is to dark-adapt the retina, attach electrodes with dim red illumination, and record the ERG with dim blue and red flashes and bright white flashes. The test would be repeated with moderate background illumination so that the retina is light-adapted (Creel, 2008). The use of log filters to reduce light intensity limits the ERG to recording rod activity because they are much more sensitive than cones.

Different solutions are known to block various cell types in the retina. The pharmacological agent L – AP5 has been shown to significantly reduce the b-wave amplitude in ERGs produced from flashes above the intensity 100 q/rod (Green and Kapousta-Bruneau, 2006). Other chemicals reduce cone concentrations, thereby limiting the cell types responding to the stimulus. This type of research helps to isolate different cells layers, making it easier to distinguish problem areas of retina.

## **2. Applications/Motivation:**

ERGs have been used to diagnose a number of eye diseases. Retinal physiology can also reflect systemic disorders such as liver and kidney disease (Creel, 2008). These techniques have been in use for a number of years. Currently, there is a need for many new drugs for eye disease to be developed and tested. An ERG provides an optimal research environment because tests can be conducted on an

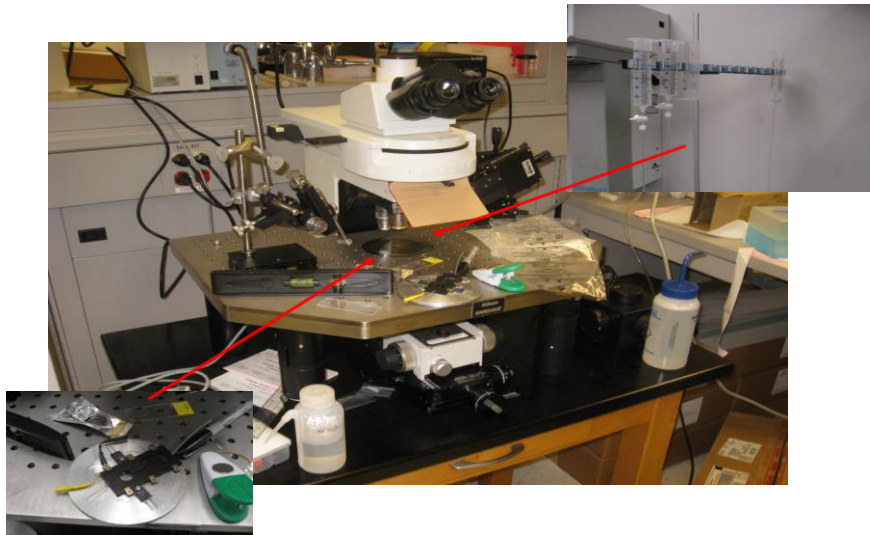
isolated retina, which allows various solutions to be in contact with the sample. Although systems exist for the measurement of ERGs in animals, it is important to create a computer program that will help integrate components of a given test design and allow researchers to visualize and analyze their data.

### 3. Problem Statement:

A computer interface will be designed to integrate aspects of research design with ERG data acquisition and analysis. These aspects include liquid exchange through valve control and incorporation of a logarithmic light wheel along with the need to export data files and graphs to a separate program to store the information in a simple and organized format. See Appendix A for more details.

### 4. Lab layout:

Our design constraints follow the setup of our client's lab (Figure 3).



**Figure 3:** Organization of our client's lab. A rat retina is placed on a slide (bottom left) under the microscope to record the ERG. The retina is perfused with drugs or solution from the liquid delivery system (top right).

**Figure 4:** A solution valve opener allows controlled flow of a chemical (Audie Technology, 2008).



Our client places electrodes on the retina and records the ERG. Light amounts are set via a light wheel and liquids are delivered via the solution delivery system. Up to eight solutions can be prepared, but the retina will only be perfused by one solution at a time (Figures 3, 4). The light wheel, solution delivery system, and ERG are all connected to a computer.

