Mid Year Project Report: Statistical models of visual neurons

Anna Sotnikova

asotniko@math.umd.edu

Project Advisor: Prof. Daniel A. Butts

dab@umd.edu

Department of Biology

Abstract

Studying visual neurons may explain how a human brain performs its sophisticated and efficient image analysis. The main question is what mathematical model can represent the processes occurring in the brain. Nowadays there have been proposed a variety of models. The measure of accuracy of such models is the difference between the predicted output (neuron's firing rate) and the real experimental data. During this semester two models were implemented, which focused on estimating linear filters by moment analysis of the data.

December, 16 2016

1 Introduction

Human brain has approximately 86 billions of neurons, and all these neurons process information non-linearly. Here is a brief introduction to the visual system of a neuron illustrated in the Fig. 1. A sensory stimulus is cast into a human eye. Stimulus attributes have to match a sensor, which is light for a visual system. Stimulus is a vector denoted by the capital letter **S**; it has x,y-coordinates, which change in time. However, we pick a point x0,y0 that the neuron is most sensitive to and reduce the stimulus to just a function of time for the rest of the report. The result of stimulus action is neural activity. The neural activity is represented by spikes. A spike is a change in the response of a neuron. The activity level is reflected by the number of spikes at a certain moment of time - spike frequency vector and denoted by the letter **n**.

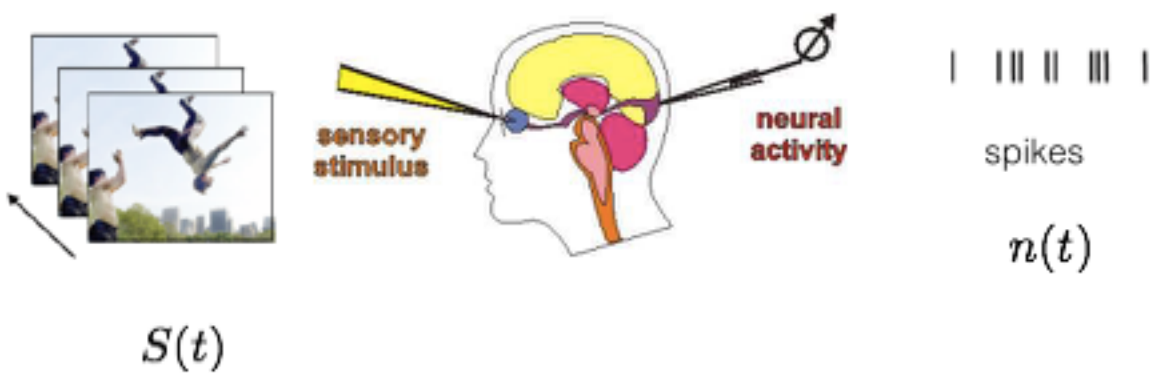

Fig. 1 Visual system of a neuron.
S(t), a stimulus value at moment t, **S** is a stimulus vector dependent on time. Typically, stimulus is displayed between 16-120 HZ (meaning 65.5 ms down to 8.3 ms). n(t) is a number of spikes at moment t, **n** is a vector of number of spikes occurred during an experiment.
picture source: http://www.pc.rhul.ac.uk/staff/j.zanker/ps1061/l2/ps1061_2.htm

Many researches are trying to find what is the mathematical description of the processes, which happen in the brain during an “image processing.” Basically, we want to predict the reaction of neurons on some input (stimulus). The prediction can be viewed as probability of a neuron to have a spike at a certain moment of time. This prediction for the whole experiment is called a firing rate and the vector denoted by the letter **r**. The main goal is to reproduce the real response of neurons. What kind of mathematical function can describe relation between the stimulus and the firing rate? It is unlikely that there exist a simple function that does this. We therefore must split this function into a set of other functions with optimally distributed properties (for instance, linear filter + non-linear estimator function) and this choice of functions might be a model. If we make our model too complicated, it might indeed cover all the

observations, but it might require too complicated parameters fitting procedure. Therefore we must consider maximally simple models.
For this semester's part of the project, the main goal was to predict firing rates of neurons based on the Linear-Nonlinear-Poisson (LNP) model. In this model, there are two major steps: estimation of linear filters that are applied to the stimulus signal and estimation of a non-linear function that converts the filtered stimulus into the firing rate. The linear filters were estimated by the Spike Triggered Average (STA) model and by the Spike Triggered Covariance (STC) model. I first practiced linear filter estimation using a synthetic data and obtained a good agreement with the results of the previous paper [2] that used the same data. Next, I implemented the full LNP model for both types of linear filters (STA and STC) and performed validation procedure using a real experimental measurement of the lateral geniculate nucleus of three cats [3].

2 Project Objective

For the first part of the project, I used Linear-Nonlinear-Poisson model for estimating firing rate and two moment-based statistical models for finding linear filters.

★ Linear-Nonlinear-Poisson model

- Spike Triggered Average (STA)
- Spike Triggered Covariance (STC)

Those two models were applied for two data sets.

- Synthetic data set (RGC data)
- Real data set (LGN data)

I start with introducing the typical raw data and my manipulations with it to bring it to a more convenient mathematical format. This is an important part because this data format is used throughout this report. In particular, all the formulas and variables in this report use the format of the data presented below and not the raw data. The two used data sets (RGC and LGN data) are described in Section 3. Formulation of the Linear-Nonlinear-Poisson model with STA and STC filters is given in Section 4. Section 5 presents the results of my analysis of the synthetic data set and their comparison to the paper [2]. The results of my analysis of the real data set are provided in Section 6. Section 7 describes the hardware and software used for the project. Section 8 provides the two project timelines : the updated one (12/08/16) and the initial one (10/01/16).

3 Data sets description

Let me start from describing data sets with which I was working. In the Section 3.1, I present the raw synthetic data and define new variables that are used in this project to analyze it using the models summarized in Section 2. In Section 3.2, I present the real data set along with its reformatted variables. I will also comment on our target goals for both data sets.

3.1 Synthetic data set (RGC data)

Synthetic data set of retinal ganglion cells (RGC), which was used in Dr. Daniel Butts' paper of 2013 [2], has the following variables :

1. Stimulus (Fig. 2).

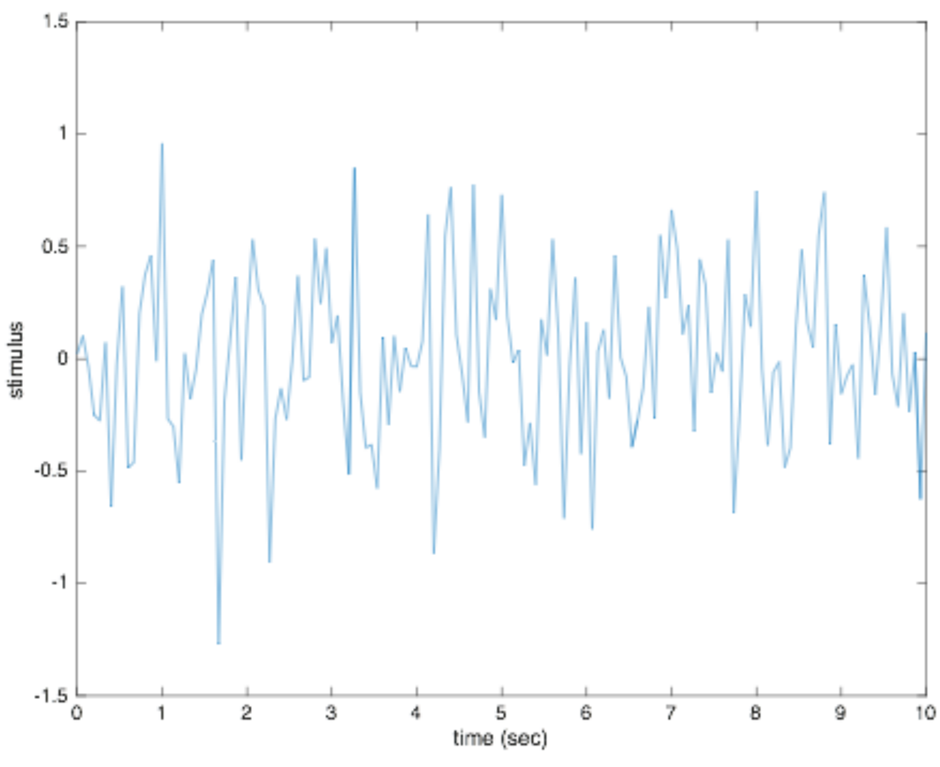

Fig. 2 First 10 seconds of the stimulus for the synthetic data.

