Rice/TMC/UH Computational Neuroscience REU

Final Report

Nonlinear Modeling of Locust Photoreceptors

Jung Uk Kang, Dept. of Computer Science, Brown University, Providence, RI

jk4@cs.brown.edu

Advisor: Fabrizio Gabbiani, Dept. of Neuroscience, Baylor College of Medicine, Houston, TX

gabbiani@bcm.edu

Summary

This report summarizes work done as part of the Rice/TMC/UH Computational Neuroscience REU. Rice/TMC/UH Computational Neuroscience REU is a program of research investigating neurobiology through math and computer science. This work was done in the Gabbiani laboratory at Baylor College of Medicine.

Introduction

Figure 1A

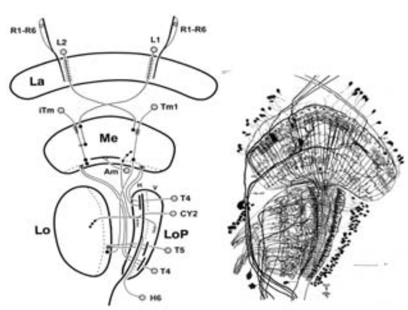


Figure 1B

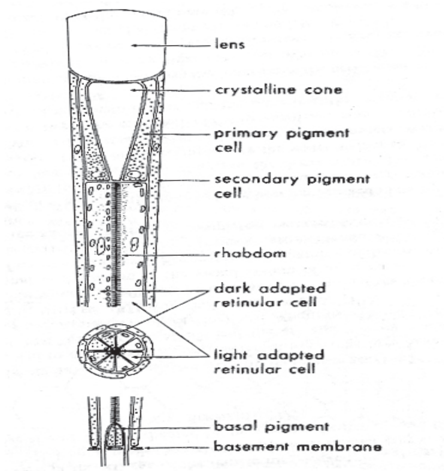
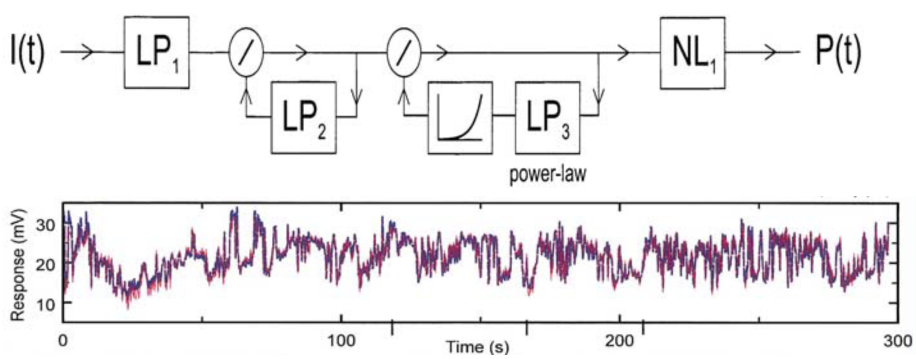
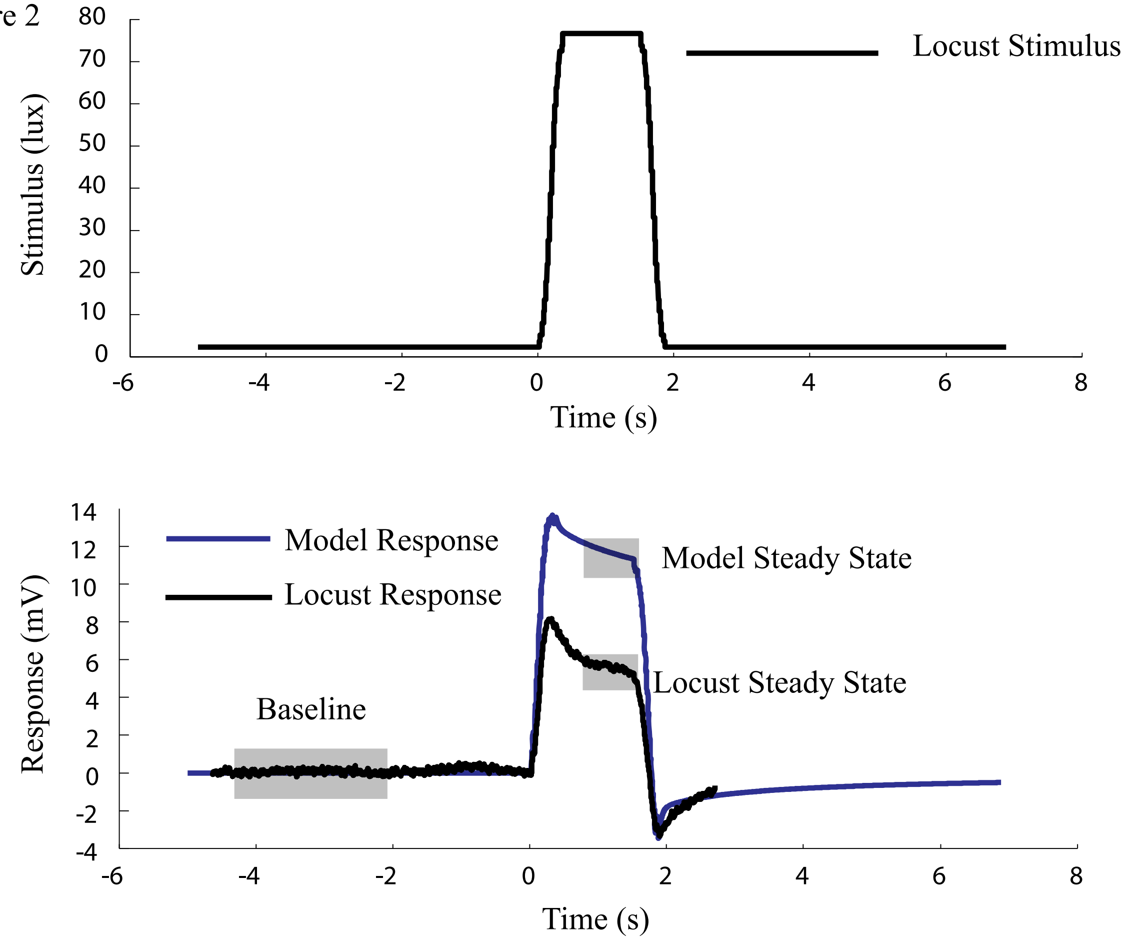