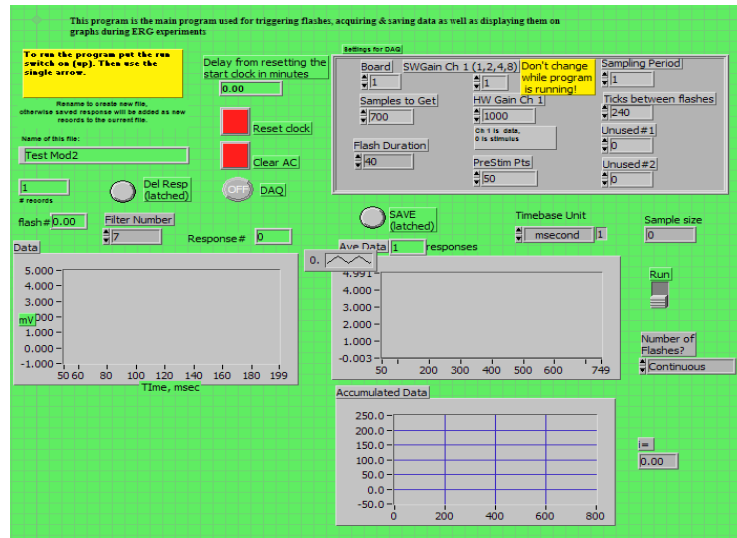
## 5. Design constraints:

The interface should help perform retinal research, and constraints are defined by acquiring suitable ERGs under different conditions. The waveform of the ERG depends on whether the retina is dark or light adapted, so testing will be performed at different light intensities. This system must be able to change light intensities over the retina. Additionally, different drugs must be perfused over the retina one at a time. Our interface must be able to manage the solution delivery system and switch between drugs. It must also ensure that more than one drug is not delivered at a time and that one drug is always being delivered so that the retina is not allowed to run dry.

Further, real-time graphical display and the ability to export raw data must be supported. It is desirable to view the ERG waveform as it is being recorded to ensure that data is being acquired. Raw data should be exported into another format for further analysis because LabVIEW is not the optimal environment to perform rigorous calculations. Although these features are required additions, functionality from the original interface should not be removed.

## 6. The current interface

The user of the interface interacts with the graphical user interface (GUI), also known as the front end of the system (Figure 5).



**Figure 5:** GUI for the original interface

The main components of the GUI are the functionality and the usability. This interface displays graphs and allows the user to change some settings like sample duration and the number of consecutive samples – it has some basic functionalities. However, what this interface lacks is:

- 1) The ability to change the light intensity by letting the user choose the position of the filter wheel
- 2) Options to set and change the solution that will be delivered
- 3) An option to export data into another format.

To make the GUI more usable, the graphs, buttons, and forms may need to be reorganized. The interface looks cluttered because there are three graphs that take up a lot of room and because there are many options that have to be changed by keyboard entry. If fewer boxes were used and alternative methods of input were considered, the GUI may be easier to use. Providing tabs may be a way to reduce the number of boxes the user views on one screen and would be a way to make boxes and font larger, increasing usability. Tradeoffs between allowing the user to manipulate many features on one screen and size of boxes and forms need to be considered and optimized.



## 7. Design 1: Tweak Existing Interface

The first design is based on keeping and slightly modifying the existing backend of the program that the client and his associates have developed (Figure 6). Because the graphical programming has already been written and functions properly, this would save time. Also, programming in LabVIEW is advantageous because it efficiently integrates hardware and software, thereby promoting data collection.