This stimulus I reformatted into a matrix of stimuli by the following formula:

(1) $$\mathbf{s}[t,] = [S(t), S(t-dt), ..., S(t-P*dt)]$$

where vector **s**[t,] is a stimuli vector at a certain moment t, S(t) is a stimulus element at a moment t, dt (time step) is a time of a stimulus update, and P is a stimuli length.

Every stimuli is a part of a stimulus that starts at time t and goes back in time up to P*time step. By formula (1) I got a matrix, where each row corresponds to a stimuli vector at moment t.

2. Spike times in units of seconds.

The given vector of spike times contains only times at which a spike occurred. If there were "n" spikes simultaneously, then the vector has "n" repetitions of the same time value. Spike times vector has it's own length, which is not related to the stimulus or stimuli length, and depends only on the number of spikes happened during the experiment time.
I reformatted this vector into a vector of the stimulus length, where every element corresponds to the number of spikes occurred at the time of stimulus update. In other words, I have stimulus values for the times from 0 to the end of experiment and now I have corresponding to these times numbers of spikes. I denoted the vector as **n**. The first 10 seconds of this new vector are shown on the Fig. 3

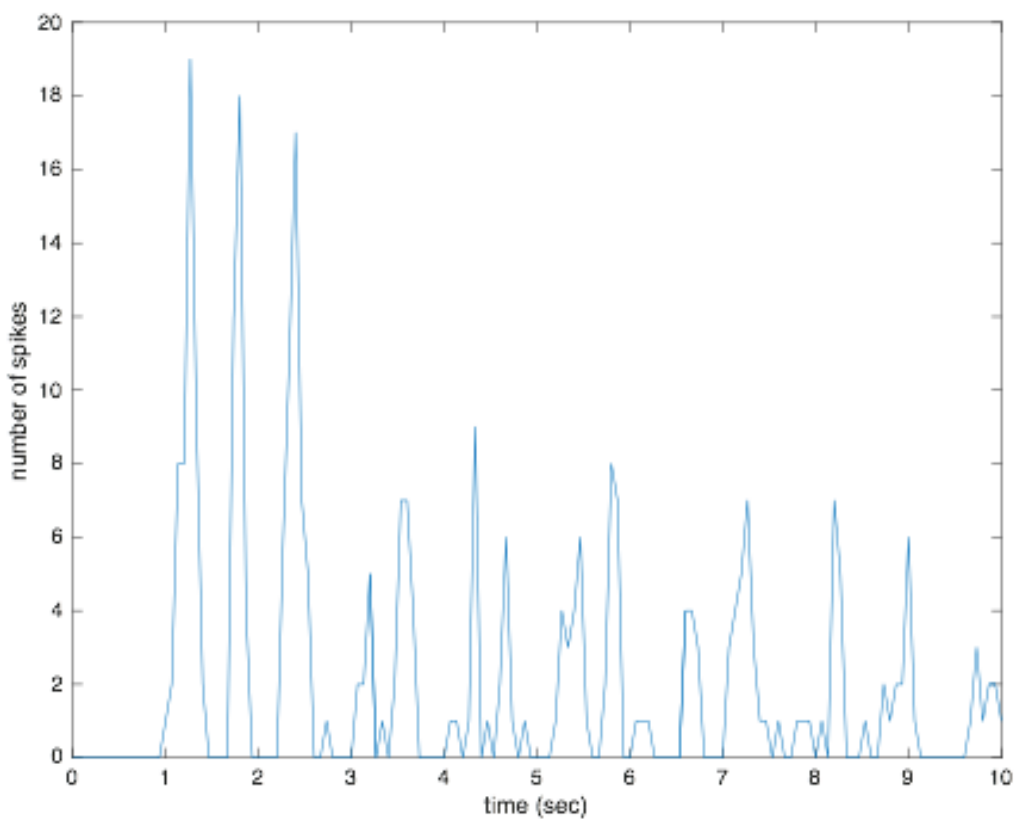

Fig. 3 First 10 seconds of the number of spikes for the synthetic data.

3. Time step(dt) is a time of stimulus update.

This data set was used in order to validate the code for STA and STC models. It allowed me to check that the filters obtained by me matched the results from Dr. Daniel Butts' paper of 2013 [2]. I did not estimate the firing rate with LNP model for the synthetic data set since no independent repeated data on neuron's response was available within this particular data set. A full comparison of predicted firing rate with the experimental observations was done using the real data, described below.

3.2 Real data set (LGN data)

Real data set ,which was used in Dr. Daniel Butts' paper of 2011 [3], of lateral geniculate nucleus of 3 cats has the following variables :

1. Stimulus for an experimental duration of 120 seconds and corresponding spike times.
2. Another stimulus of 10 seconds length, which was cast into cats' eyes 64 times, and the corresponding spike times. (Further I am referring to this stimulus as to the repeated stimulus.)
3. Time step(dt) is the same for both stimulus.

I applied formula (1) to both stimulus. For spike times vector transformations I followed the procedure, which was described in part 2 of Section 3.1. The real data set was used in order to get the spike rates using both models for filters estimation (STA and STC). The obtained spike rates were compared with the real averaged spike rates and R-squared tests were performed. This allowed me to check how close my predictions were to the real measurements. This type of the tests is called cross-validation.

Cross validation procedure consists of first extracting the model for predicting the neuron's firing rate using the stimulus and spike times data for an experimental duration of 120 second. In the second step, we applying the extracted model to an independent piece of data, the stimulus part of it, to predict the firing rate and compare it directly to the observed rate, which is obtained by averaging the spike count over the trials. The R -squared test was used in order to get a quantitative comparison of how close my predicted spike rate are to the measured one. For a perfect match, the R-squared test yield a value of unity by definition (see Eq. 2). The deviations from a unity indicate a discrepancy.

The formula for R-squared is the following:

(2)
$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i y_i^2}$$

where y_i - measured data, and $\hat{y}_i$ - estimated data.

The good R-squared rate is when the ratio in (2) approaching zero. In other words, the closer R-squared value to 1, the better estimated spike rate matches to the measured spike rate.

4 The Linear-Nonlinear-Poisson model

In this section I present the description of Linear-Nonlinear-poisson model. This is a basic model for estimating neuron's firing rate. An important step in LNP model implementation is a filter estimation, which can be done by using one of two moment-based statistical models: either STA or STC model. Procedures of filters' estimations are provided in Sections 4.1.1 and 4.1.2 respectively.

The linear models project the stimulus onto a filter, then map this projection nonlinearly into a firing rate. The firing rate at a certain moment of time gives us the probability of the neuron spiking at this moment of time [7].

(3) $$r(t) = F(\mathbf{k} \cdot \mathbf{s}(t))$$

where F is non-linear function, **k** is a linear filter, and **s**(t) is a stimuli vector at moment t, which represents a row of the matrix of stimuli from formula (1).

The algorithm, which I used for finding the firing rate in the formula (3), may be detailed as follows:

1. Obtain a matrix of stimuli from the given stimulus by the formula (1). Notation for a stimuli vector is **s**(t).
2. Estimate the linear filter **k** by one of the moment-based statistical models (STA or STC).
3. Project all stimuli onto a filter and get the generator signal. The value of the generator signal at a certain moment t is given by the formula:

(4) $$g(t) = \mathbf{k} \cdot \mathbf{s}(t)$$

4. Estimate non-linear function F.
5. Apply formula (3) for finding the firing rate.

4.1 Moment-based statistical models for filter estimation

4.1.1 STA Model

In the Spike Triggered Average(STA) model the output is the linear filter, which does not depend on time. The STA is represented by the formula (5)

(5) $$\boldsymbol{k}_{sta} = \frac{1}{N}\sum_{j=1}^{M} n(t_j)(\hat{\boldsymbol{s}}[t_j,] - \bar{\boldsymbol{s}})$$

where N is the total number of spikes per experiment, M is the total number of stimuli per experiment, n(t) is the number of spikes at time t.