Figure 1C



The visual systems of animals have long been studied as models of how the neural circuits process information. This is because individual cells throughout the system can be recorded and their responses related to the light input presented. Photoreceptors, the cells responsible for transducing incident light energy into neural signals, are the first stage of this system. A schematic drawing of the insect visual system is shown in Figure 1A, and a sketch of a photoreceptor is shown in Figure1B. How photoreceptors respond to a wide range of light intensities and patterns has been well characterized over the years, and detailed models describing these responses have been developed. One particularly successful model was developed by Hans van Hateren and colleagues (van Hateren and Snippe 2001) based on responses from fly photoreceptors. This model describes the membrane potential fluctuations resulting from external luminance changes over 3 orders of magnitude using several sequential nonlinear filtering steps. The second stage of processing, Large Monopolar Cells (LMCs) in the lamina have also been well studied, and have also been successfully modeled using nonlinear-linear-nonlinear (NLN) cascade model (Juusola et al, 1995). The structure of the photoreceptor model and the response produced by the model are presented above in figure C. LP stands for Low-Pass filter, and NL stands for Non-Linear filter. LP1 and LP2 has tau1 and tau2 as its parameter respectively. LP3 has two parameters: k1 and k2. The output of LP3 is k1exp(k2*input).

Figure 2



With the parameter values (tau1 = 1. 69; tau2 = 71.8; k1 = 0.689; k2 = 9.07) for fly photoreceptor, the output of the model did not match with the actual data from photoreceptors. The actual response of a locust photoreceptor and that of the van Hateren model with default parameters is shown above. The model and the actual response show significant difference in steady state mV value. Since the van Hateren model is designed for fly photoreceptors, we need to find a new set of parameters of the model that can simulate the response of actual locust photoreceptors.

Goal

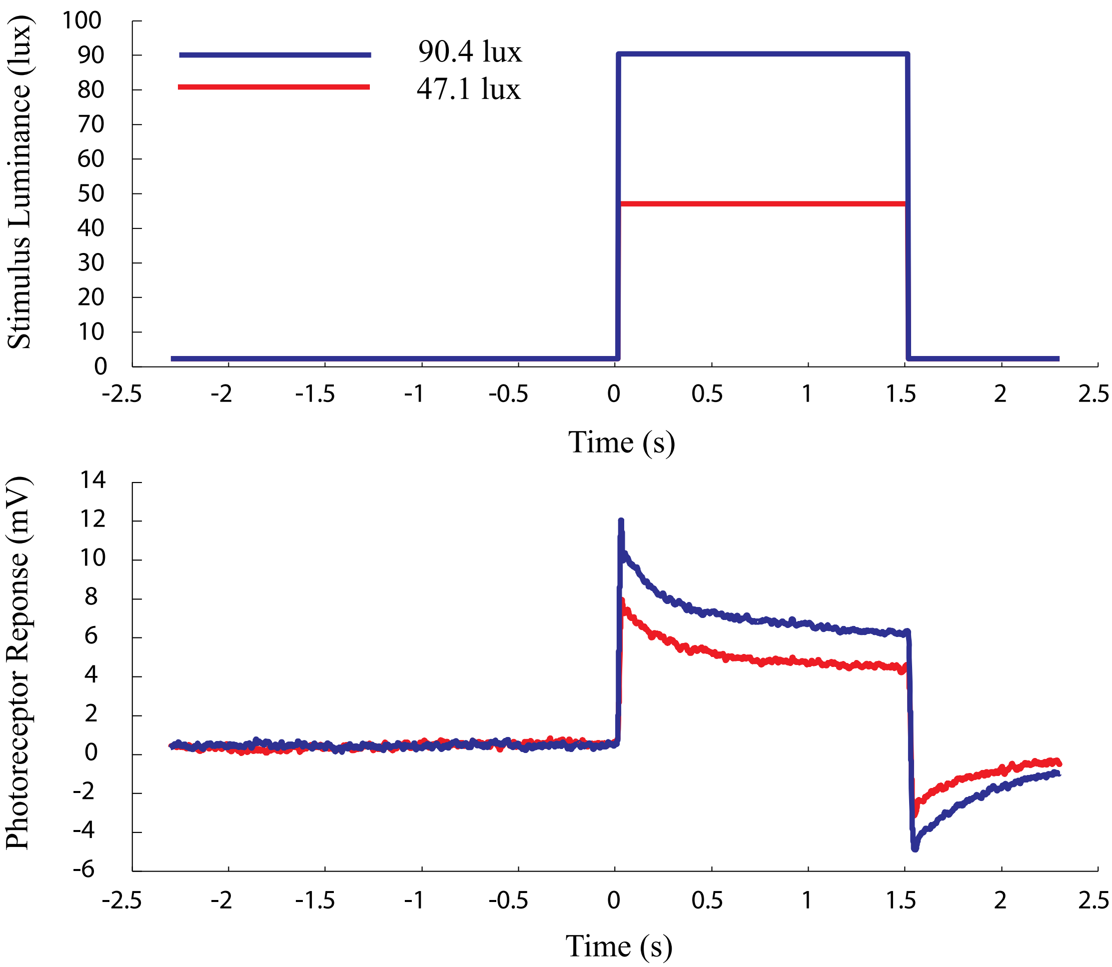
The goal of this project was to adapt existing models of early visual processing in flies, the van Hateren photoreceptor model and the NLN cascade LMC model, to fit the responses of locust photoreceptors. This will allow the Gabbiani laboratory to run large-scale simulations of higher order visual processing in the locust visual system using the outputs of these models as realistic neural inputs.

