CHARACTERIZATION OF NONLINEAR NEURON RESPONSES

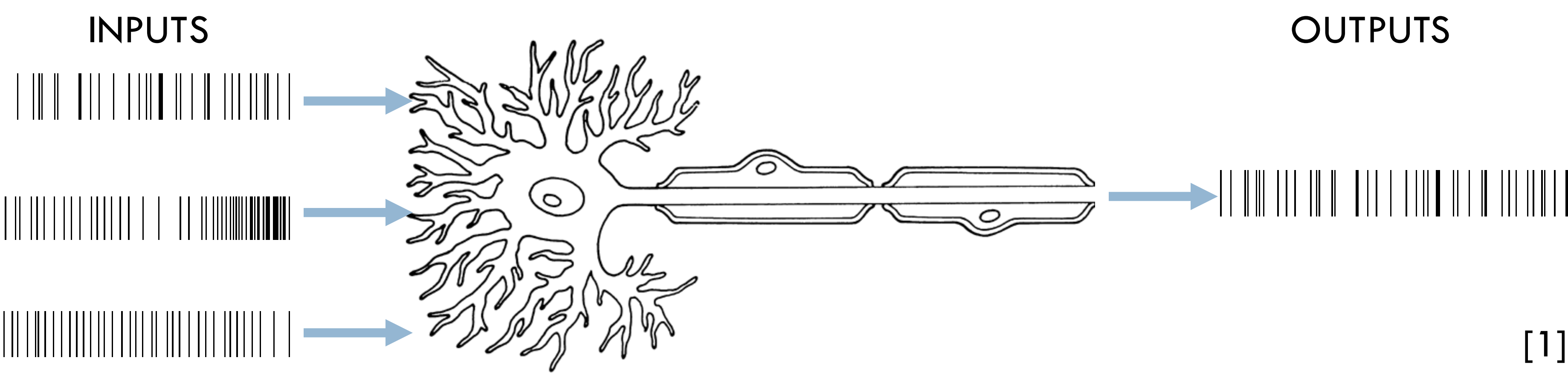
Matt Whiteway
whit8022@umd.edu

Dr. Daniel A. Butts
dab@umd.edu
Neuroscience and Cognitive Science (NACS)
Applied Mathematics and Scientific Computation (AMSC)
Biological Sciences Graduate Program (BISI)

AMSC 663 Mid Year Presentation

Background

- The human brain contains 100 billion neurons
- These neurons process information nonlinearly, thus making them difficult to study



[1]

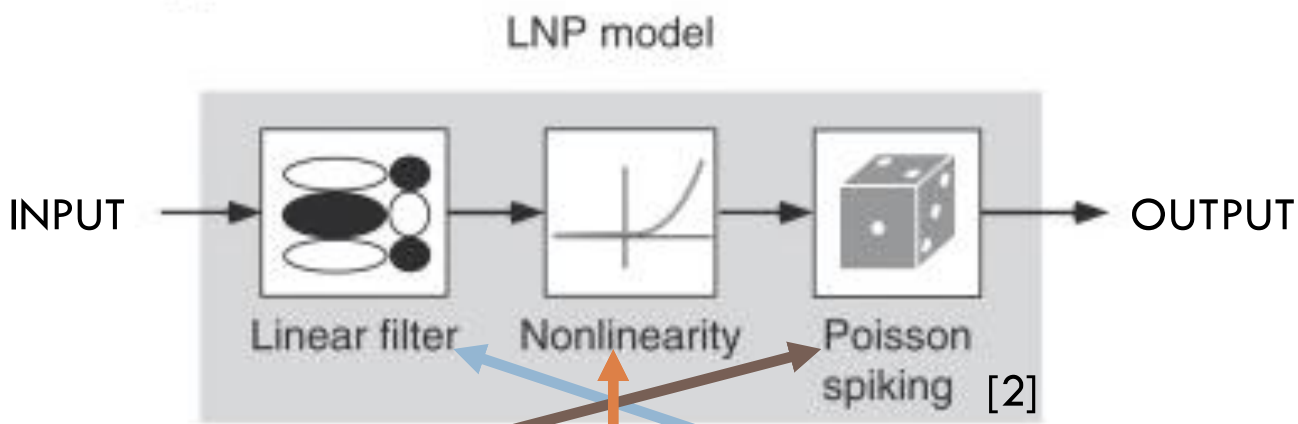
- Given the inputs and the outputs, how can we model the neuron's computation?

[1] http://msjensen.cehd.umn.edu/webanatomy_archive/Images/Histology/

The Models

- Many models of increasing complexity have been developed
- The models I will be implementing are based on statistics
  - Linear Models – Linear Nonlinear Poisson (LNP) Model
    1. **LNP using Spike Triggered Average (STA)**
    2. **LNP using Maximum Likelihood Estimates – Generalized Linear Model (GLM)**
    3. Spike Triggered Covariance (STC)
  - Nonlinear Models
    4. Generalized Quadratic Model (GQM)
    5. Nonlinear Input Model (NIM)

The Models - LNP



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

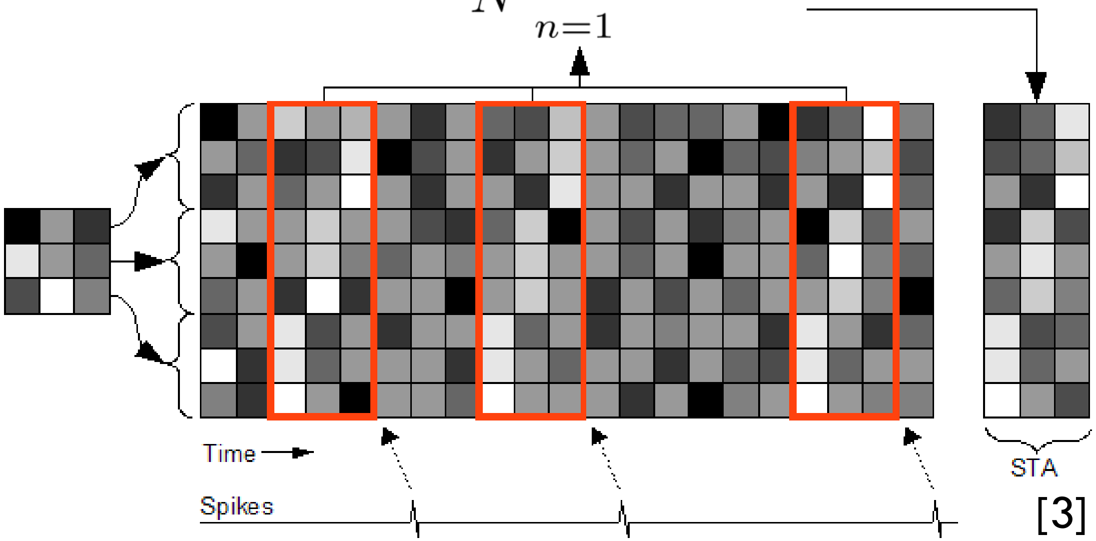
- Knowns
  - $\bar{s}(t)$ is the stimulus vector
  - Spike times
- Unknowns
  - $\bar{k}$ is a linear filter, defines the neuron's stimulus selectivity
  - F is a nonlinear function
  - $r(t)$ is the instantaneous rate parameter of an non-homogenous Poisson process

[2] http://www.sciencedirect.com/science/article/pii/S0079612306650310

STA[1]