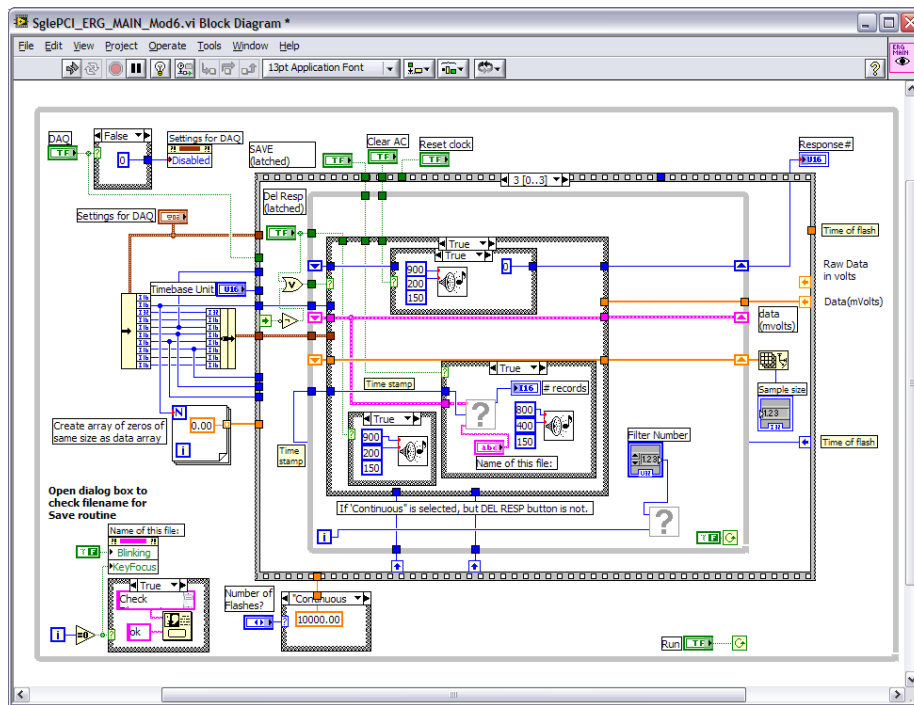


Figure 6: The existing backend

### A. Functionality

The most important part of each individual design plan is to add the necessary functionality to the client's system. This includes all of the issues mentioned in the PDS such as solution valve integration, light wheel control, and data storage. In this design, these tasks would be accomplished by rewriting some of the LabVIEW code. As mentioned above, the majority of the code would remain the same, and only enough would be rewritten as is necessary to include all of the functionality.

## B. GUI Modification

The aspect of the first design that makes it most unique involves the modification of the graphical user interface. In this plan the GUI would undergo only minor modifications. This would allow for the correction of some of the simpler features which could use some improvement. Again this option would also save time, which would be advantageous if integration of the necessary functionalities takes longer than expected (Hobart).

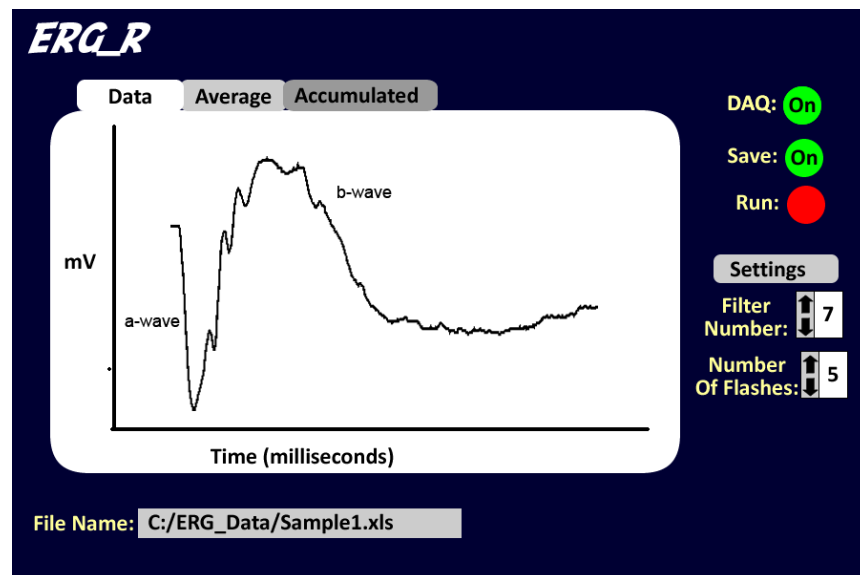
The first design component would be to rearrange all of the controls into places where the user would be most likely to look for them. This process would involve grouping related commands together. Given a logical arrangement of controls, the user would be able to find the target button with the greatest possible speed and least possible frustration. Rearranging could also aid in clearly defining which items are displays and which are user controls. These modifications could be easily implemented, as LabVIEW allows the dragging of objects to their necessary positions.

Another simple improvement would be to change the color scheme of the interface. Because the system operates in the dark, it should have a sufficient level of contrast. While the current display does have contrast, the color scheme could potentially create frustration or annoyance if the user is staring at it for too long. For this reason, an improvement of color harmony would be beneficial. In modifying the color scheme, steps would be taken to decrease the overwhelming feeling while making sure not to create a color scheme that fails to fully capture the user's attention in the first place (Morton, 2008). In redesigning the color of the GUI care would be taken to find an adequately contrasting scheme to meet the client's needs, while adhering to the previously mentioned principles.

Finally another fairly simple fix would involve changing the existing font sizes. Overall, font size would be increased because the most important design criteria in a visible element is readability. In some cases this would involve a minor rewording of the labels so that they don't take up too much space. After making sure that all of the words can be accurately read, font sizes could be altered to convey their importance to the execution of the experiment. This would direct the user's eyes to the buttons which he should be using the most; further increasing ease and speed of execution.