Two Important Factors of Stimuli: Light Intensity and Luminance Change Speed

1) Light Intensity

Locust photoreceptors change their membrane potential in response to changes in light intensity incident on the eye. When light intensity increases, photoreceptors depolarize; and when intensity decreases, they hyperpolarize. Shown below are photoreceptor responses to light pulses of two different intensities, with the stimulus pulses on the top and the recordings below. The data I worked with contained steps to three different lux values: 47.1, 77.5 and 90.4.

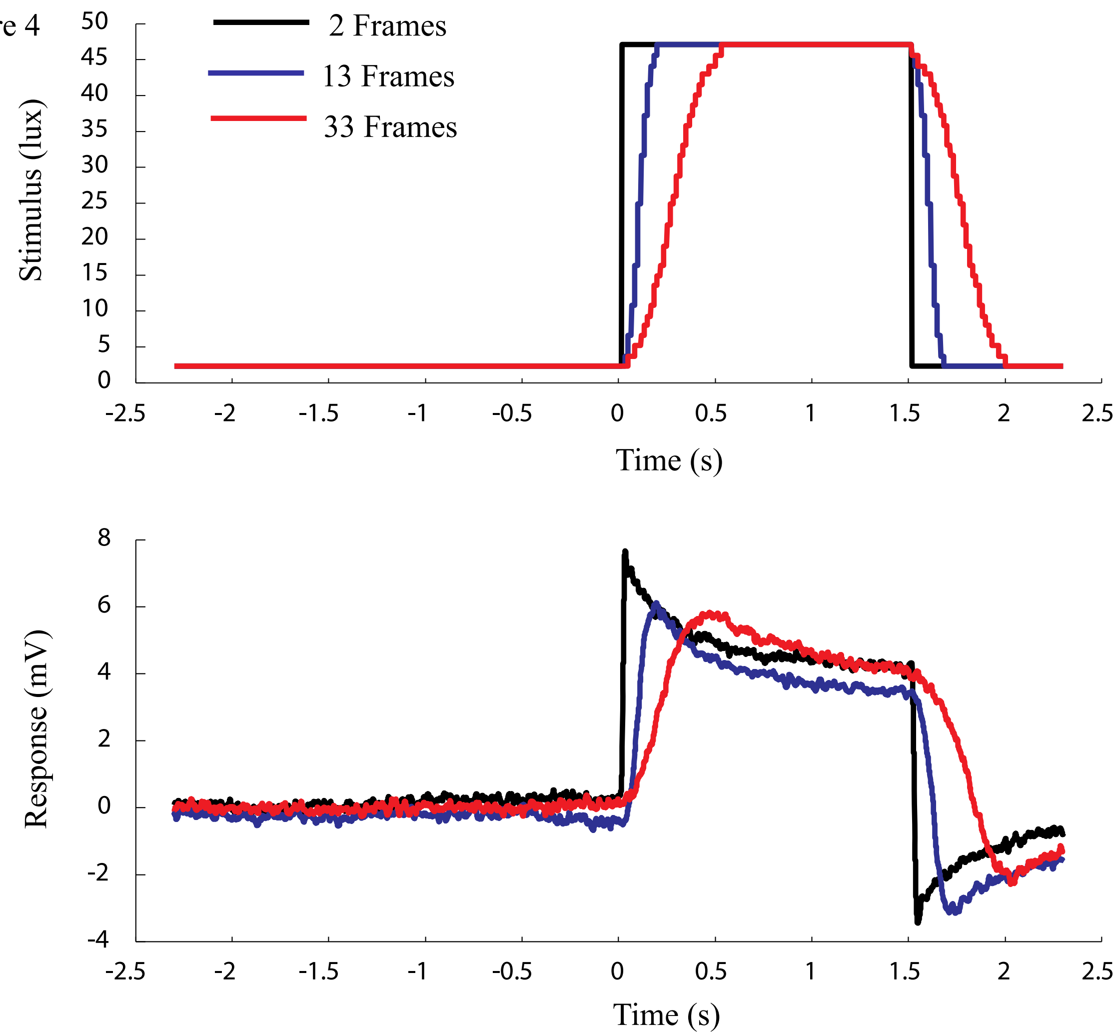
Figure 3



2) Luminance Change Speed

Stimulus pulses whose brightness increased and decreased at different speeds were presented while recording the membrane potential of a single photoreceptor. Stimulus traces are shown on the top, and recordings below. The response speed of photoreceptor tracks that of the stimulus change. One stimulus frame equals 1/60 sec. The Gabbiani Laboratory collected data after giving stimuli with six different transition speeds: 2, 4, 8, 13, 23, and 33 frames.

Figure 4



The Gabbiani Laboratory, in total, collected 18 different trial types of data: 3 different colors * 6 different transition time frames. I tried to find an optimized model that can simulate the responses of these 18 stimulus conditions.

Fitting the Model

1) Finding the Best Model Input

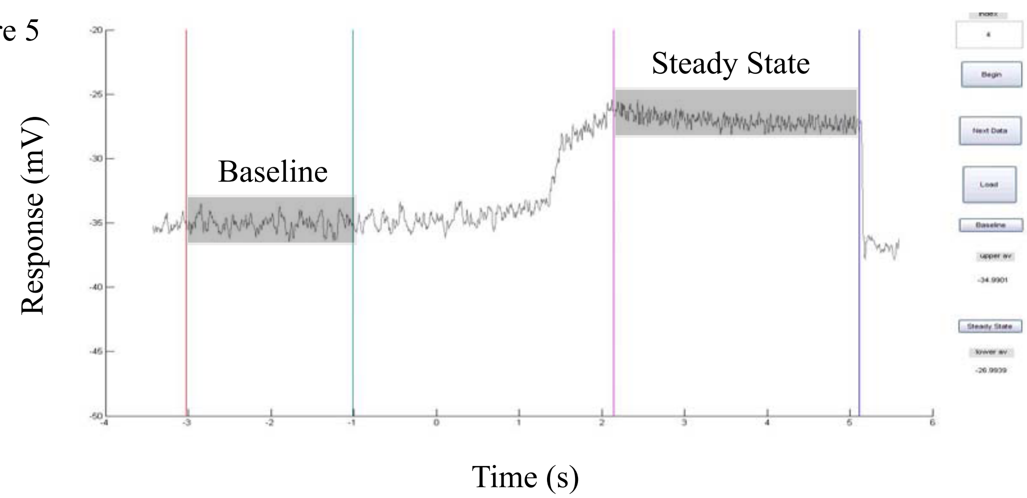
The van Hateren model was developed using data that was not calibrated with regards to the absolute light intensity. Thus, we needed to find the best linear transform to match the light intensities used during the photoreceptor recordings (I_{exp}) to the corresponding range of model input values (I_{model}). Since all we had to use were photoreceptor responses, we decided to fit the linear function

$$I_{model} = A \bullet I_{lux} + B \qquad (1)$$

by minimizing steady state response magnitudes of the van Hateren model and the data for 3 different stimulus intensities.

Methodology: I implemented GUI in MATLAB to measure ΔmV value of each photoreceptor's response.

Figure 5



The four vertical lines (red, green, purple, blue) are all movable.

To get an average value for the baseline, I calculate the mean value for all the data that are between red and green vertical lines; to get an average value for the steady state, I calculate the mean value for all the data that are between purple and blue vertical lines. The length of time windows of baseline and those of steady state were kept the same for each trial type. The range of time windows was from 1 to 3 seconds.

The "Begin" button initializes the GUI experiment. The "Next Data" button prepares to load the next data file (another stimulus presentation), and the "Load" button actually loads the data. The "Baseline" button calculates the mean value for the baseline. The "Steady State" button calculates the mean value for the steady state.

After going through the 60 data sets that the Gabbiani laboratory previously collected, I finally got the average values with different lux input values (31.7, 61.2 and 90.4). The final average ΔmV were 7.7948, 9.4390, and 10.1634 for each lux input value respectively.

For finding the parameters A and B in (1), I fit the model to the data using two methods of calculating error: least-square method and minimizing sum of the absolute value of the errors. Yet, the least-square method did not converge to a solution. Minimizing the sum of the absolute value of the errors, on the other hand, returned a reasonable solution. The fitted values of A and B were 16.1738 and 186.6813 respectively.