- The STA is the average stimulus preceding a spike in the output, where N is the number of spikes and $\bar{s}_{spike}(t)$ is the set of stimuli preceding a spike

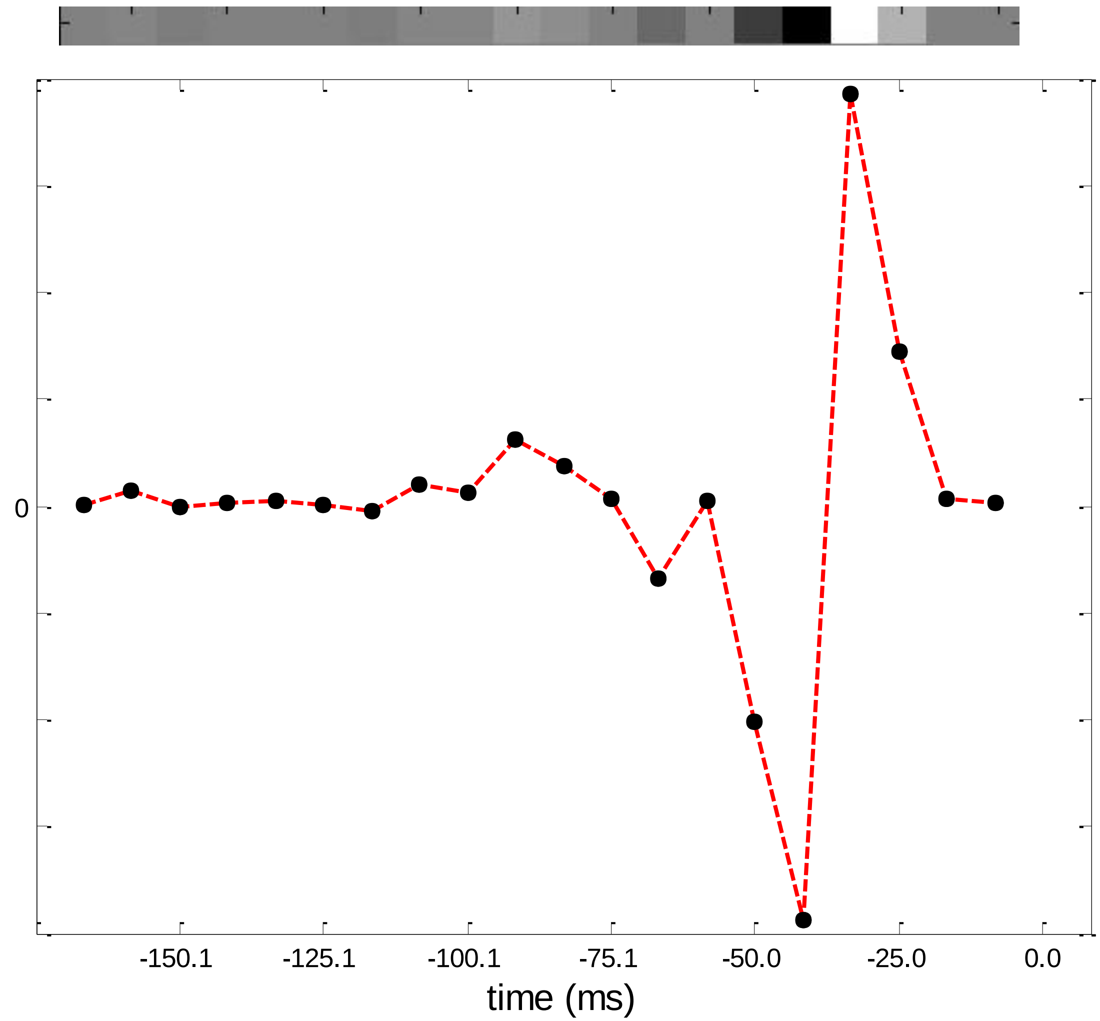
$$STA = \frac{1}{N}\sum_{n=1}^{N} \bar{s}_{spike}(t)$$



1. Chichilnisky, E.J. (2001) A simple white noise analysis of neuronal light responses.
[3] http://en.wikipedia.org/wiki/Spike-triggered_average