## 8. Design 2: Complete GUI Overhaul

The second design plan incorporates some of the same features as the first. This design would again build off of the existing graphical programming that the client has written in LabVIEW. All of the necessary functionalities would again be added to this program with few other changes to the code.



*Figure 7: Possible GUI Overhaul*

### A. GUI Plans

As with design one, design two is unique in its approach to the GUI. In this case, however, the GUI would undergo a complete overhaul (Figure 7). This process would allow for a fresh start and the

opportunity to include all of the modifications from the first design while additionally adhering to various GUI design principles that would otherwise be out of reach if only minor changes were possible. This plan would be more time consuming but would open the door to creating a user friendly interface.

## **B. GUI Design**

A staple characteristic of good GUIs is simplicity. The user should not feel overwhelmed by the number of visible buttons on program start. Also, navigating through a large number of controls will usually take longer than desired. To avoid this, the number of items on the main GUI would be greatly reduced. Only the most important features would be included so that the user's eyes fall directly on these items (Hobart).

Along with simplicity, the GUI must also have functionality. Completely removing some of the features from the old GUI would be unbeneficial and might even break the requirements for the design.

Therefore, rather than getting rid of controls to increase simplicity, various controls would be moved to separate menus. Using menus, even more controls could be added while still encouraging a strong feeling of simplicity. Another benefit of adding menus would be to foster the organization of the GUI as a whole, by again grouping related commands together as mentioned for the first design. A related change would also involve moving displayed graphs to tabbed windows rather than trying to display them all at once. This would greatly increase their possible size and visibility.

Another equally important quality of a successful GUI is consistency. As various windows and menus are incorporated, care must be taken to make sure that the layout of each screen resembles the layout of the others. Another point of importance is to ensure consistency of labels between windows. Even slight mislabeling could cause the user to become confused about a value that is inputted and run the experiment in a way other than is intended.

Finally, familiarity is a quality that aids a user in operating any GUI. Concerning the GUI as a whole, its layout and appearance should be similar to frequently used programs. However, when incorporating new ideas, care should be taken not to create something entirely unique, even if it works. Familiarity at the level of single controls is also important and would be improved by changing the way that certain commands are inputted. For example, a box that previously required the user to type in a value manually or adjust the value with an arrow might be better suited to be a drop down menu. The advantage of such menus is that the user is only able to set valid values, adding to the efficiency of program navigation.

Aside from these encompassing concepts of GUI design, creating one from scratch would permit the addition of other optional features are deemed necessary as the project progresses. In these ways, the second design option increases the potential for a satisfactory user interface.

## **9. Design 3: Start over from scratch**

The third design option encompasses an entire redesign of the project and involves the comparison of different solutions available as well as other programming languages. This establishes which computing environment would best address the problem statement.

### **A. Existing Solutions**

Electro-Diagnostic Imaging offers an ERG measuring solution, though it does not allow the capability of interfacing with additional hardware. While it performs similar measurement operations, it is designed for clinical settings and not a research environment (Electro-Diagnostic).

AD Research has both hardware and software systems developed for data analysis and acquisition (AD Instruments). This is similar to LabVIEW, though their LabChart software already has measurement

capabilities and data analysis built in. While there would not be software programming involved in this system, it does not allow the addition of capabilities that are required, including solution delivery and light wheel control.

## **B. Programming Languages**

Using a Unix-based environment would allow better control over timings of actions that need to be performed in milliseconds. While the flash duration and other settings are on the order of 40ms, the less precise Windows time management which meets the needs of this project. Also, there has been a stated preference for a Windows-based system since many of the research labs doing similar work operate in a Windows environment.

Visual Basic and C++ are two alternative programming languages that would allow much tighter control over the code through manually implementing loops and data analysis techniques. Through this management, it would also be possible to reduce the amount of resources this program would use compared to a similar LabVIEW program.

LabVIEW allows for both the hardware and software interactions needed to produce the functionality needed by the client. The programming interface is more intuitive than that of C++ or Visual Basic, although it doesn't offer the same level of control. In addition, many functional tasks exist in LabVIEW prepared as Sub-VI's to be easily implemented into other designs. For these reasons, more complex analysis tasks can be performed with significantly less development time using LabVIEW.