$\hat{s}[t_j,]$ is a stimuli vector at time t and

(6) $$\bar{\boldsymbol{s}} = \frac{1}{M}\sum_{j=1}^{M} \hat{\boldsymbol{s}}[t_j,]$$

which is the average stimuli with M equals to the total number of stimuli per experiment. The average stimuli was obtained by averaging stimuli, which precede the spike [6].

STA estimates the linear stage, which corresponds to one filter, and has the length of a stimuli [1]. However, this model very often does not fit fully because neural responses are mostly non-linear. The most general physical interpretation of STA is the receptive field of a neuron, which defines the preferred stimulus for the neuron [1]. In other words, the receptive field is the location in space where the presence of visual stimulus can produce a spike.

4.1.2 STC Model

The STA model analyzes changes in the spike-triggered stimulus's mean for estimating linear part of linear non-linear model. However, it corresponds only to the single direction of the stimulus, and also it means that we are working with one dimensional problem. The Spike Triggered Covariance(STC) is used when we need to predict a probability of a spike along more than one direction. This model gives us a variance-covariance matrix. The stand-alone eigenvalues of this matrix reveal us possible filters as the corresponding eigenvectors. Notice that having more than two filters makes the problem of fitting the model complicated, because a very large number of stimulus-spikes data points is required for accurate extraction of the multi-dimensional histograms. For a given data set, we are able to work at most with two filters. Consequently, in STC model case we are working with the two dimensional problem.

STC matrix is represented by the formula:

(7) $$\boldsymbol{\phi}_{stc} = \boldsymbol{\phi}_{trig.stc} - \boldsymbol{\phi}_{untrig.stc}$$

where

(8) $$\boldsymbol{\phi}_{trig.stc} = \frac{1}{N-1}\sum_{j=1}^{M} n(t)(\hat{\boldsymbol{s}}[t_j,] - \boldsymbol{k}_{sta})(\hat{\boldsymbol{s}}[t_j,] - \boldsymbol{k}_{sta})^T$$

and

(9) $$\boldsymbol{\phi}_{untrig.stc} = \frac{1}{M-1}\sum_{j=1}^{M}(\hat{\boldsymbol{s}}[t_j,])(\hat{\boldsymbol{s}}[t_j,])^T$$

where N is the total number of spikes, n(t) is the number of spikes at time t, M is the total number of stimuli per experiment, STA filter is defined by the formula (5).

STC gives a quadratic model for neural responses and as well as STA cannot fit data completely. That is why it is often used as starting point for estimation of another model. Geometrical idea of STC is that we are looking for such directions along which the variance of spike-triggered stimulus differs from the raw stimulus. STC model determines excitatory or inhibitory properties of neurons' responses. Excitatory property is defined by the increase in variance, and inhibitory property is defined by the decrease in variance [5].

STC matrix gives us at most two possible filters. Consequently, I adjusted the formula (3) for the firing rate for the 2D case:

(10) $$r(t) = F(\mathbf{k_1} \cdot \mathbf{s}(t), \mathbf{k_2} \cdot \mathbf{s}(t))$$

where instead of one filter as it was in formula (3), we have two filters, which can be extracted from STC matrix defined in formula (7).

5 Results of analysis of the synthetic data set

For the synthetic data set I only estimated filters by using STA and STC models. The main purpose of using this data set was to validate my code and algorithms for these two models. The detailed explanation for the synthetic data set is given in the Section 3.

5.1 STA filter for the synthetic data set

My first step for the project was to write a code for STA model, obtain the filter using formula (5) and compare the filter with results from Dr. Daniel Butts' paper of 2013 [2]. I took stimuli length P equals to 120 time steps.

As a result, I got

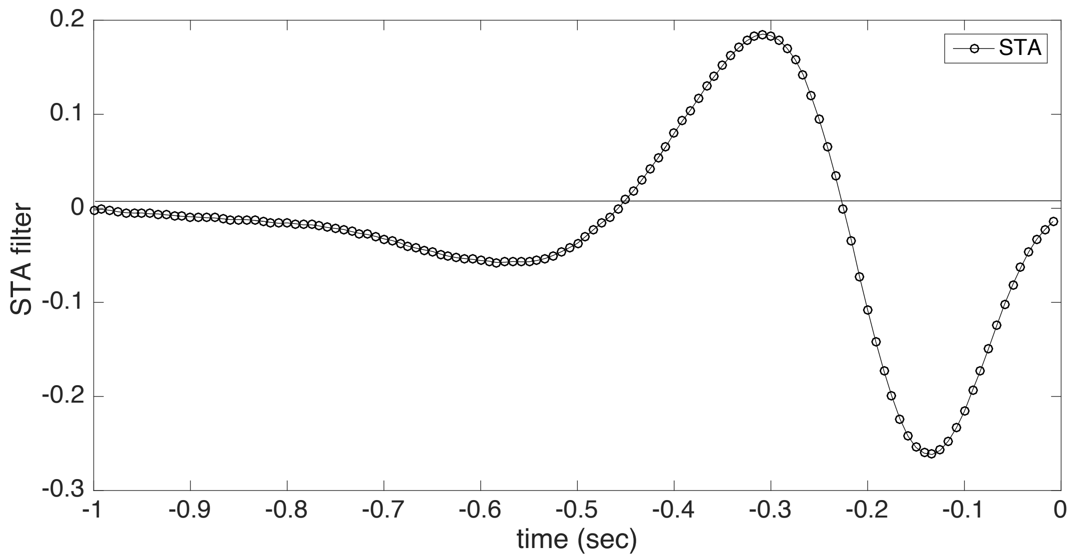

Fig. 4 STA filter(which is **k** in my notations) extracted for a stimuli of the length of 120 time steps, which corresponds to 1 second.

Fig. 4 was visually compared with Fig. 7. As we can see the result coincides. That is why I assume that the model was implemented correctly.

5.2 STC filter for the synthetic data set

I took stimuli length P equals to 120 time steps. For obtaining variance-covariance matrix formula (7) was used. The next step was to find it's eigenvalues. Since the most eigenvalues have approximately the same values, then only the outliers will give me the desired filters (Fig. 5). The respective two eigenvectors are the possible choice for filters **k**. Only one filter was shown in the paper of 2013 [2], which I am using for the validation of my results. Consequently, I used the respective filter in my Fig. 6, which I visually compared with Fig. 7.