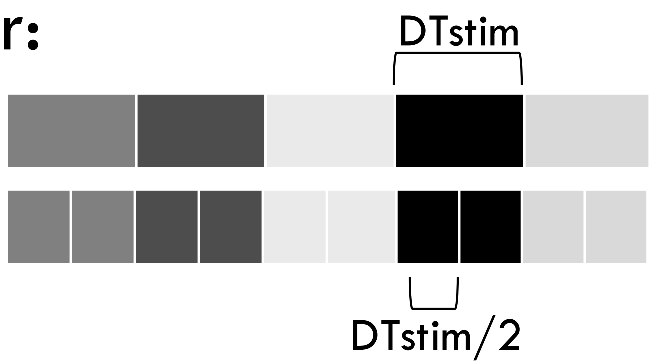
STA Implementation

- Filter length of 20 time steps
- Upsampling factor of 1



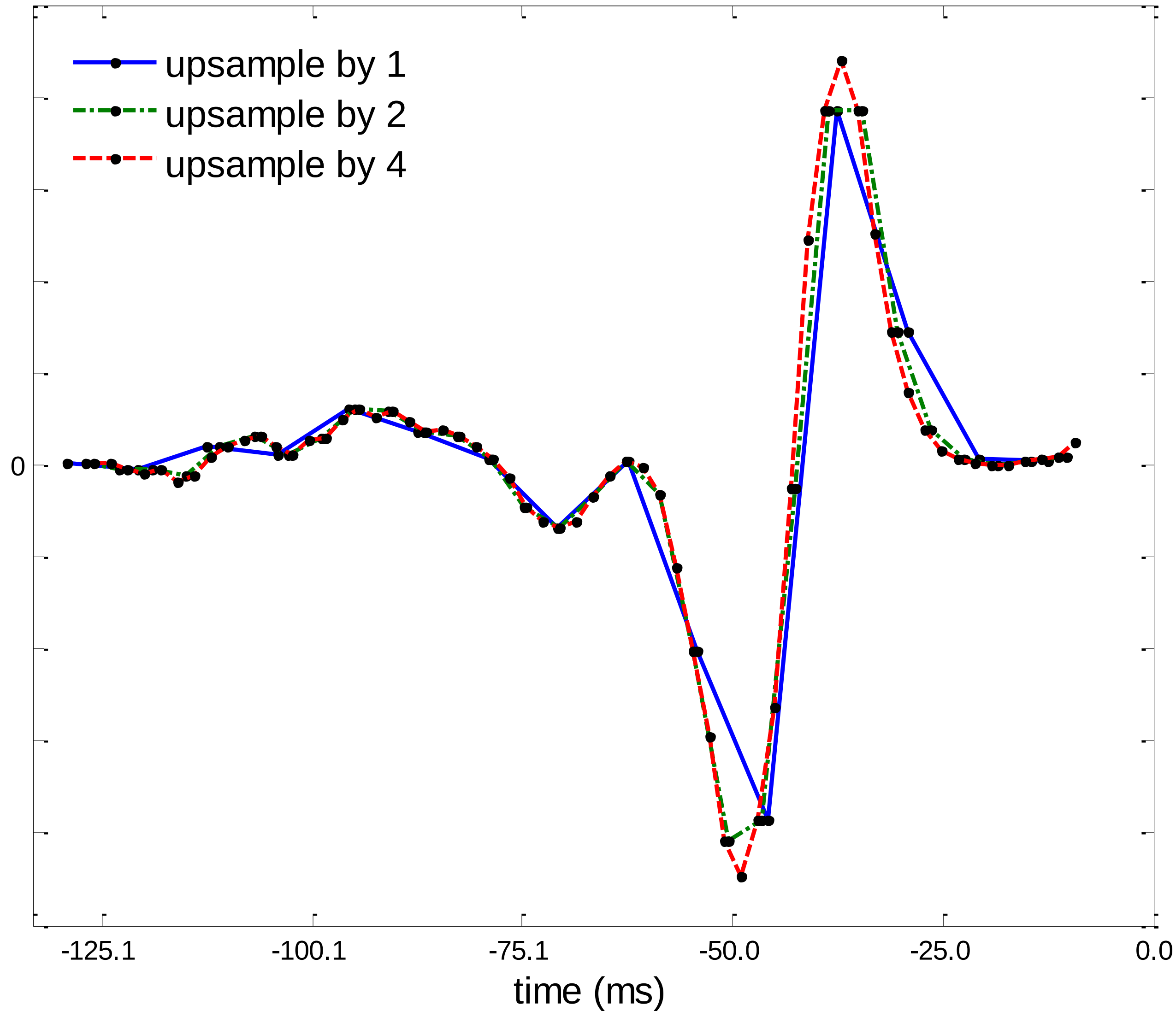
STA Implementation

- Resolution of the filter is determined by time interval between measurements
- We can artificially increase resolution by upsampling the stimulus vector:
  - Upsample by 1:
  - Upsample by 2:



STA Implementation

- Filter length of 15 time steps



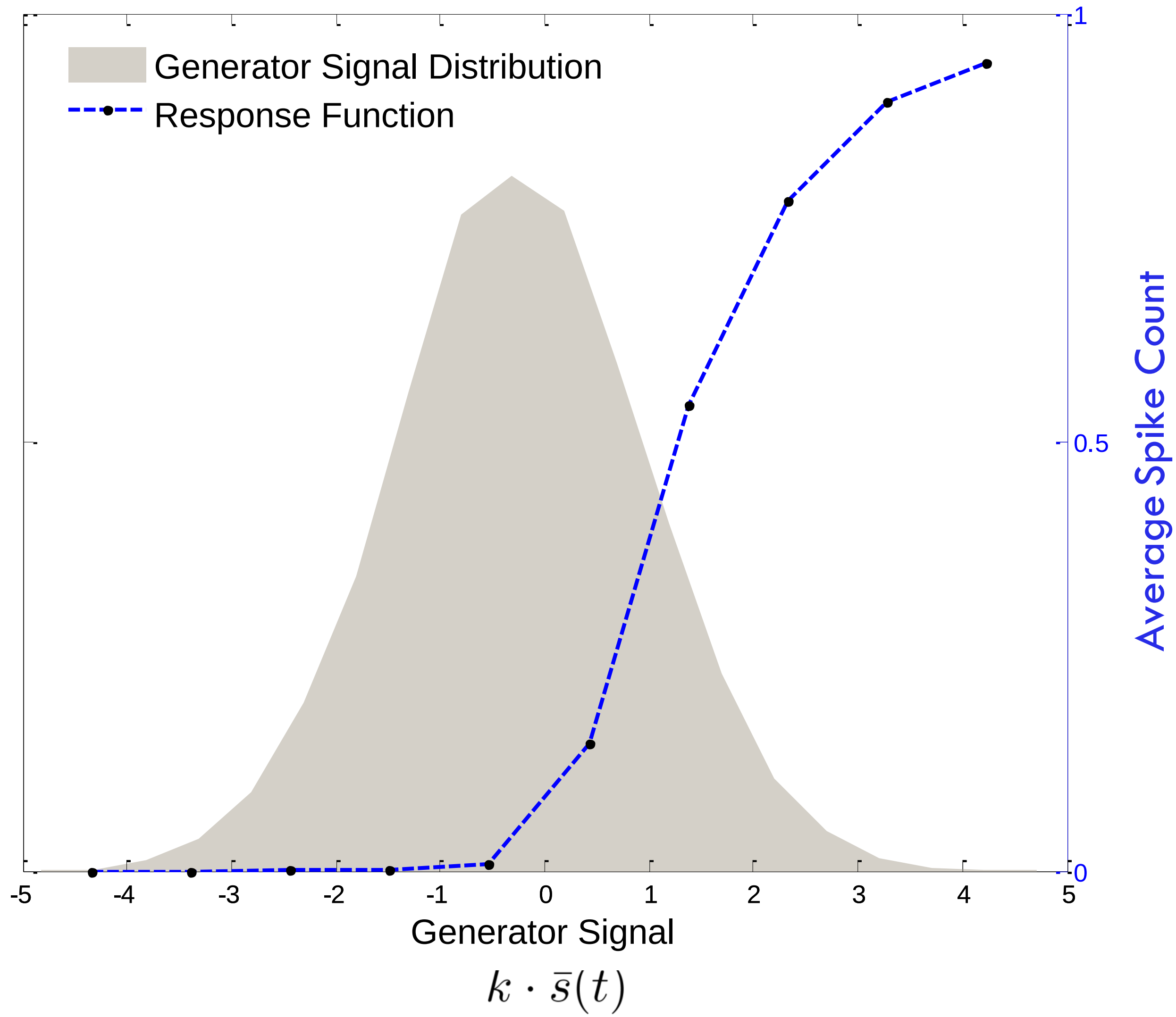
STA Implementation

- For the STA, a common approach to finding the nonlinear response function is to use the histogram method
- Method creates bins for the generator signal, $\bar{k} \cdot \bar{s}(t)$, and plots average number of spikes for each bin