## **C. The Design**

This design builds off the second design option (Figure 7) and involves using LabVIEW in a Windows environment for the platform. It would encompass starting with a new program, recreating the basic

functionality of receiving signals, displaying the graphs, averaging multiple tests together, and taking input from the user. Afterwards, the requested functionality would be added along with the implementation of the new user interface.

The advantage of this design is that starting over provides for complete control over the code, resulting in a modular system that can be more easily imported onto other platforms. It is also possible that some commands could be simplified, making the program more efficient. These benefits do not directly impact the user of the program, but will be noticed if future modification occurs. Modifications could include the changing of a DAQ or addition of another feature – though neither of these possibilities are likely in the near future.

## 10. Design Matrix

After establishing three distinct design options, different criteria were established and assigned weights according to their importance and relevance within the project (Figure 8). The criteria are: functionality, ease of use, complexity, aesthetic appeal, and modularity.

### A. Definition of Criteria

- **Functionality:** How well does the final product address the needs expressed by the client? This includes the addition of valve control, light wheel control, and data storage.
- **Ease of Use:** How intuitive is the interface to use? The graphs should be large and visible along with all text. Similar items should be grouped together and formatting consistent.
- **Complexity:** What is the level of difficulty involved with developing and implementing this solution? Is it an optimal use of available resources?
- **Aesthetic Appeal:** The user spends a significant amount of time in front of the interface setting up experiments, running them, and analyzing data. In addition to being intuitive (Ease of Use),

the user should be welcomed into the program with appealing colors, a good level of contrast and legible labels in both light and dark environments.

- **Modularity:** This programming technique addresses the need to separate and group functional elements of the program together (Lesiecki). The resulting program is more easily modifiable and adaptable since modules require one change instead of individually addressing every location in the program where that module is used.

Criteria (weight)	Tweak	Overhaul	Start over
Functionality (30)	30	<b>30</b>	30
Ease of Use (30)	10	<b>30</b>	30
Complexity (20)	20	<b>10</b>	0
Aesthetic Appeal (15)	5	<b>15</b>	15
Modularity (5)	2	<b>2</b>	5
<b>Total (100)</b>	<b>67</b>	<b>87</b>	<b>80</b>

**Figure 8: The Design Matrix**

The weights used in the matrix are relative since categories such as aesthetic appeal have a subjective component and cannot realistically obtain a perfect score. The 'Overhaul' and 'Start Over' design options are very similar, as the front end is identical. The difference is a better underlying program (back end), which also results in a much longer development time. For this reason, Design 2, the GUI overhaul has been chosen as the route to proceed.

## 11. Potential Problems

Some of the hardware components that were ordered have not arrived yet. While the hardware is not an aspect of this design, it may pose a problem during testing because it is not possible to debug the functionality without the underlying hardware. It may not be possible to set up the GUI according to



universal design principles because there may be some restrictions in the graphical interface of LabVIEW. Based on the current design, there is the possibility that additional functionality may conflict with structures in the current code, requiring an overhaul of the backend of the program.

## **12. Future Work**

The most important aspect of the project is adding the functionality needed by the client to the current program. These features include the light wheel control, solution delivery, and data storage. In addition, a new user interface will be developed through comparison of the different options available in LabVIEW using principles of human factors and ergonomics. It is also important to verify that the final design is visible in the lab environment, which involves a dark room and a filter over the screen. The project code will also be examined to verify its functionality and assist in linking the new features together.

## **13. Conclusion**

Overhauling the current interface while adding required functionality was selected for the design because it addresses the design requirements while improving ergonomics and the user experience over the current design. This provides the capabilities needed to properly record and analyze the ERG, potentially leading to new solutions for debilitating eye disease.