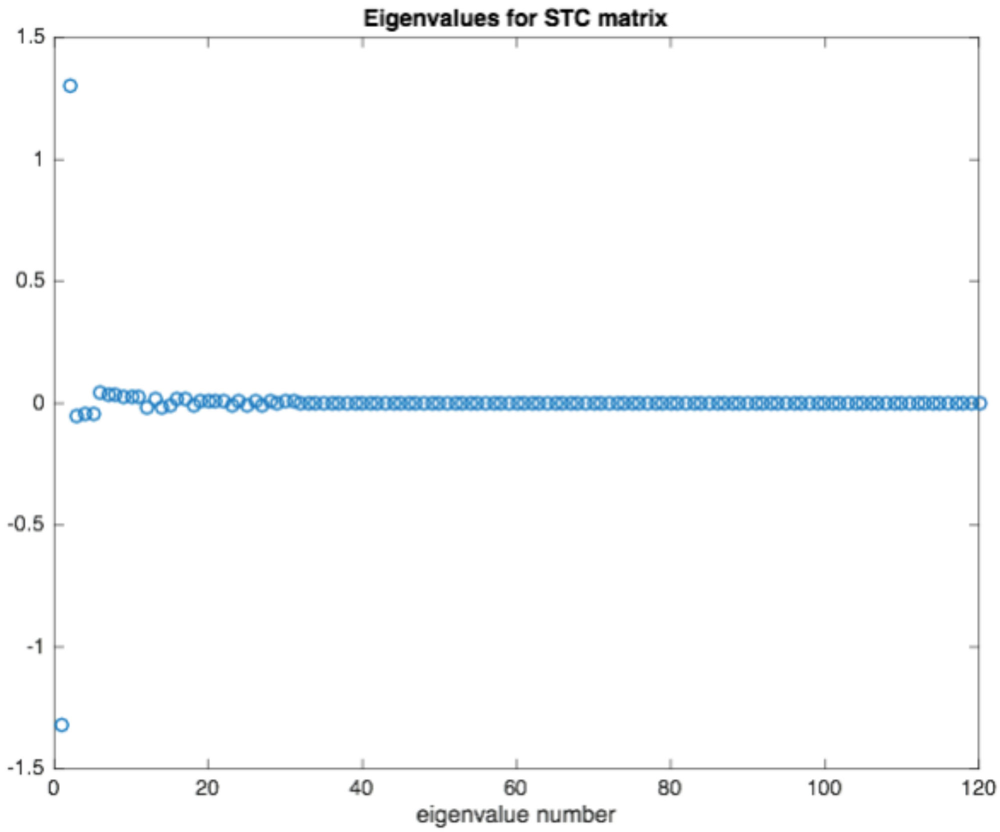

Fig. 5 STC matrix eigenvalues. Note the presence of two special eigenvalues (the lowest and the highest), corresponding eigenvectors are filters for the data set.

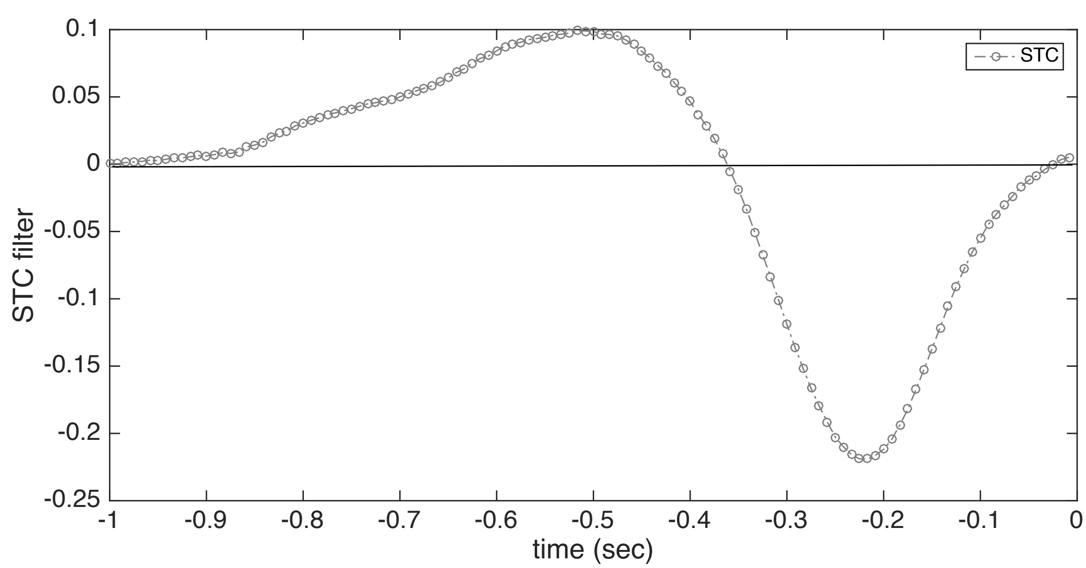

Fig. 6 One of STC filters (this is **k** in my notations), which was used for comparison with Fig. 7. It was extracted for a stimuli of the length of 120 time steps, which corresponds to 1 second of data.

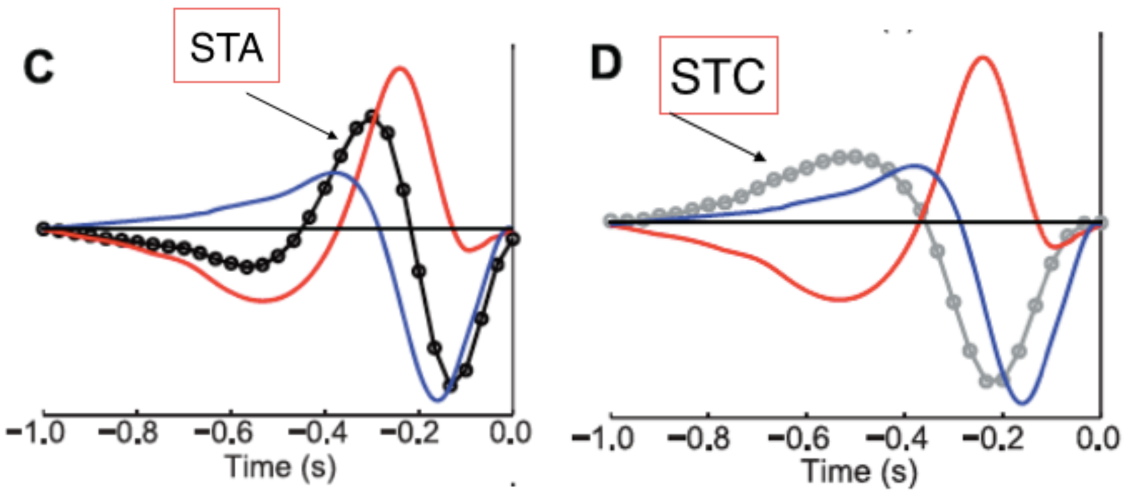

Fig. 7 STA and STC filters copied from the reference [2]. The unmarked curves(red and blue) are irrelevant for the purposes of the project. The black curve represents STA filter for the synthetic data set. The grey curve represents one of STC filters for the synthetic data set used in the reference [2].

6 Results of the analysis of the real data set

For the real data set I found firing rates using two models for filters estimations (STA and STC). I cross-validated my estimated results by comparing them with the measured values. I validated my results by obtaining R-squared values and by comparing them with expectations for both models' performances.

6.1 Firing rate obtained using STA filter for the real data set

I used the following scheme in order obtain all mentioned above checks.

- Take stimulus of 120 seconds duration and the corresponding spike times.
    1. Estimate a single linear filter **k** using STA model, formula (5).
    2. Estimate non-linear function F.
- Take the repeated stimulus, where the 10 seconds stimulus was repeated 64 times, and the corresponding spike times.
    3. Apply **k** & F from 1st and 2nd steps to the stimulus and obtain the firing rate by the formula (3).
    4. Calculate the average spike rate by averaging the repeated spike times.
    5. Compare the prediction with actual measurements, calculate R-squared value.

6.1.1 Step 1 : estimate STA filter

Here I defined stimuli length P equal to 15 time steps. The STA filter was obtained by the formula (5) from the stimulus and the respective spike times for the experiment duration of 120 seconds and represented on Fig. 8.

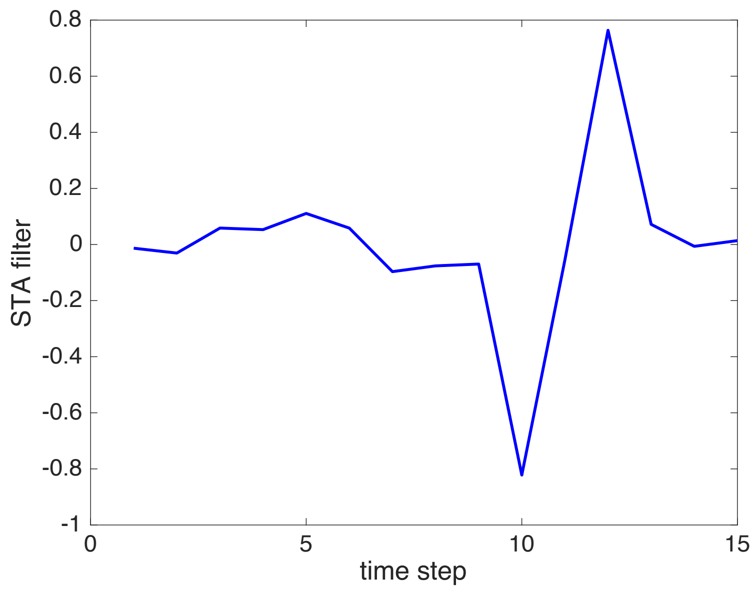

Fig. 8 STA filter, which is **k** in my notations, extracted for 120 seconds stimulus from the real data set. The filter length is 15 time steps, which corresponds to 0.1251 seconds.

