Discussion on “Dynamic Analysis of Neural Encoding by Point Process Adaptive Filtering”

Zhang GAO

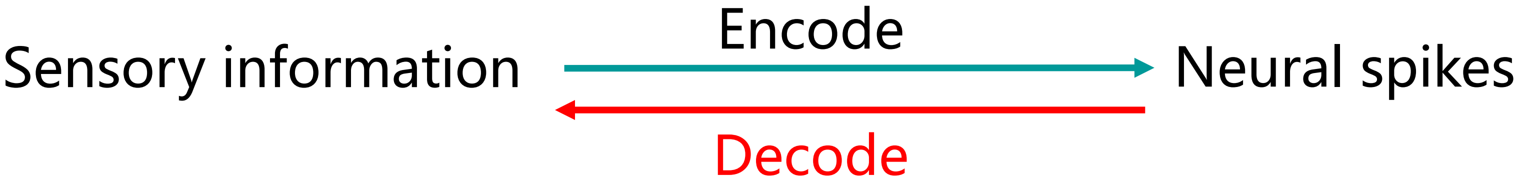
Outline

1. Overview

2. Simulation Results Analysis

1. Overview

Problem



Sensory information

Encode

Neural spikes

Decode

Decode biological information θ from observed neural spikes.

- "Biological information" includes
  - Sensory information
  - Parameters in encode models
- Neural spikes are characterized by
  - Spike times
  - Spike numbers in a fixed-length time period

1. Overview

Contributions

1. Modeling the encode process by point process

System evolvement $\boldsymbol{\theta}_{\mathbf{k}} = f\left(\boldsymbol{\theta}_{\mathbf{k-1}}\right)$

Observation $\Delta N_k \sim \Pr\left(\Delta N_k \mid \boldsymbol{\theta}_{\mathbf{k}}, \mathbf{H}_{\mathbf{k}}\right) = \begin{cases} 1-\lambda(t_k \mid \boldsymbol{\theta}_{\mathbf{k}}, \mathbf{H}_{\mathbf{k}})\Delta t, \Delta N_k = 0 \\ \lambda(t_k \mid \boldsymbol{\theta}_{\mathbf{k}}, \mathbf{H}_{\mathbf{k}})\Delta t, \Delta N_k = 1 \end{cases}$

$\lambda(t_k \mid \boldsymbol{\theta}_{\mathbf{k}}, \mathbf{H}_{\mathbf{k}})$ is the intensity function generalized from inhomogeneous Poisson process

2. SSSPF (Stochastic State Point Process Filter) algorithm for state estimation

Formulae (2.7) ~ (2.10)

1. Overview

SSSPF Summary

Based on the approximation of the Bernoulli probability,

$$\begin{aligned} &\Pr(\Delta N_k \text{ spikes in } (t_{k-1}, t_k] \mid \boldsymbol{\theta}_k, \mathbf{H}_k) \\ &\quad = \exp(\Delta N_k \log(\lambda(t_k \mid \boldsymbol{\theta}_k, \mathbf{H}_k)\Delta t_k) - \lambda(t_k \mid \boldsymbol{\theta}_k, \mathbf{H}_k)\Delta t_k) \end{aligned} \tag{2.3}$$

the article estimates states by maximizing the instantaneous log likelihood.

$$\max_{\theta} \sum_{j=1}^{C} \Delta\mathbf{N}_k^j \log(\lambda^j) - \lambda^j \Delta t_k$$

System evolvement model are involved to utilize the prior knowledge, and provide confidence bounds for estimates.