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

STA Implementation

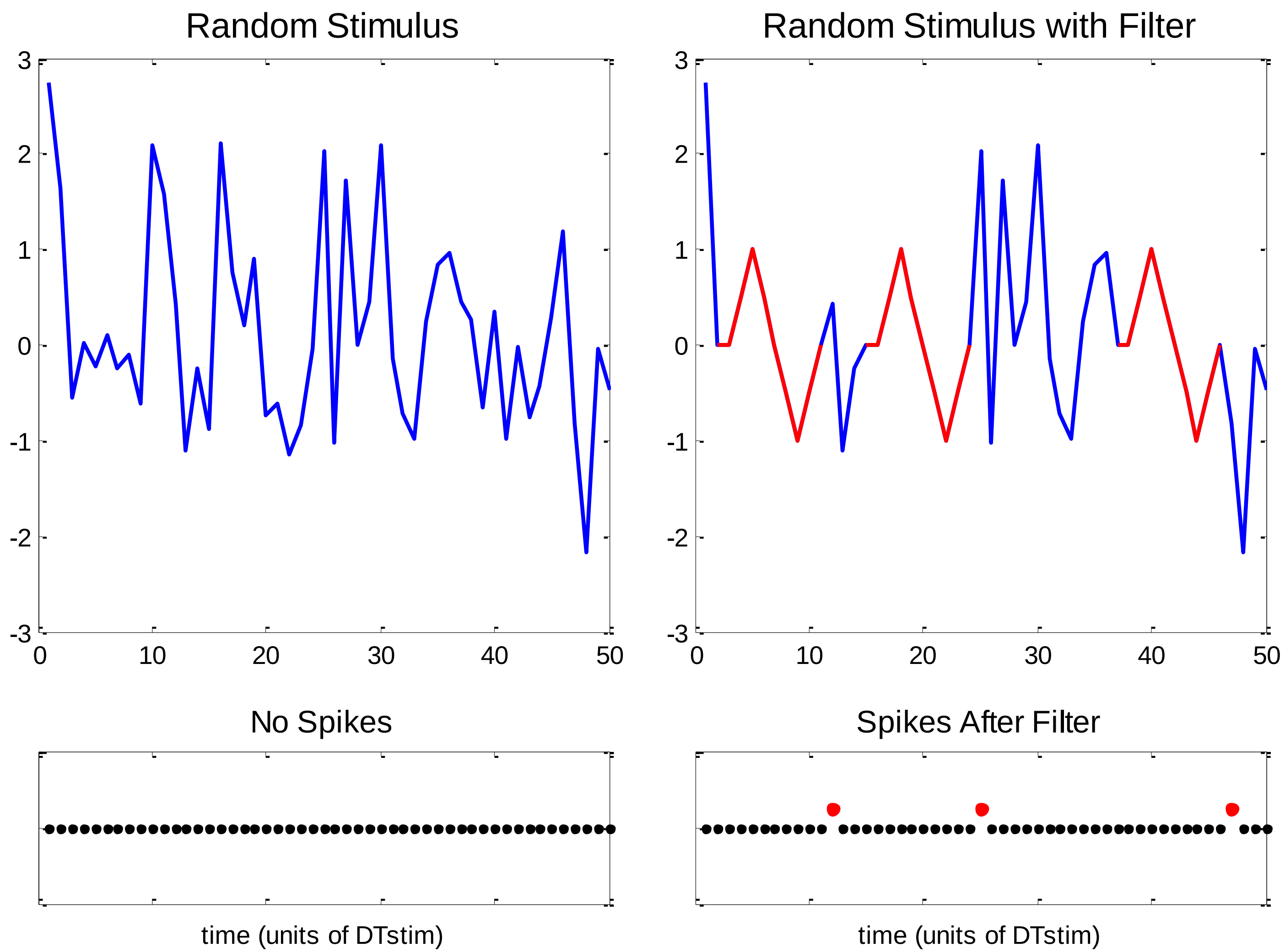
- Filter length of 15 time steps
- Upsampling factor of 1



STA Validation

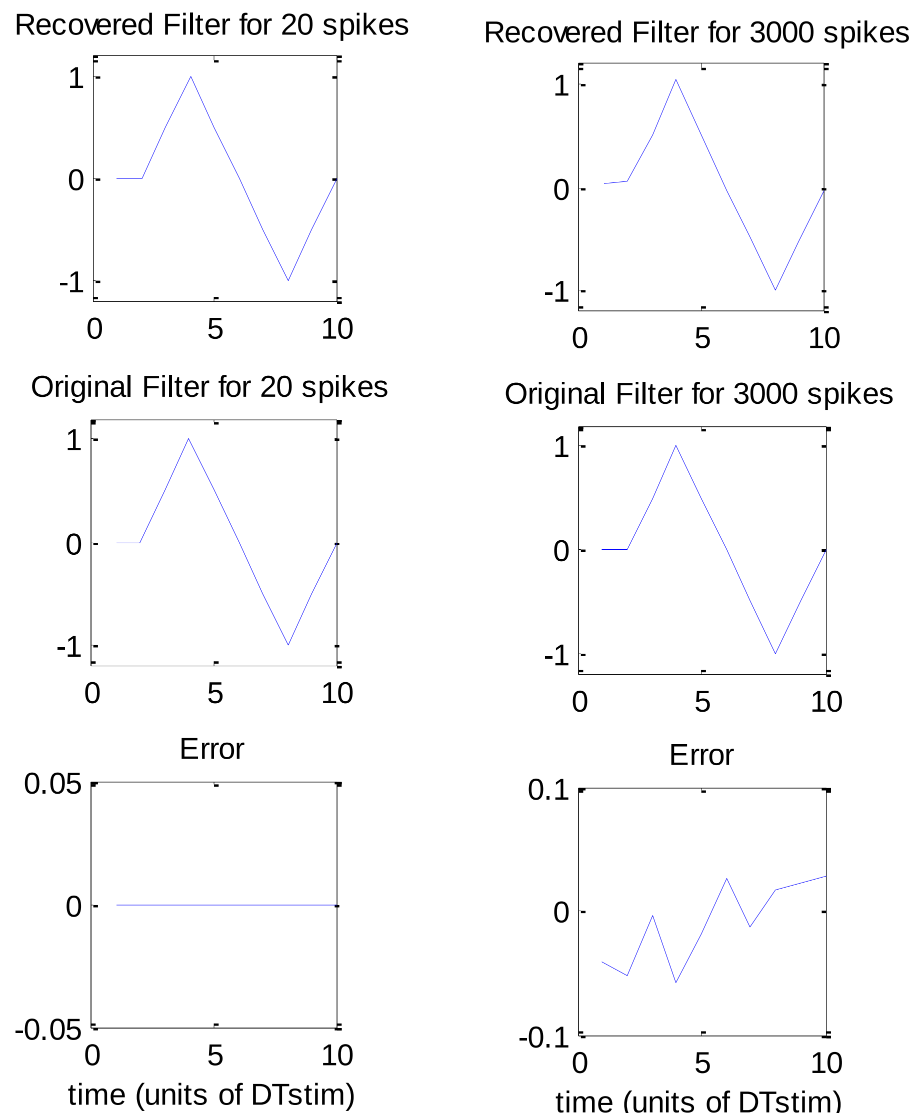
- For filter validation, I
  - created a stimulus with Gaussian random noise
  - added an artificial filter at random points
  - Recorded a spike for each instance of the artificial filter
- If the STA code is working properly, and if none of the artificial filters overlap, then the code should exactly recover the artificial filter

STA Validation

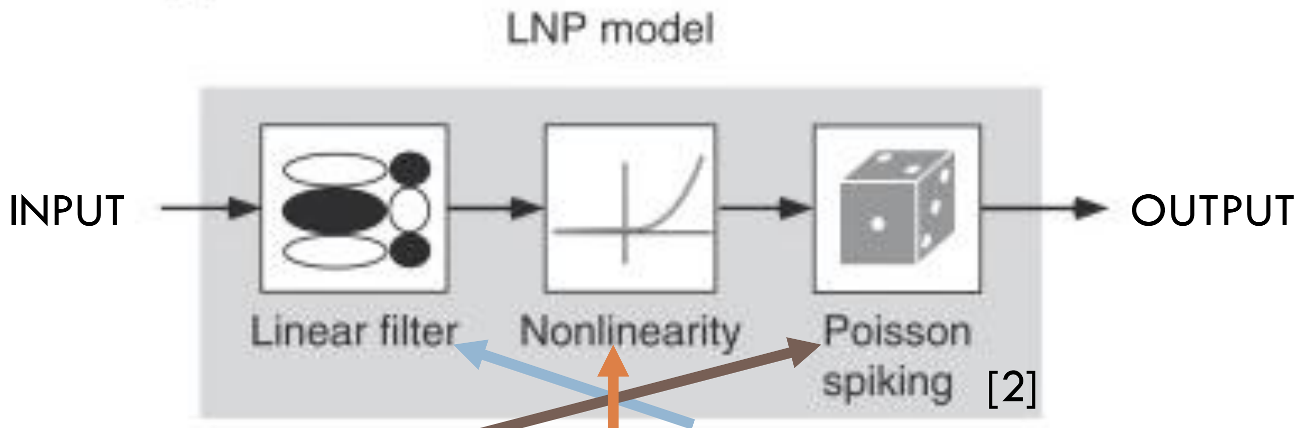


STA Validation

- filter length: 10
- stimulus length: 15000
- 20 spikes: No overlap of filters in stimulus, STA code recovers exact filter
- 3000 spikes: Substantial overlap of filters in stimulus, STA code recovers exact filter with some error