The found STA filter **k** was used at the step three of the scheme described in the section 6.1.

6.1.2 Step 2 : estimate non-linearity

In order to estimate nonlinearity, I find the generator signal by the formula (4). Then use histogram method for non-linearity estimation. The histogram of the generator signal is on the Fig. 9.

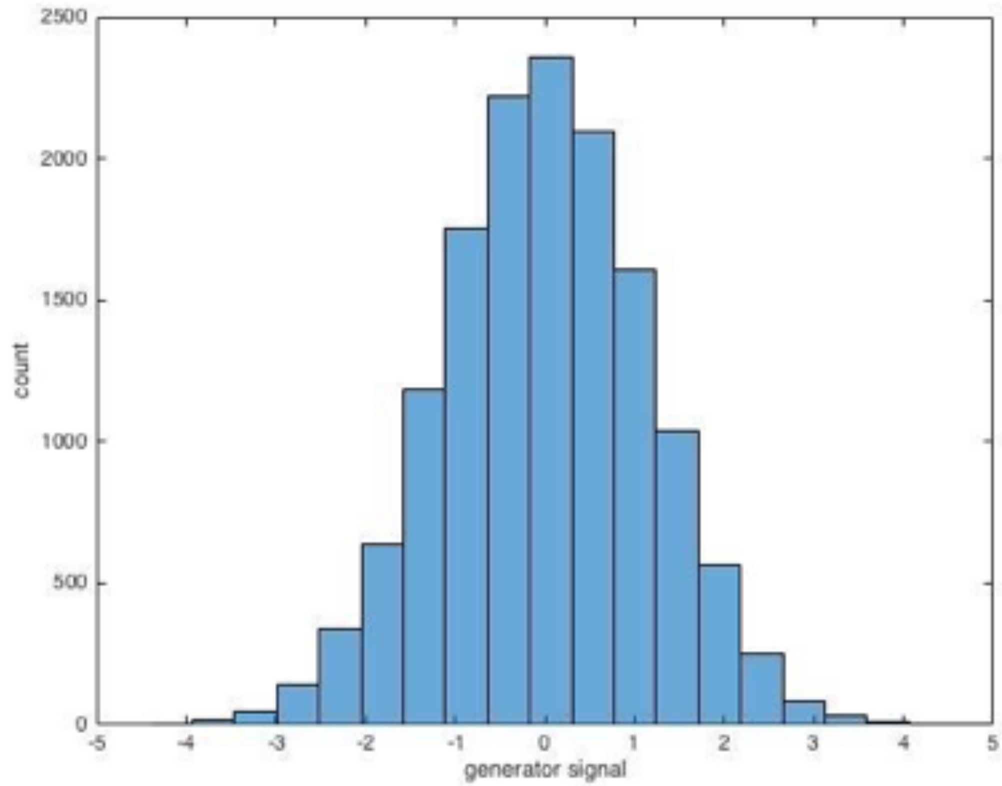

Fig. 9 The histogram of the generator signal values.

For non-linearity estimation I calculated the average number of spikes per bin. To do this, I looked up the values of times that corresponded to the stimulus that fits into a particular bin, and, by looking at the spikes data calculate the average number of spikes corresponding to every bin. This, by definition, gives as the non-linearity function F evaluated for discrete values of the argument, given by the bins in Fig. 9. This is the essence of the so-called “histogram method”.

The histogram method gives the non-linear function F that maps generator signal onto a spike rate, which is shown on the Fig. 10. This estimated non-linearity F was used at the step three described in the Section 6.1.

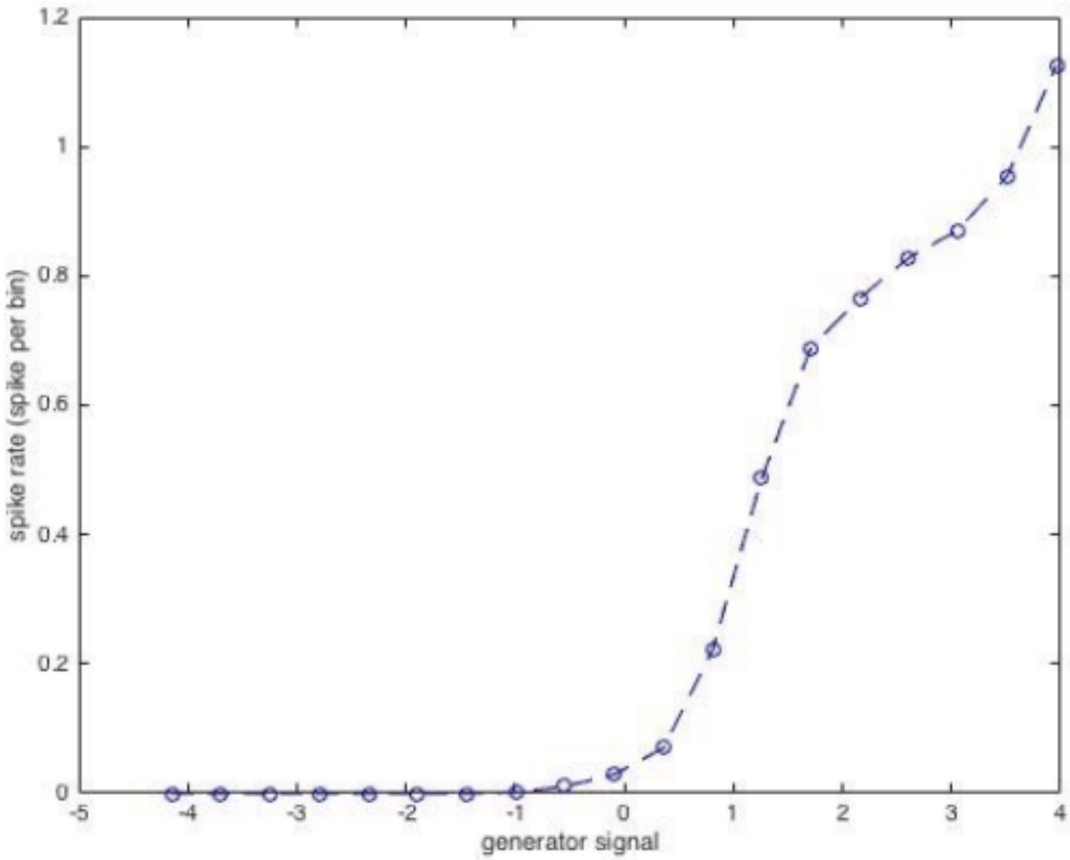

Fig. 10 Non-linearity F, which maps generator signal values onto spike rate.

6.1.3 Steps 3-5: cross-validation

The length of stimuli for the repeated stimulus is the same as for stimuli used in Sections 5.1.1 and 5.1.2, i.e. 15 time steps. Here I applied the filter found at the first step to the repeated stimulus; this gives me the generator signal for the repeated stimulus. Then I apply non-linearity F found at the second step to this generator signal. In order to apply the non-linearity F to the new generator signal, I projected stimuli at a time t onto the linear filter to obtain the argument of F at a time t. I then applied F to this argument and obtained the spike rate. This procedure estimated the spike rate for the repeated stimulus. Also in order to compare my found spike rate I need to average the number of spikes vector for the repeated stimulus (averaging across 64 repetitions). As a result, we can see that estimated spike rate matches to the measured very well (Fig. 11) since the R-squared test defined by formula (2) is 0.7696, which is considered to be a good value because R-squared value equal 1 means the absolute match. My result goes along with expectations for LNP model using STA filter. I should get a good performance for the real data in case of 1 dimensional problem.