1. Overview

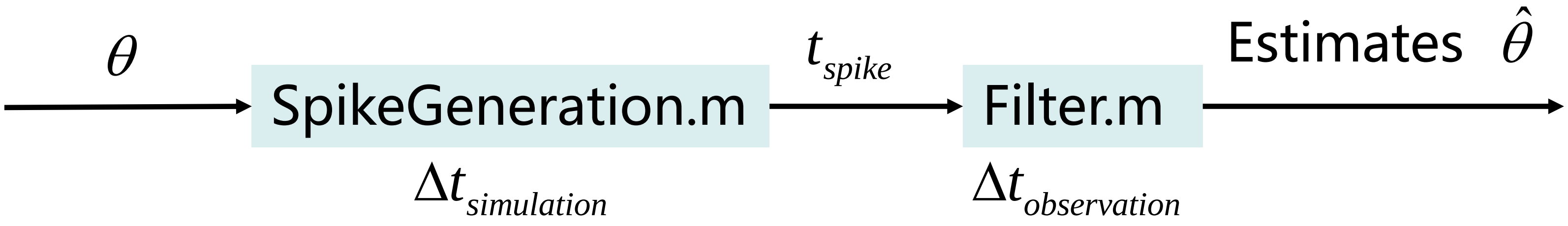
SSSPF Compared To Other Methods

- EKF
  - Introducing system evolvement model
  - Minimizing mean-square error of parameters
  - Does not promise optimality in nonlinear cases
  - Does not contain second order terms

- Steepest Descent Method
  - Without system evolvement model
  - General framework for unconstraint optimizing problems

2. Simulation Results Analysis

Code Structure



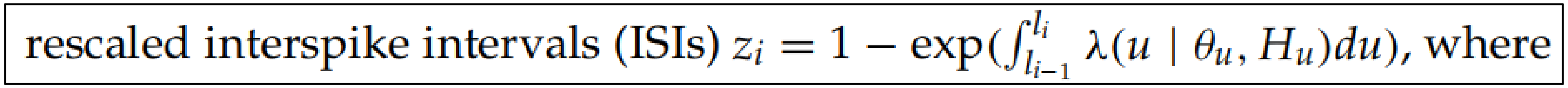
θ → SpikeGeneration.m → t_{spike} → Filter.m → Estimates $\hat{\theta}$

SpikeGeneration.m: $\Delta t_{simulation}$

Filter.m: $\Delta t_{observation}$

Difference Between Code & Article

- Typo in Page 798.



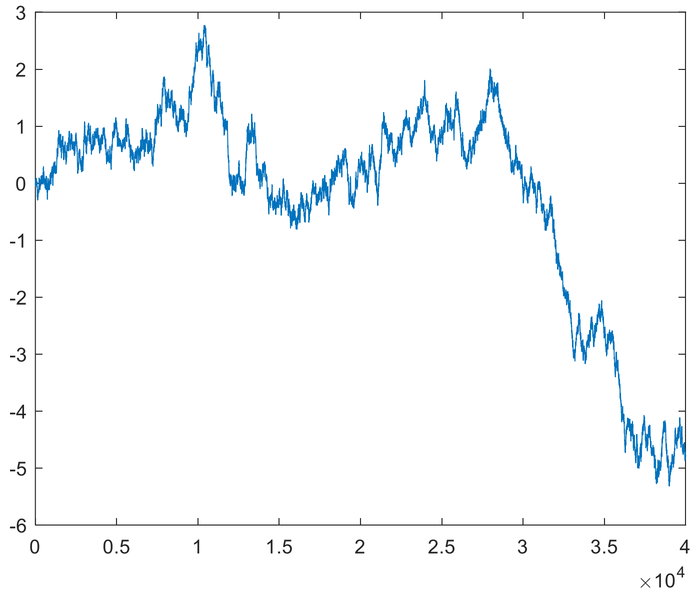
rescaled interspike intervals (ISIs) $z_i = 1 - \exp(\int_{l_{i-1}}^{l_i} \lambda(u \mid \theta_u, H_u)du)$, where

- Experiment2: Velocity Decoding
  - True evolvement model
  - Rescaled velocity range