MATLAB code for calculating the sum of the absolute value of the errors for given values of A and B

```
function y = to_minimize_return(m,n)

data_delta_vm = [7.7948 9.4390 10.1634];
% mean delta vm from actual data set.

rint = zeros(1,8000);
% rint is input to the van Hateren Model
st_start = 2001; st_end = 5000; end_point = 8000;
lux = [31.7 61.2 90.4];
vm_tocompare = zeros(1,3);
rint(1:st_start-1) = 2.35*m+n; rint(st_end:end_point) = 2.35*m+n;

for i = 1:3
rint(st_start:st_end) = lux(i)*m+n;
[x_a x_b x_c x_d x_e x_f x_g x_h] = fly_phot9(rint);
%fly_phot9 executes the van Hateren model, and x_h is the final output of
%the model.
initial_vm = mean(x_h(1:1000));
upper_vm = mean(x_h(st_end-1000:st_end));
vm_tocompare(i) = upper_vm - initial_vm;
% storing delta vm of the model
end

to_minimize = zeros(1,3);
for i = 1:3
    error = vm_tocompare(i) - data_delta_vm(i);
    to_minimize(i) = abs(error);
end

num_to_minimize = sum(to_minimize);

y = num_to_minimize;
end
```

MATLAB code for searching minimum error

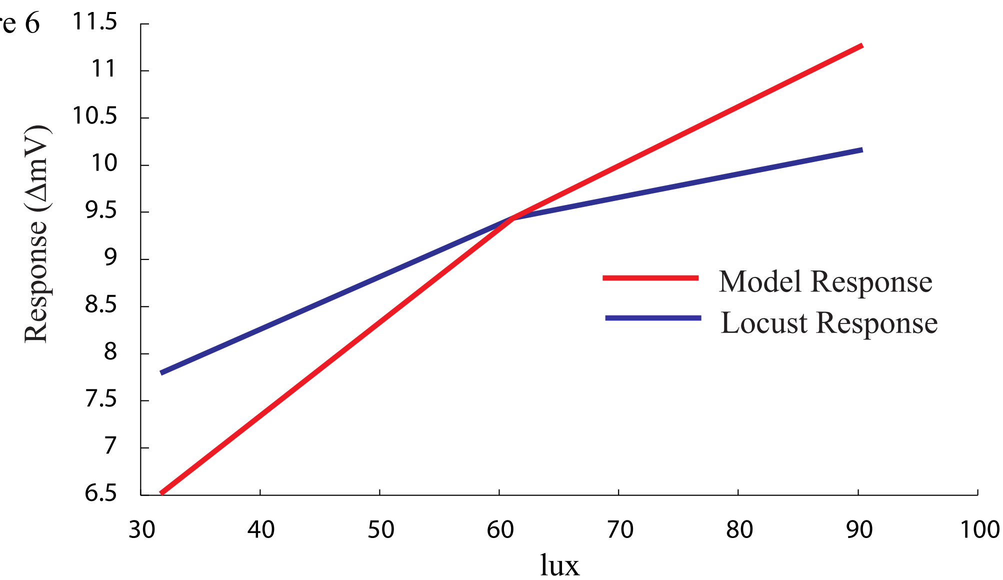
```
function [a b] = minsearch(x)
to_minimize = @(x)to_minimize_return(x(1),x(2));
```

```
[answer fval] = fminsearch(to_minimize, [16.1738 186.6813]);

a = answer(1);
b = answer(2);

end
```

Figure 6



For the three input values (31.7, 61.2 and 90.4) to observe, the differences between the model response and locust response were 1.2801, 0.0020, and 1.1096 respectively.

2) Finding the Best Model Parameters

After finding the best fit luminance transform while keeping the model's parameters constant, we needed to fit the 4 model parameters (tau1, tau2, k1, and k2). We first tried to freely vary all 6 parameter values (model parameters plus A, B), but that didn't converge to a solution, and the end parameter values didn't yield good fits to the data (Figure 7A, red line). We also tried to hold the parameters k1 and k2 constant, thinking that these govern responses over longer time courses than the stimuli were presented, but the solution arrived at was clearly not a good fit (Figure 7A, green line). After limiting the algorithms iterations to 300, but changing start conditions, we found a reasonable solution shown in blue. It took 7 hours on average to obtain each set of parameters in Figure 7A. These fits are shown for multiple stimulus values Figure 7B and speeds Figure 7C to show that they fit the range of experimental data well. All data analysis and stimulations were done in MATLAB.

Summary of procedures

minsearch(a,b,c,d,e,f) is a function implementing fminsearch to find minimum error; it finds optimal values of the model's parameters by executing the model with the six variables (A=a, B=b, tau1=c, tau2=d, k1=e, k2=f).

$$\text{Error} = \sum_{i=1}^{18} \sum_{t=0}^{1500} (\text{data}(i,t) - \text{model}(i,t))^2$$

i is an index designating the data set. There were 18 different data types with different lux input and transition time frames. Stimulus in the data last for 1500 ms, and t is a time index of stimulus. Error is calculated through sumreturn(a,b,c,d,e,f). How sumreturn(a,b,c,d,e,f) works is explained below.