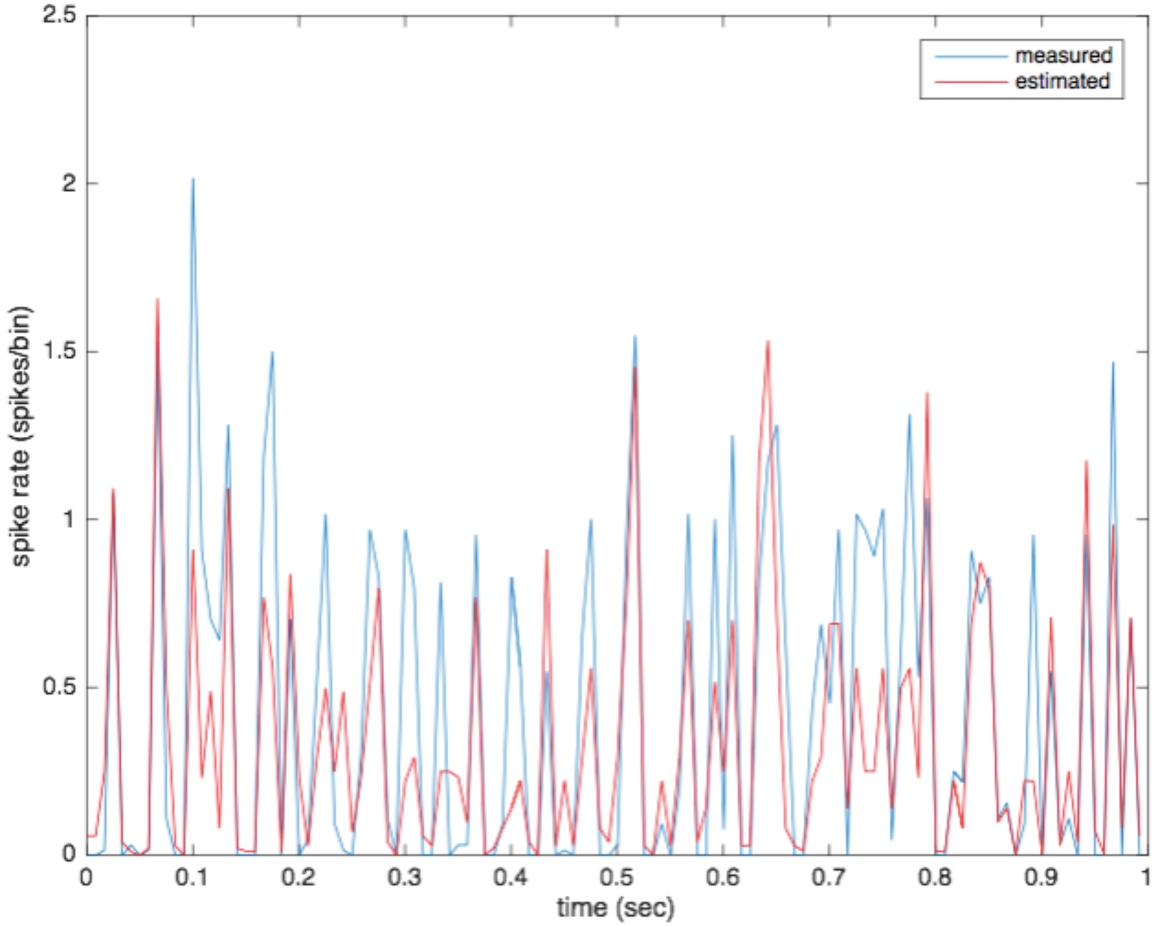

Fig. 11 Cross-validation for spike rates. Red is estimated spike rate, blue line is measured spike rate.

6.2 Firing rate obtained using STC filters for the real data set

Since STC model is two dimensional, it makes the problem of finding the firing rate 2 dimensional. I used the formula (10) for the firing rate.
There are two options for different types of data in choosing those two filters. The first option is to work with the given time step at higher resolution, then STC variance-covariance matrix has two outliers, which means we can use two respective eigenvectors as two filters. The second option is to run at the lower resolution, then STC matrix gives one obvious outlier, which is used as one filter, as the second filter we can choose either the eigenvector corresponding to the largest eigenvalue of STC matrix or STA filter. I have tried both options, the second one was presented in class. However, it gave me the lower R-squared value defined in formula (2) : 0.2424 versus 0.5374. The second option was useful to explore; it can be used for fitting another data in other algorithms, which I am going to study. For my particular case with the real data (LGN) set and LNP model, it performed worse. In general, the expectations for LNP model with

STC filter were not as high as for 1D case with STA filter, but 0.2424 rate is a bit below the expected. Here I am presenting only the first option as the most successful one.

I used the following scheme in order obtain all mentioned above checks.

- Take stimulus of 120 seconds duration and the corresponding spike times.
  6. Estimate two filters **k** using STC model, formula (7).
  7. Estimate non-linear function F.
- Take the repeated stimulus, where the 10 seconds stimulus was repeated 64 times, and the corresponding spike times.
  8. Apply **k**'s & F from 1st and 2nd steps to the stimulus and obtain the firing rate by formula (10).
  9. Calculate the average spike rate by averaging the repeated spike times.
  10. Compare the prediction with actual measurements, calculate R-squared value.

6.2.1 Step 1 : estimate STC filters

Here I defined stimuli length P equals to 15 time steps. The STC matrix was obtained by the formula (7), and it's eigenvalues are on Fig. 12

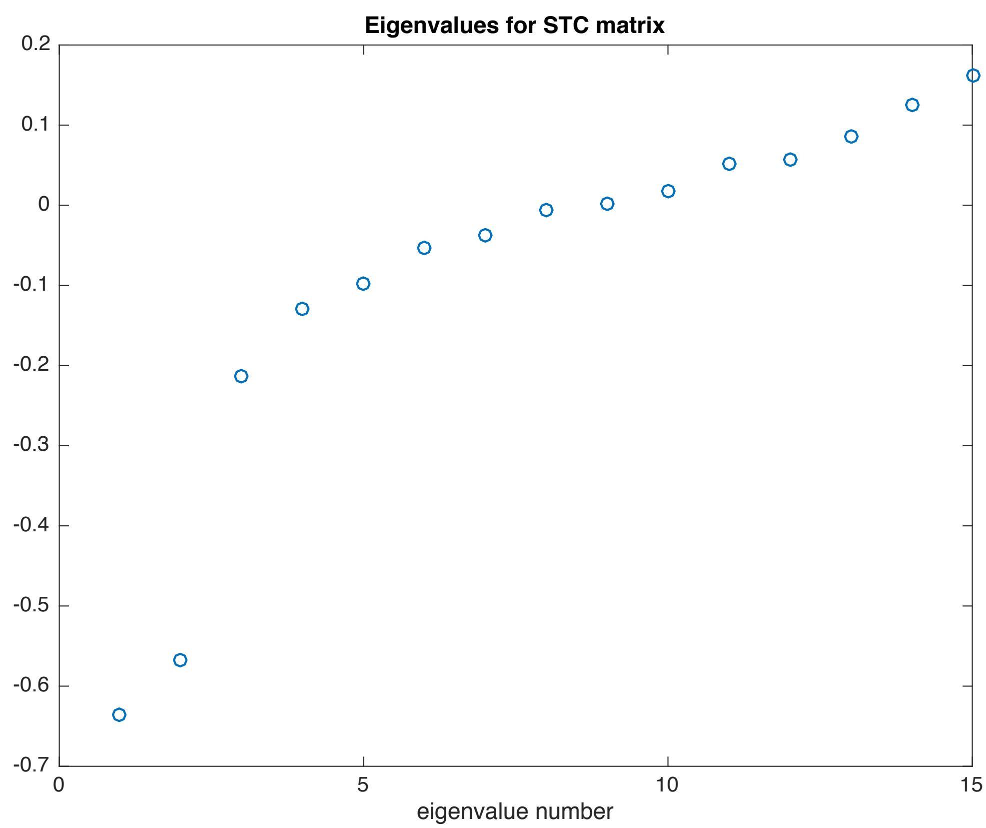

Fig. 12 STC matrix eigenvalues. Note two special eigenvalues

We can observe two outliers at the lower left corner. The respective eigenvectors are the desired filters.

6.2.2 Step 2 : estimate non-linearity

Ideologically, the procedure is the same as in the Section 6.1.2, but now I have two dimensional problem. Consequently, I find two generator signals by the formula (4). Then use 2D histogram method for non-linearity estimation. The histogram of the generator signals is on the Fig. 13.