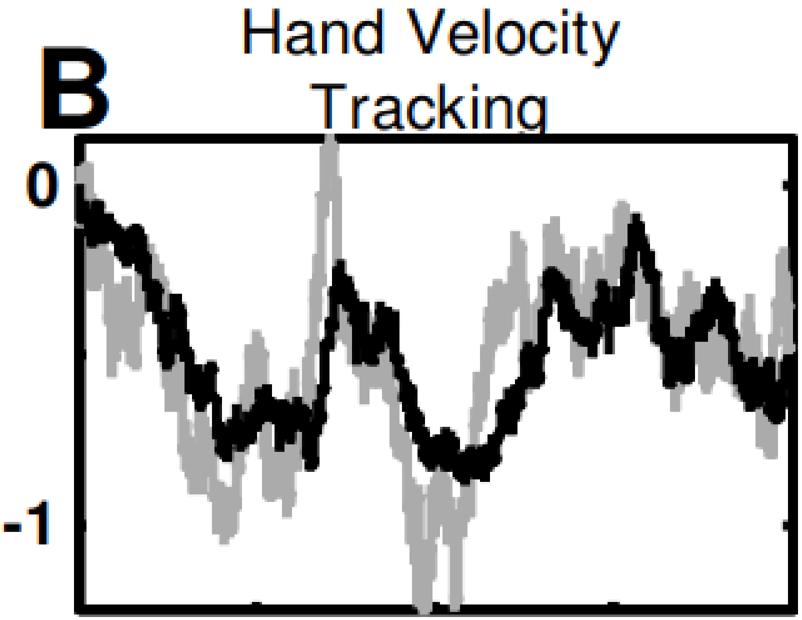
2. Simulation Results Analysis

Experiment 2: Velocity Decoding

1. Velocity scale.



Generated velocity scales from -5 to 3.



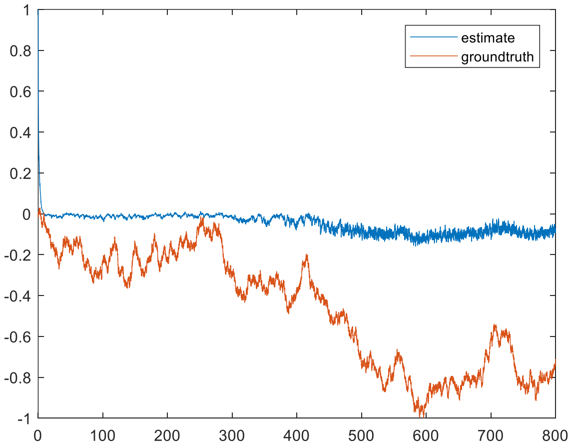
The velocity plotted in the article scales from -1 to 0.

Rescale operation on generated velocity.

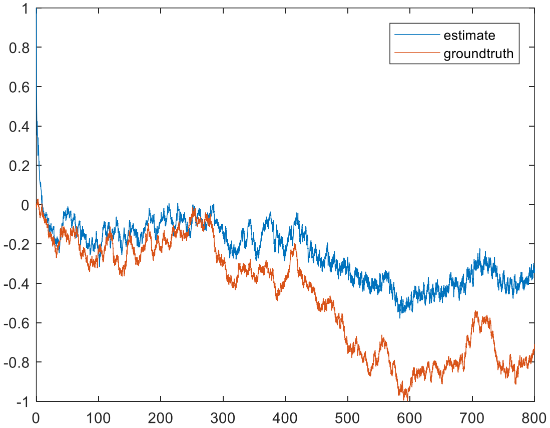
2. Simulation Results Analysis

Experiment 2: Velocity Decoding

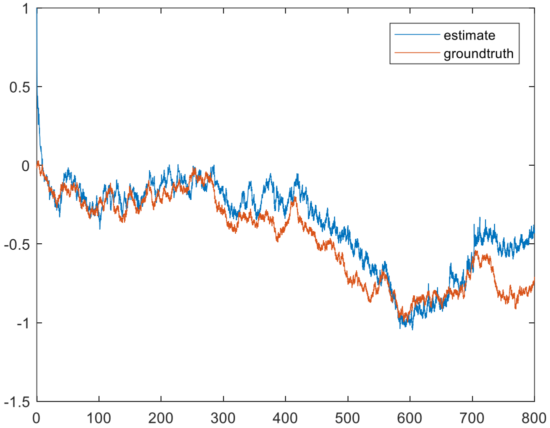
2. Accuracy of evolvement model.