Red: Free variation of all six parameters.

sumreturn(a,b,c,d,e,f) takes all six parameters. Before executing the van Hateren Model, I first transfer experimental stimulus values (lux) to the model input (parameters a and b are used in here). The input values are stored in matrix 'rint.' fly_phot10(rint,c,d,e,f) executes the van Hateren model with four variables (tau1=c, tau2=d, k1=e, k2=f).

Green: Variation of four parameters (A, B, tau1, tau2) and holding k1 and k2 constant

I used the function sumreturn(a,b,c,d,e,f) in here as well. Yet, I used a different implementation of the van Hateren model. fly_phot11(rint, c, d) executes the van Hateren model with four variables with constant value for k1 and k2 (tau1=c, tau2=d, k1 and k2 are held constant).

Blue: Limiting the number of iterations to 300.

The process of Blue is exactly the same with that of Red, but minsearch function (explained above) limits the number of iterations to 300.

Parameter values

A	8.743	0.0368	0.2001
B	340.8	0.8146	6.793
tau1	4.955e-6 ms	3.630e-6 ms	1.301e-6 ms
tau2	153.4 ms	300.5 ms	611.9 ms
k1	1.406	1.4061	3.521
k2	7.351	11.1719	8.447

Figure 7A

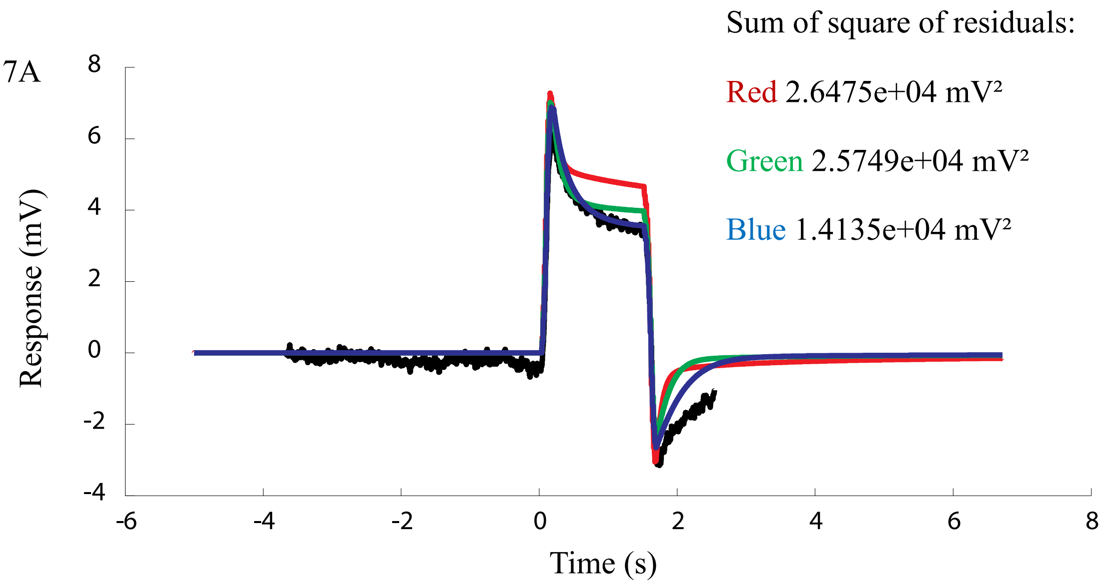


Figure 7B

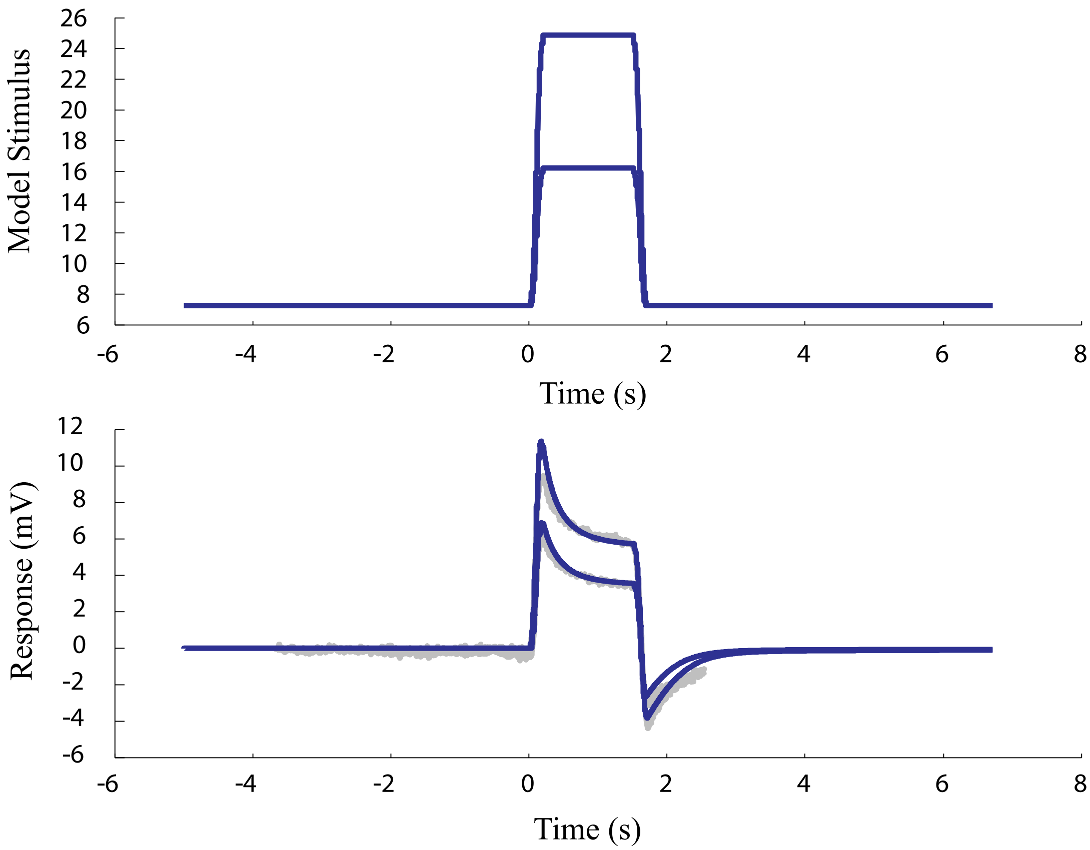
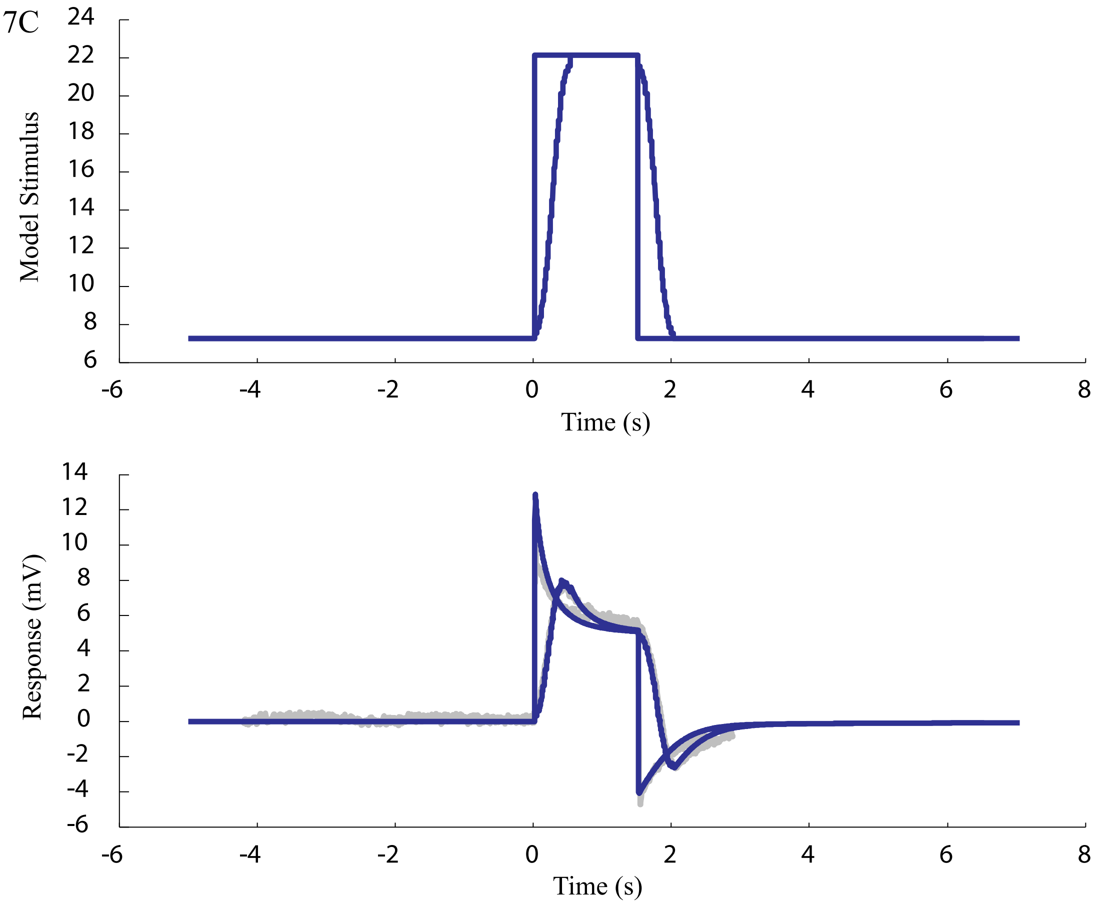
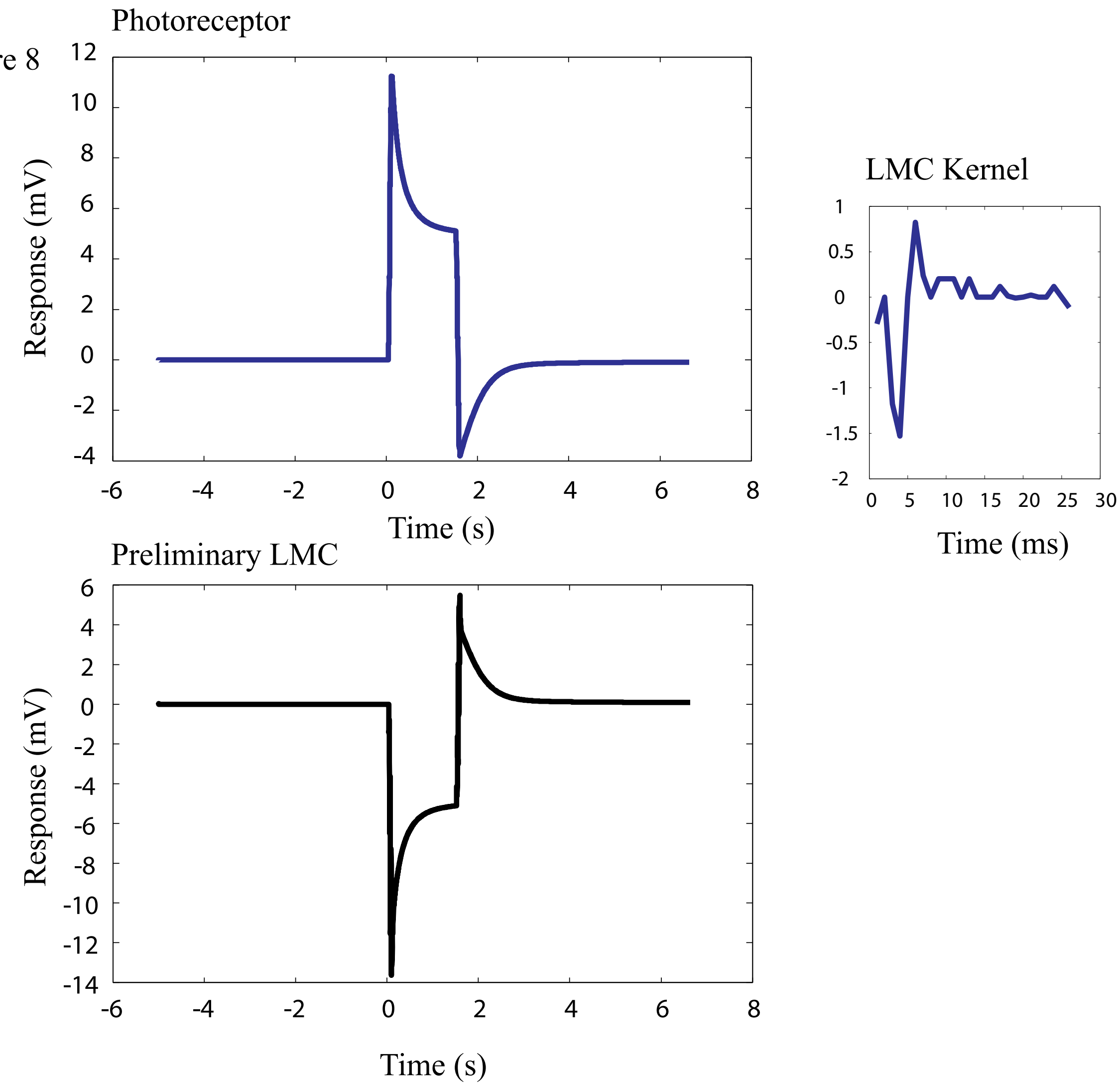