The Models - LNP



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

- Knowns
  - $\bar{s}(t)$ is the stimulus vector
  - Spike times
- Unknowns
  - $\bar{k}$ is a linear filter, defines the neuron's stimulus selectivity
  - F is a nonlinear function
  - $r(t)$ is the instantaneous rate parameter of an non-homogenous Poisson process

[2] http://www.sciencedirect.com/science/article/pii/S0079612306650310

GLM[2]

- Now we will approximate the linear filter using the Maximum Likelihood Estimate (MLE)
- A likelihood function is the probability of an outcome Y given a probability density function with parameter θ
- The LNP model uses the Poisson distribution

$$P(Y|\theta) = \prod_t \frac{(r(t)\Delta)^{y_t}}{y_t!} e^{-r(t)\Delta}$$

where Y is the vector of spike counts binned at a resolution Δ

- We want to maximize a log-likelihood function

$$\mathcal{L} = \sum_{t=spike} log(r(t)) - \Delta \sum_t r(t) + constants$$

2. Paninski, L. (2004) Maximum Likelihood estimation of cascade point-process neural encoding models.

GLM Implementation

- Can employ likelihood optimization methods to obtain maximum likelihood estimates for linear filter $\bar{k}$
- If we make some assumptions about the form of the nonlinearity F, the likelihood function has no non-global local maxima – gradient ascent!
  - F(u) is convex in u
  - log(F(u)) is concave in u
- I use F(u) = log(1+exp(u-c))

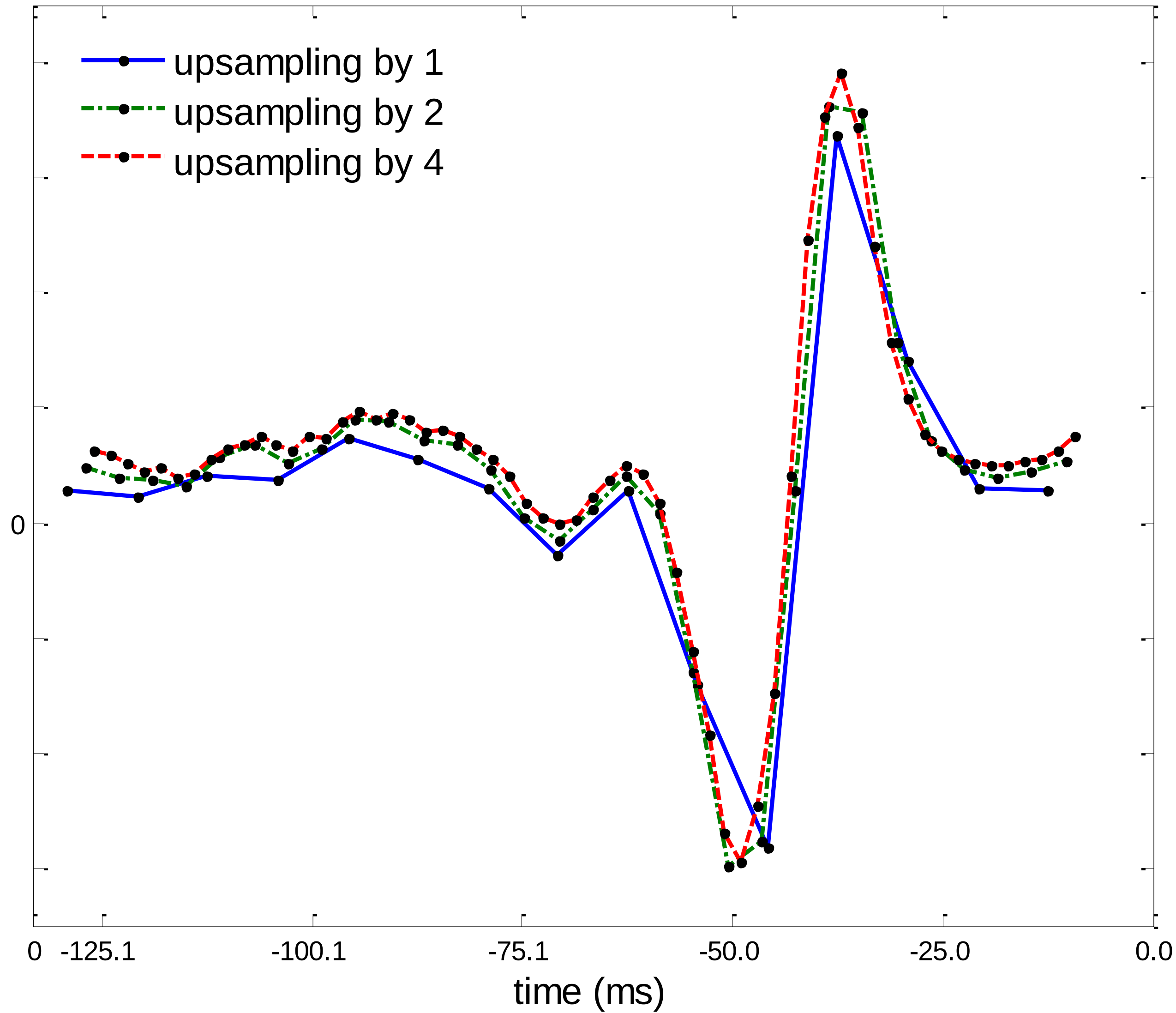
$$\mathcal{L} = \sum_{t=spike} log(log(1 + exp(\bar{k} \cdot \bar{s}(t) - c))) - \Delta \sum_{t} log(1 + exp(\bar{k} \cdot \bar{s}(t) - c))$$

GLM Implementation

- Originally coded a gradient descent method – took too many function evaluations
  - About 1000 iterations for a filter of length 15 at ~1s per function evaluation
- Next used a Newton-Raphson method – less iterations, but needed to compute Hessian
  - About 150 for a filter of length 15 at ~2s per function evaluation
- Need a quasi-Newton method
- Now use Matlab's fminunc
  - About 10 – 150 iterations at ~1s per function evaluation