F=0.99



F=0.999



F=1.0

Model information plays a dominant role in recursive calculation here.

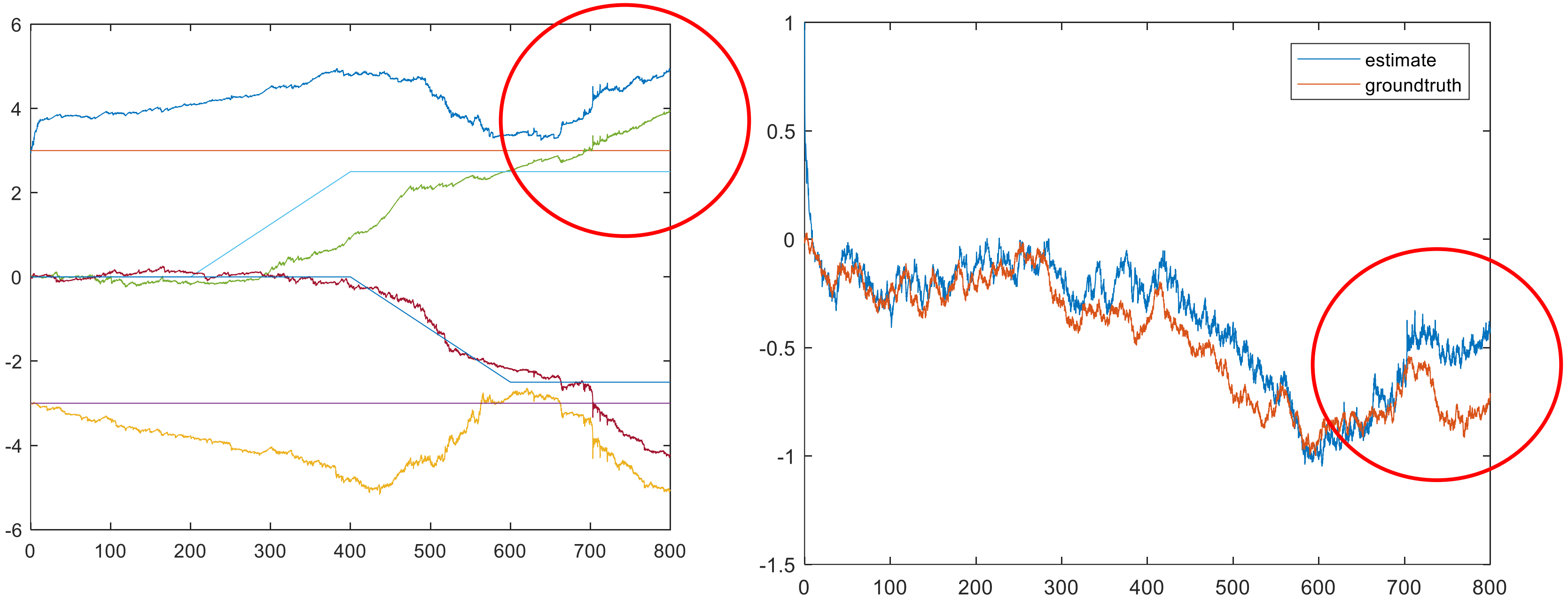
2. Simulation Results Analysis

Experiment 2: Velocity Decoding

3. Ill-Posed problem.

$$\lambda^j(t_k) = \exp(\mu + \beta_k^j v_k)$$

For every beta, here exists a v to get the same products.