## 14. References

- AD Instruments. "ADInstruments Products". 2008.  
<<http://www.adinstruments.com/products/research>>
- Audie Technology. 2008. "Flow Pro Automatic Valve Opener". <<http://www.audiotech.com/flow-pro/valve-opener-flow-pro.html>>
- Creel, D. 2008. "Clinical Electrophysiology". John Moran Eye Center, University of Utah.  
<<http://webvision.med.utah.edu/ClinicalERG.html>>
- Green and Kapousta-Bruneau. 2006. "Evidence that L-APS and D,L-APH can preferentially block cone signals in the rat retina." *Visual Neuroscience*. 24(1):9-15.
- Electro-Diagnostic Imaging, Inc. 2008. "VERIS™ Science Software." <<http://www.veris-edi.com>>
- Hobart, J. "Principles of good GUI Design." <[http://www.iie.org.mx/Monitor/v01n03/ar\\_ihc2.htm](http://www.iie.org.mx/Monitor/v01n03/ar_ihc2.htm)>
- Kazaryan AA. 2007. "Functional characteristics of the retina in Glaucoma." *Acta Ophthalmologica Scandinavica* 85(240).
- Lesiecki, N. 01 Jan 2002. "Improve modularity with aspect-oriented programming." IBM.  
<http://www.ibm.com/developerworks/java/library/j-aspectj>
- Matsui, Y *et al.* 1994. Electroretinogram b/a wave ratio improvement in central retinal vein obstruction. *British Journal of Ophthalmology*. 78(3):191-8.
- Medline Plus. 2008. "Electroretinography." *US National Library of Medicine*.  
<<http://www.nlm.nih.gov/medlineplus/ency/article/003388.htm>>
- Morton, JL. 2008. "Color Theory." <<http://www.colormatters.com/colortheory.html>>
- Ye JH, Goo YS. 2007. "The slow wave component of retinal activity in rd/rd mice recorded with a multi-electrode array." *Physiol Meas* 28:1079-1088.

## 15. Appendix A: Product Design Specifications

**Team Members:** Dan Jonovic  
Andrew Dias  
Nick Balge  
Whitney Johnson

**Client:** Dr. Bikash Pattnaik  
**Advisor:** Prof. Mitch Tyler

**Function:** Electroretinogram (ERG) is the measurement of voltage across the retina in response to light flash. This project's goal is to create a system controlled through LabVIEW to monitor, control, and record values from an ERG corresponding to liquid exchange, light intensity, flash duration and frequency, as well as the waveforms.

### Client requirements:

- Filter number, corresponds to the light intensity
- Flash duration (in ms)
- Sample duration (in ms)
- Noise reduction (averaging data)
- Reset clock (timestamps)
- Liquid delivery through valve control of up to 8 different solutions
- Checkpoints in graphs for later analysis
- Export data to Excel for further analysis
- Animal testing may be performed
- Recording of a-waves, b-waves, and slow-waves

### Design requirements:

#### 1. Physical and Operational Characteristics

a. Performance requirements: Used daily on a Windows platform. It will likely be used by itself so RAM and processing power is not an issue. Delivery of one, and only one, liquid into the testing environment at a time.

c. Accuracy and Reliability: The interface is as accurate and precise as the selected programming environment is on the chosen operating system.

d. Life in Service: The software program should run without modification until the data acquisition hardware is changed.

e. Operating environment: The system should be functional in a Windows environment.

f. *Ergonomics*: The user interface on the computer should be intuitive: Items displayed should be easily understood with consistent formatting and a clean interface.

g. *Materials*: Windows-based computer with LabVIEW 7.6 installed.

h. *Aesthetics, Appearance, and Finish*: The program should be easy to use and functional with the following considerations:

- Text easily readable
- Contrasting Colors
- Intuitive Organization
- As simple as possible without dropping features

## 2. Production Characteristics

a. *Quantity*: 1

b. *Target Product Cost*: Software and hardware have already been purchased so no additional costs should be incurred.

## 3. Miscellaneous

a. *Standards and Specifications*: This project will not need to conform to FDA standards, as human testing is not going to be performed. It is necessary, however, to follow the RARC (Research Animal Resources Center) for animal testing.

b. *Customer*: The client would like to use a system in a Windows environment.

c. *Competition*: Our system is intended for research and there are no patents covering the scope of this project. There are some software solutions that have similar capabilities:

- CSH Protocols has developed a protocol for progressive ERG measurement during the development of larval zebrafish noninvasively.  
<<http://cshprotocols.cshlp.org/cgi/content/abstract/2008/4/pdb.prot4973>>
- VERIS EDI software is Macintosh based and has protocols for mfERG and transient ERG recording. Their software provides data analysis and storage capabilities but does not interface with the additional components (valves, etc.) that the client desires.  
<<http://www.veris-edi.com/>>
- AD Instruments LabChart Software is a data acquisition solution that, when combined with LabScope, offers powerful data analysis techniques. This software does not provide the hardware interface to communicate with valves or record other settings such as the light wheel.  
<<http://www.adinstruments.com/products/software/research/LabChart-Software>>