GLM Implementation

- Filter length of 15 time steps

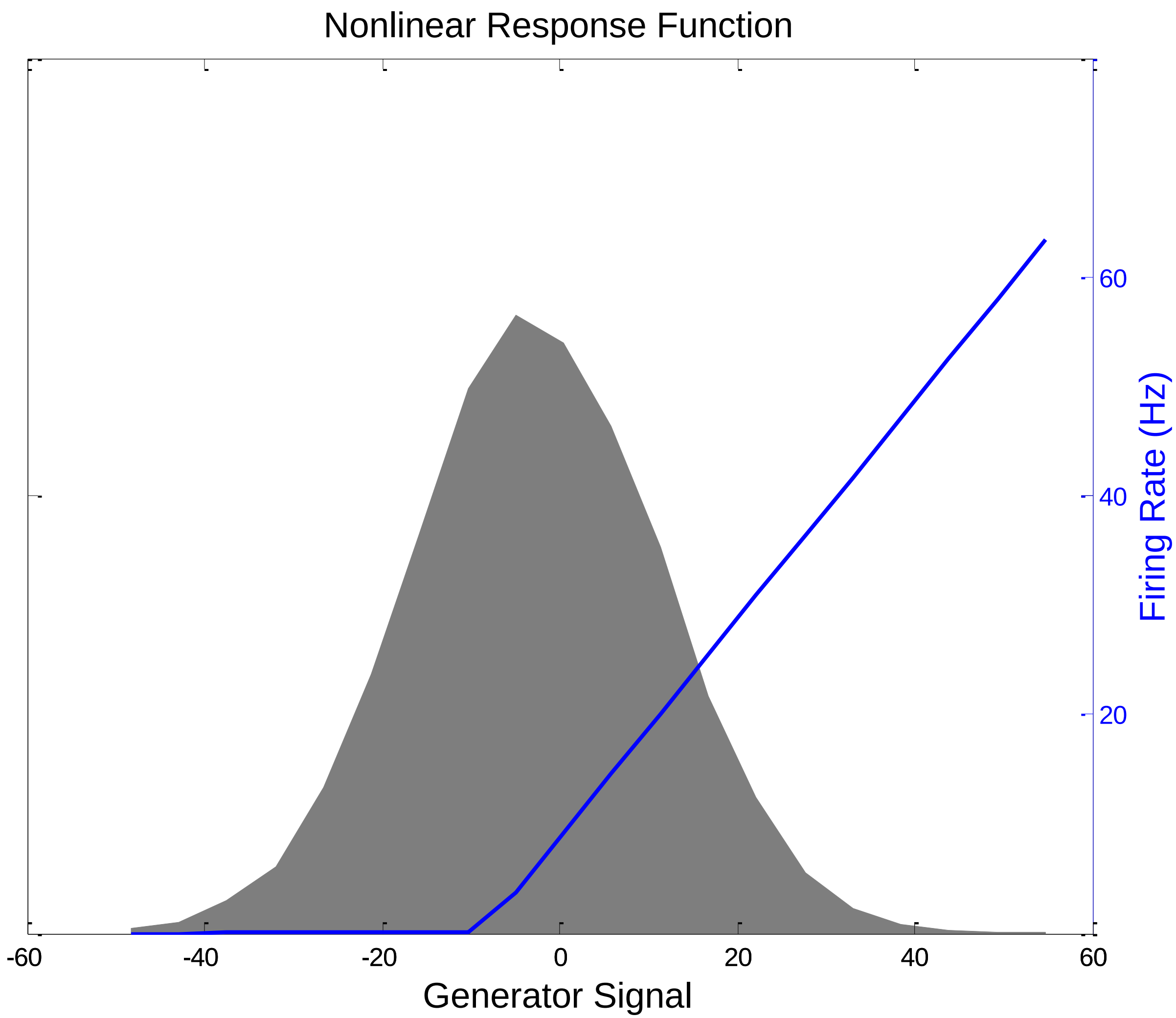


GLM Implementation

- For the GLM, finding the parameters to the nonlinearity can be done at the same time as finding the filter
- Assume the parametric form F(u) and include its parameter(s) in the optimization
  - Use log(1+exp(x-c)), fit offset c

GLM Implementation

- Filter length of 15 time steps
- Upsampling factor of 1



GLM Implementation

- We can also use regularization to add additional prior knowledge about solution attributes
- We know the filters should be smoothly varying in time
  - Penalize large curvatures in filter
  - Laplacian gives us the second derivative; we want to maximize likelihood while minimizing the L2 norm of the Laplacian of the filter

$$\mathcal{L} = \sum_{t=spike} log(r(t)) - \Delta \sum_{t} r(t) - \lambda \|L^t \bar{k}\|_2^2$$

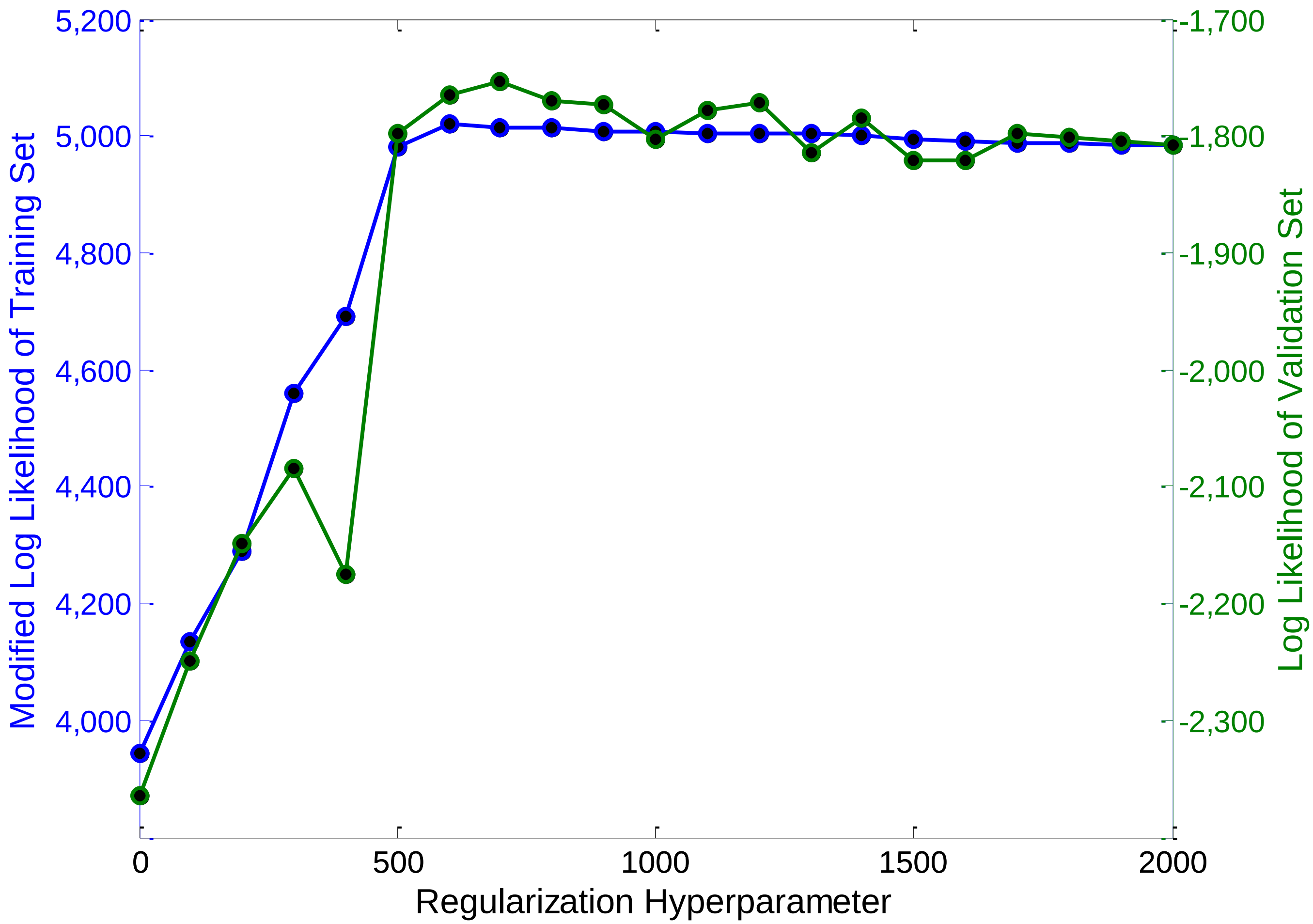
- λ is a parameter that is not explicitly part of the model, hence called a hyperparameter