Figure 7C



We also started to implement the NLN model of LMCs, but were limited by time and lack of quality recordings against which to fit the model. We implemented the linear filter stage of the model, which produces responses that are inverted and emphasize transient portions of the response when compared to photoreceptor output. The response of LMC is produced by a convolution of LMC Kernel and a photoreceptor response. The LMC Kernel has a brief negative pulse so the convolution with a photoreceptor response produces a trace similar to an inverted photoreceptor trace yet emphasizing the fast changing portions. This is shown in Figure 8.

Figure 8



Conclusion

We were able to adapt a nonlinear model of fly photoreceptors to locust photoreceptor responses. We began implementing a model to describe the filtering properties of Large Monopolar Cells (LMCs). As of now, however, the Gabbiani Laboratory does not have enough data for analyzing the LMC model.

References

Juusola et al (1995) Nonlinear Models of the First Synapse in the Light-Adapted Fly Retina.

Journal of Neurophysiology 74: 2538-2547.

Mah et al (2008) Implementation of an elaborated neuromorephic model of a biological photoreceptor. Biological Cybernetics 98: 357-369.

van Hateren and Snippe (2001) Information theoretical evaluation of parametric models of gain control in blowfly photoreceptor cells. Vision Research 41: 1851-1865.

Acknowledgements

This work was partially supported by NSF REU Grant DMS-0755294. I would like to thank Dr. Fabrizio Gabbiani and Peter Jones for their mentoring. Also, thanks to the entire members of Rice/TMC/UH Computational Neuroscience REU 2008.