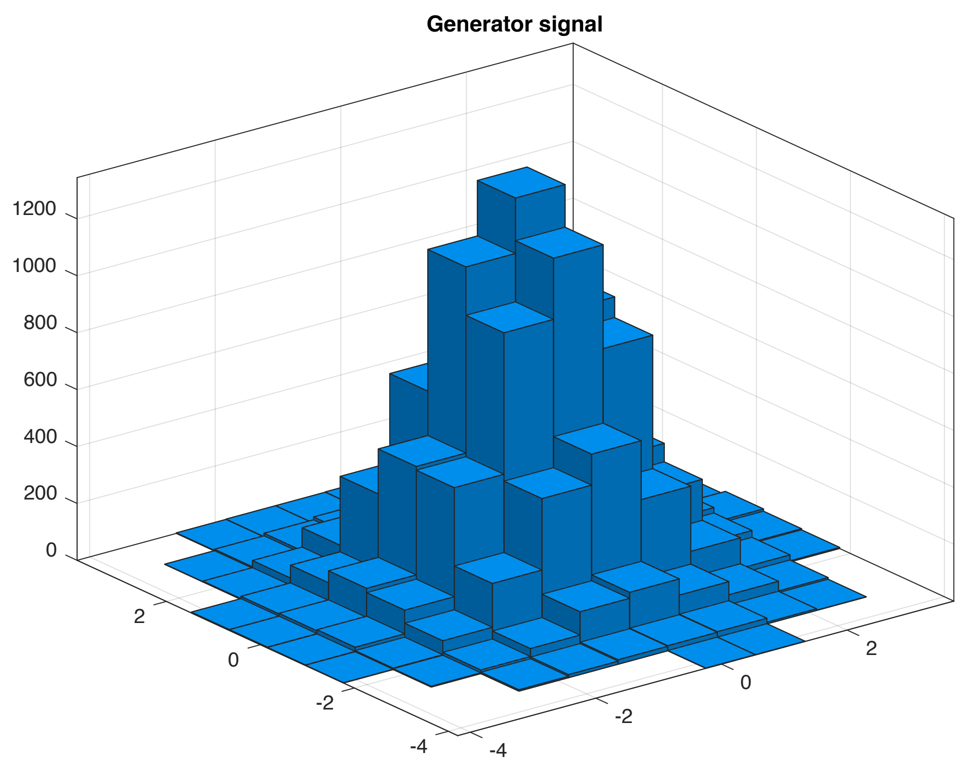

Fig. 13 2D generator signal for the two STC filters (See Eq. 10).

Here we can see the lack of the data. If in 1D STA case I used 20 bins, here I have 100 bin (10 per dimension), but the stimulus length is still the same 120 seconds. STC model performs better on larger vectors.
For non-linearity estimation I also followed the procedure explained in Section 6.1.2 i.e. I calculated the average number of spikes per bin. This gives 2D non-linear function F that maps generator signals onto a spike rate, which is shown on the Fig. 14 (main 2D picture) and the Fig. 15 (1D perspective)

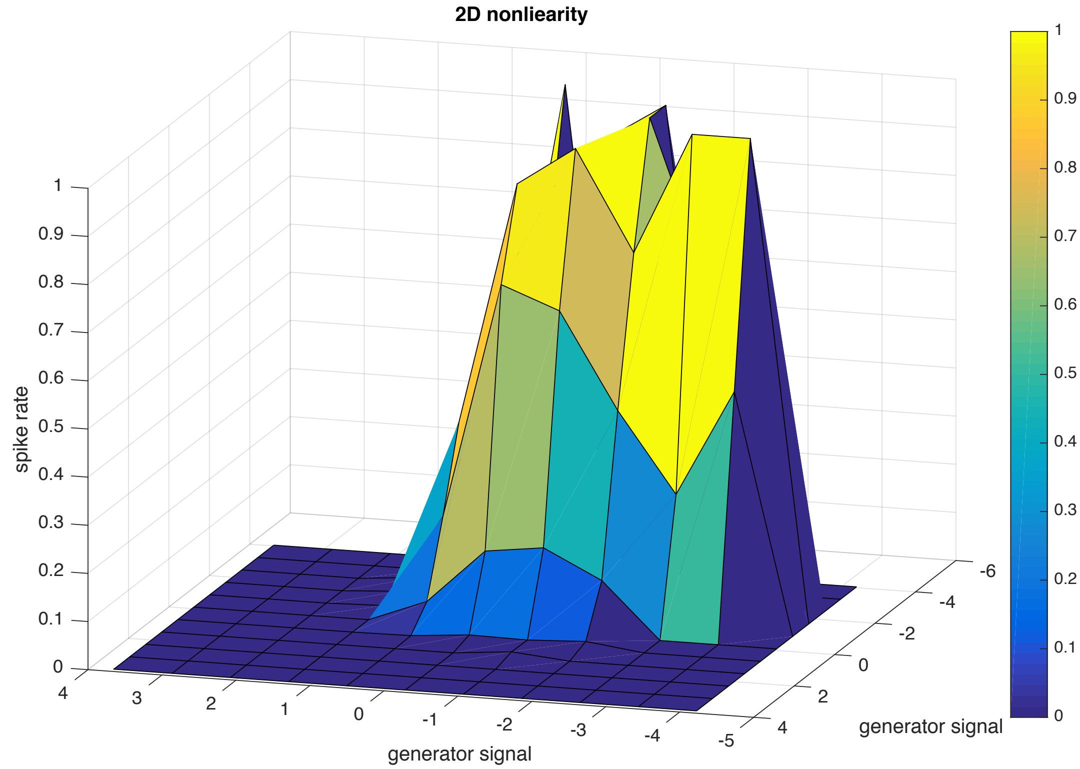

Fig. 14 2D non-linearity F(*,*).

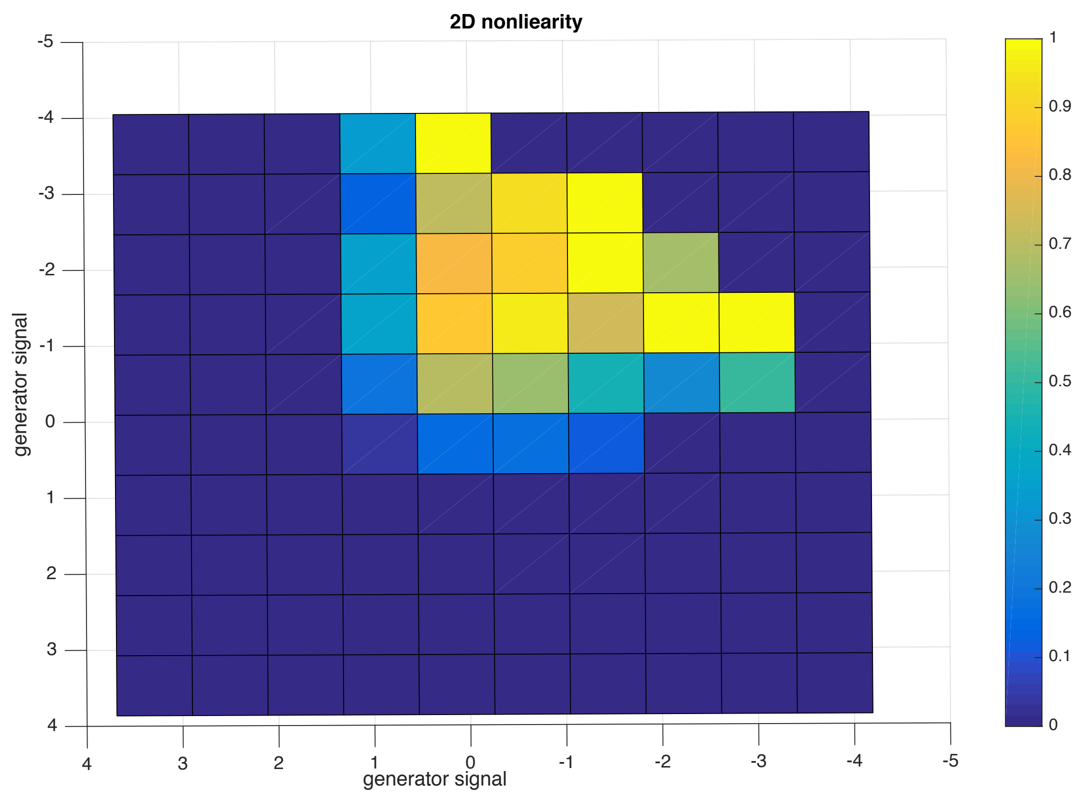

Fig. 15 1D perspective of 2D non-linearity F(*,*).