GLM Implementation

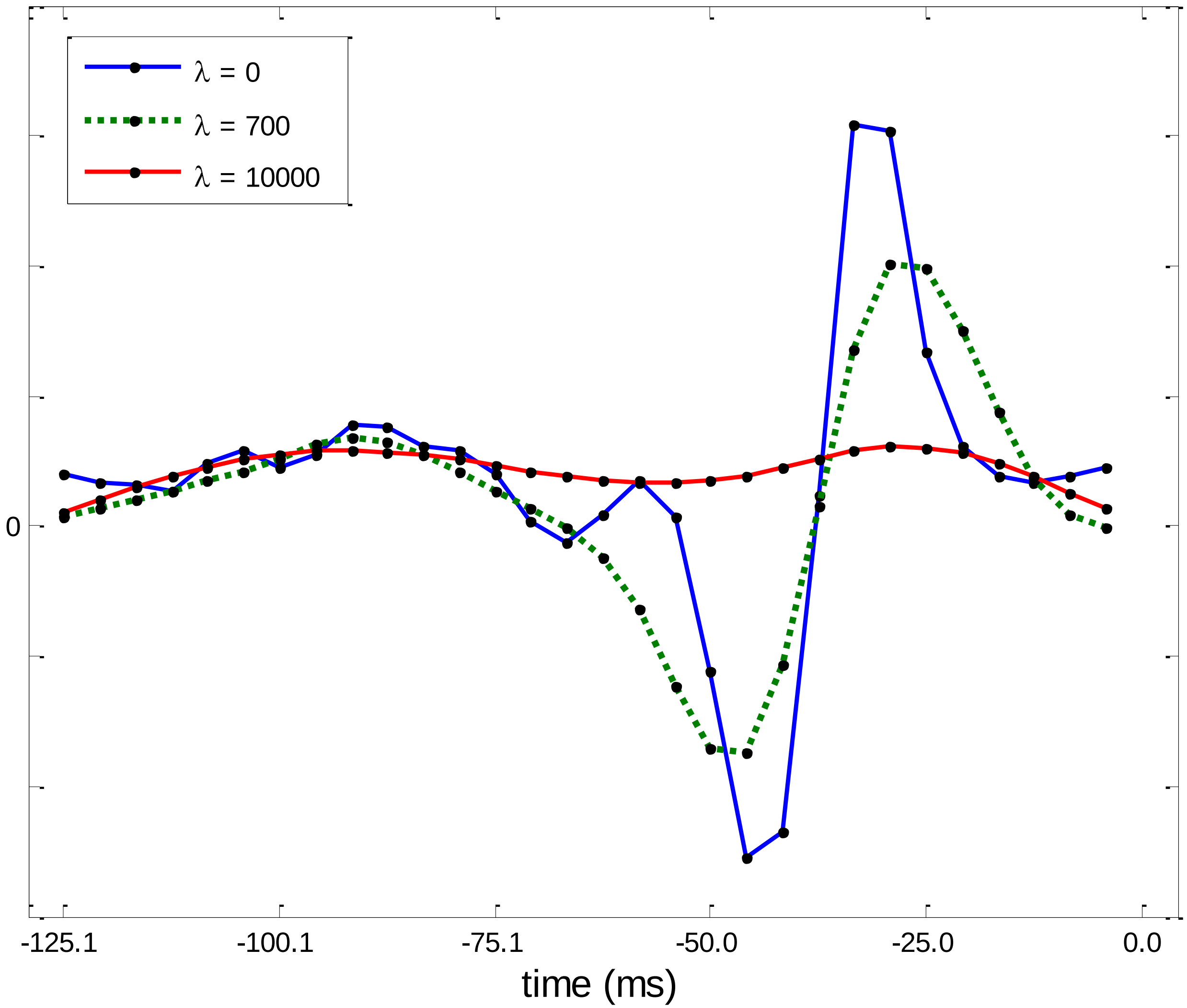
- How to choose optimal λ?
- For a variety of λ values,
  - fit model parameters using part of the data (80%)
  - validate model on rest of the data (20%)