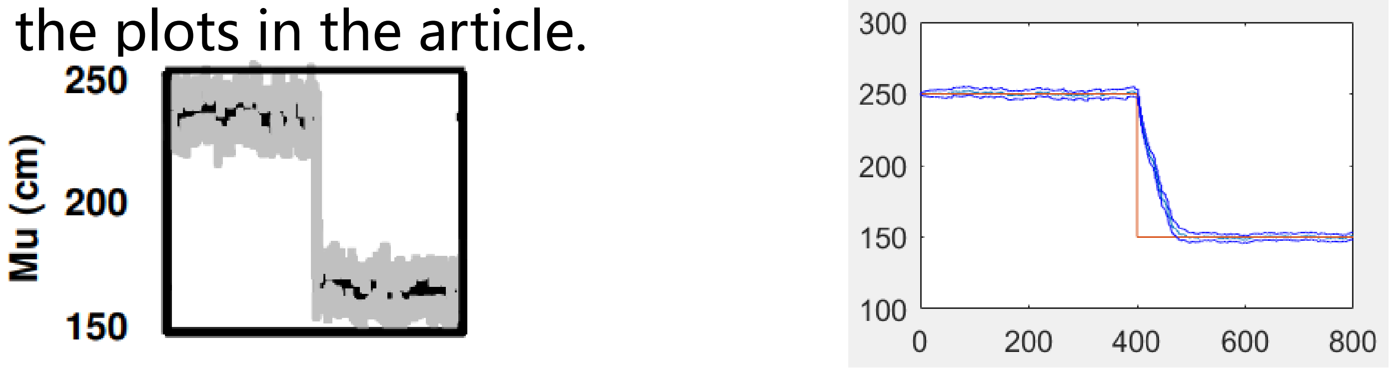


The larger beta estimates, the smaller v estimates.

2. Simulation Results Analysis

Experiment 1: Spatial Receptive Field Analysis

1. The code shows the tracking ability, but performance is not as well as the plots in the article.

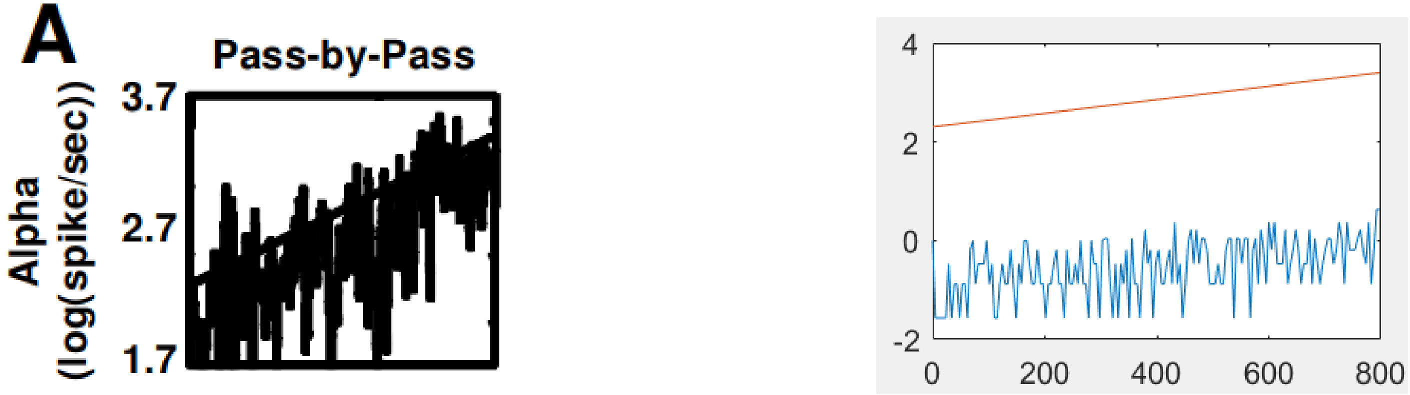


(a). The article.