6.2.3 Steps 3-5 : cross-validation part

Here I applied the filters found at the first step in the Section 6.2.1 to the repeated stimulus; this gives me the generator signals for the repeated stimulus. Then I apply non-linearity F found at the second step in the Section 6.2.2 to these generator signals and get spike rate for the repeated stimulus. Also in order to compare my found spike rate I need to average the number of spikes vector for the repeated stimulus (averaging across 64 repetitions). As a result, we can see that estimated spike rate matches to the measured worse than 1D case with STA filter (Fig. 16). The R-squared test formula (2) is 0.5374. My result goes along with expectations for LNP model with STC filter. The R-squared test should be lower for 2D case simply because we have less data per bin to fit the non-linear function reliably. In fact, STC model was implemented for only the training purpose.

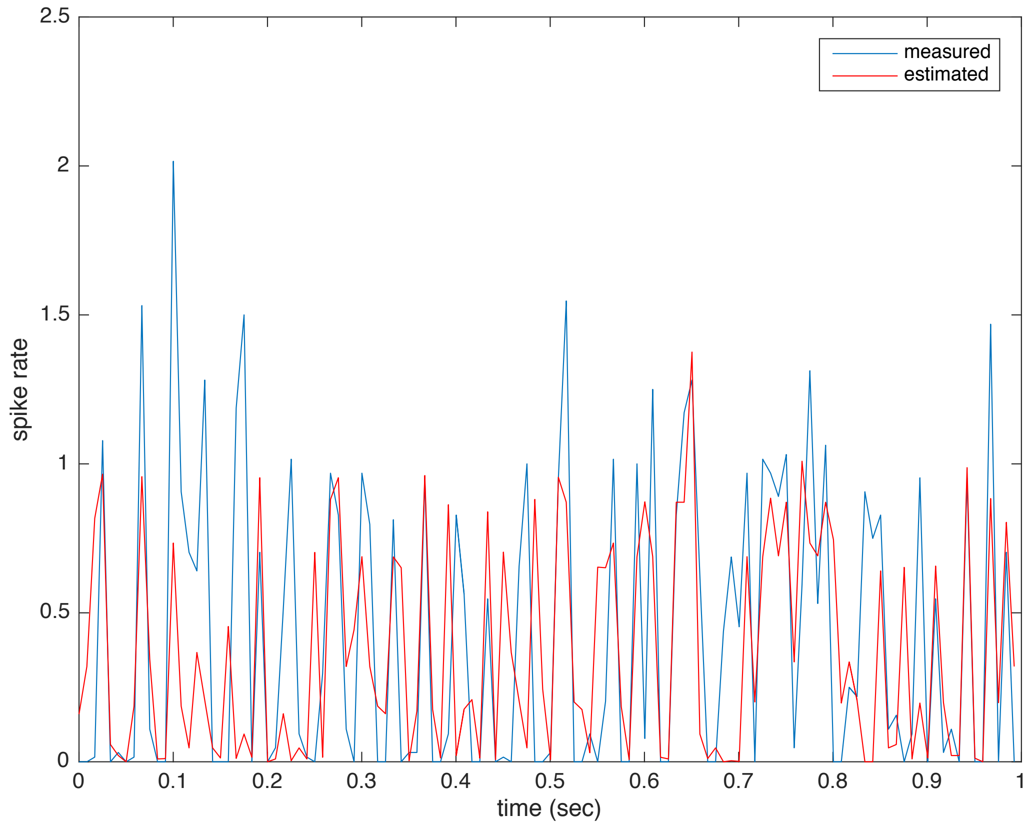

Fig. 16 Cross-validation for spike rates. Red is estimated spike rate, blue line is measured spike rate.

Conclusion

In conclusion, I implemented the Linear-Nonlinear-Poisson model with filters estimated by the two moment-based statistical models, the Spike Triggered Average and the Spike Triggered Covariance models. I validated both STA and STC models using the synthetic data set by obtaining the results similar to the previous research that used the same data set [2]. Next, I reconstructed the LNP model from a real data set and performed cross-validation by comparing the predicted spike rate to the measured one using a different data set from the same neuron. The STA-based model matched the data with a good precision (R-squared approximately 0.77), while the STC-based model resulted in the worse agreement (R-squared approximately 0.54). I suspect this is because the 2D STC-based model requires a much larger data set that the 1D STA-based model. A longer data set would probably improve the performance of the STC-based model.

The present stage of my project was about the basic and rather simple LNP model with two moment-based statistical models. These two moment-based models were the first one which scientists started to use for the firing rate prediction. They work with the given data by applying relatively simple mathematical forms.
In the next semester I moving to more sophisticated models such as Generalized Linear Model, Generalized Quadratic Model and Nonlinear Input Model. These models are based on maximum likelihood estimations. They require complicated parameters' fitting procedures.

7 Implementation

Hardware

- MacBook Air, 1.4 GHz Intel Core i5, 4 GB 1600, MHz DDR3

Software

- Matlab_R2015b

8 Updated project schedule (12/08/16)

October - November

- Implement Spike Triggered Average (STA) and Spike Triggered Covariance models (STC)
- Test models on synthetic data set and validate models on LGN data set

December - Mid February

- Implement Generalized Linear Model (GLM)

- Test model on synthetic data set and validate model on LGN data set

Mid February - Mid April

- Implement Generalized Quadratic Model (GQM) and Nonlinear Input Model (NIM)

- Test models on synthetic data set and validate models on LGN data set

Mid April - May

- Collect results and prepare final report

8.1 Original project schedule (10/01/16)

October - mid November

- Implement STA and STC models

- Test models on synthetic data set and validate models on LGN data set

November - December

- Implement GLM

- Test model on synthetic data set and validate model on LGN data set

January - March

- Implement GQM and NIM

- Test models on synthetic data set and validate models on LGN data set

April - May

- Collect results and prepare final report

Despite some changes to the schedule in the updated version, I am still staying within the time frames for the project, which I set up originally at the meeting with the course instructors in October/2016. I achieved this by decreasing the time for results collection and report preparation from 1.5 month to 1 month. However, I do not think that it should affect the quality of the results collection since most part of it will occur in parallel with testings of my models.

9 Deliverables

- Matlab code for all 5 models (LNP model with STA filter, LNP model with STC filter, GLM,GQM,NIM)
- List of models' fitted parameter
- Reports and presentations
  - Project proposal report and presentation
  - Mid-year progress presentation and Mid Year report
  - Final paper and presentation

10 References

[1] Chichilnisky EJ (2001) A simple white noise analysis of neuronal light responses. Network 12:199 –213.
[2] McFarland JM, Cui Y, Butts DA (2013) Inferring nonlinear neuronal computation based on physiologically plausible inputs. PLoS Computational Biology 9(7): e1003142.
[3] Butts DA, Weng C, Jin JZ, Alonso JM, Paninski L (2011) Temporal precision in the visual pathway through the interplay of excitation and stimulus-driven suppression. J. Neurosci. 31: 11313-27.
[5] Simoncelli EP, Pillow J, Paninski L, Schwartz O (2004) Characterization of neural responses with stochastic stimuli. In: The cognitive neurosciences (Gazzaniga M, ed), pp 327–338. Cambridge, MA: MIT.
[6] Aljadeff J, Lansdell BJ, Fairhall AL, and Kleinfeld D (2016) Analysis of Neuronal Spike Trains, Deconstructed. Neuron, Volume 91, Issue 2, Pages 221–259
[7] Paninski, L., Pillow, J., and Lewi, J. (2006). Statistical models for neural encoding, decoding, and optimal stimulus design.