GLM Implementation

- Filter length of 15 time steps
- Upsampling factor of 2

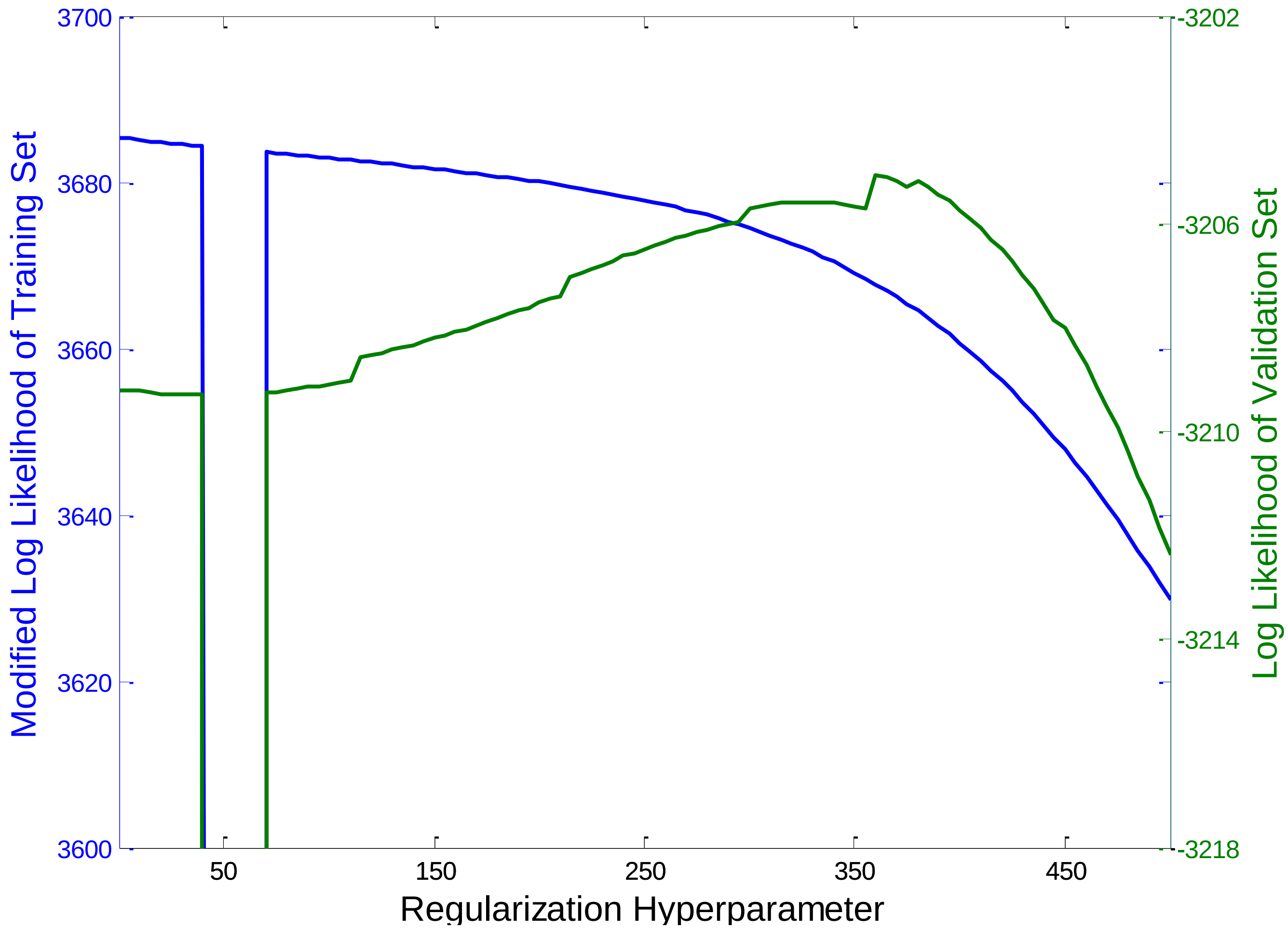


GLM Implementation

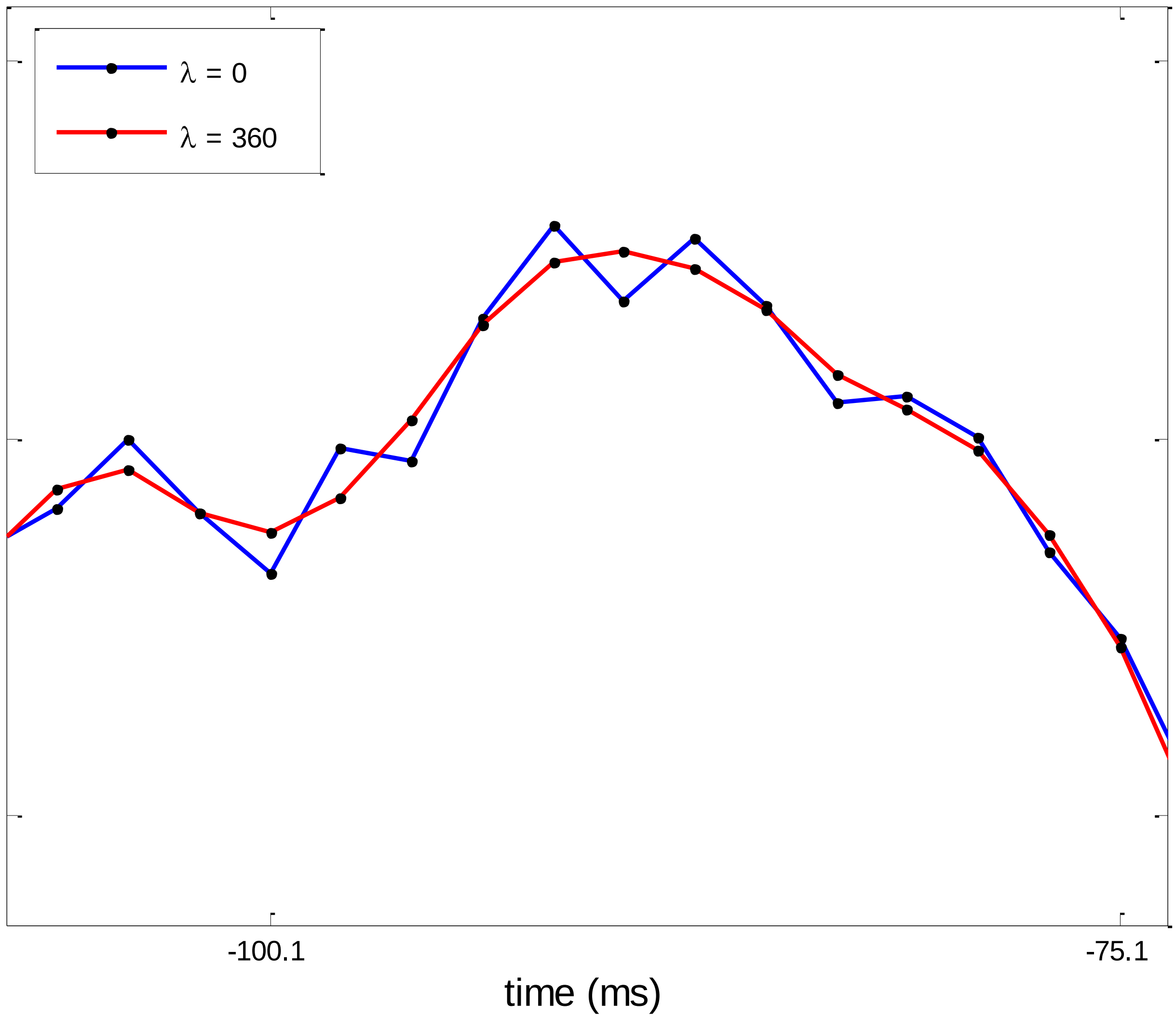


GLM Implementation

- Filter length of 15 time steps
- Upsampling factor of 4



GLM Implementation



Schedule

- PHASE I (October-December)
  - Implement and validate the LNP model using the STA (October)
  - Develop code for gradient descent method and validate (October)
    - Done, but not efficient enough. I am currently using MATLAB's fminunc command instead
  - Implement and validate the GLM with regularization (November-December)
  - Complete mid-year progress report and presentation (December)

Schedule

- PHASE II (January-May)
  - Implement quasi-Newton method for gradient descent (January)
  - Implement and validate the LNP model using the STC (January-February)
  - Implement and validate the GQM with regularization (February)
  - Implement and validate the NIM with regularization using rectified linear upstream functions (March)
  - Test all models (April)
  - Complete final report and presentation (April-May)

References

- Chichilnisky, E.J. (2001) A simple white noise analysis of neuronal light responses. *Network: Comput. Neural Syst.,* 12, 199-213.
- Schwartz, O., Chichilnisky, E. J., & Simoncelli, E. P. (2002). Characterizing neural gain control using spike-triggered covariance. *Advances in neural information processing systems*, *1*, 269-276.
- Paninski, L. (2004) Maximum Likelihood estimation of cascade point-process neural encoding models. *Network: Comput. Neural Syst.* ,15, 243-262.
- Schwartz, O. et al. (2006) Spike-triggered neural characterization. *Journal of Vision*, 6, 484-507.
- Paninski, L., Pillow, J., and Lewi, J. (2006) Statistical models for neural encoding, decoding, and optimal stimulus design.
- Park, I., and Pillow, J. (2011) Bayesian Spike-Triggered Covariance Analysis. *Adv. Neural Information Processing Systems* ,24, 1692-1700.
- Butts, D. A., Weng, C., Jin, J., Alonso, J. M., & Paninski, L. (2011). Temporal precision in the visual pathway through the interplay of excitation and stimulus-driven suppression. *The Journal of Neuroscience*, *31*(31), 11313-11327.
- McFarland, J.M., Cui, Y., and Butts, D.A. (2013) Inferring nonlinear neuronal computation based on physiologically plausible inputs. *PLoS Computational Biology*.

Figures

1. http://msjensen.cehd.umn.edu/webanatomy_archive/Images/Histology/
2. http://www.sciencedirect.com/science/article/pii/S0079612306650310
3. http://en.wikipedia.org/wiki/Spike-triggered_average