(b). Simulation results.

SSSPF methods results for jump track

2. Different spike time data might be used.



(a). The article.

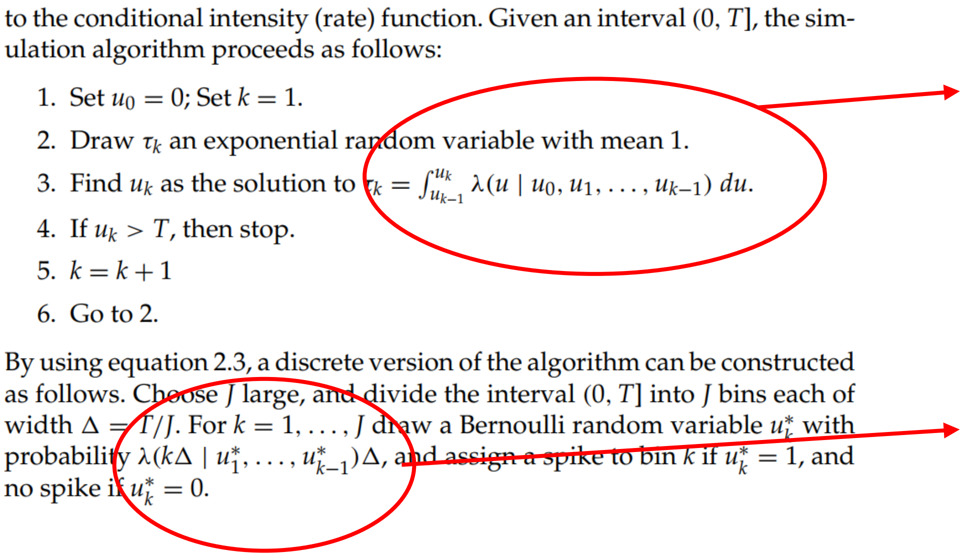
(b). Simulation results.

Pass-By-Pass methods results for linear track

2. Simulation Results Analysis

Experiment 1: Spatial Receptive Field Analysis

3. To verify the data generation code, 2 realizations are implemented.



to the conditional intensity (rate) function. Given an interval $(0, T]$, the simulation algorithm proceeds as follows:

1. Set $u_0 = 0$; Set $k = 1$.
2. Draw τ_k an exponential random variable with mean 1.
3. Find u_k as the solution to $\tau_k = \int_{u_{k-1}}^{u_k} \lambda(u \mid u_0, u_1, \ldots, u_{k-1})\, du$.
4. If $u_k > T$, then stop.
5. $k = k + 1$
6. Go to 2.

By using equation 2.3, a discrete version of the algorithm can be constructed as follows. Choose J large, and divide the interval $(0, T]$ into J bins each of width $\Delta = T/J$. For $k = 1, \ldots, J$ draw a Bernoulli random variable u_k^* with probability $\lambda(k\Delta \mid u_1^*, \ldots, u_{k-1}^*)\Delta$, and assign a spike to bin k if $u_k^* = 1$, and no spike if $u_k^* = 0$.

Implementation 1: solve the integral equations.

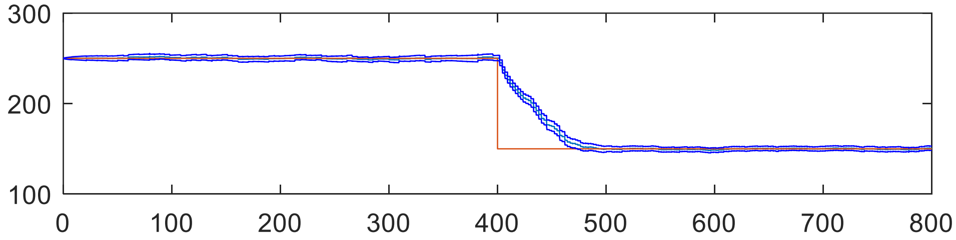
Implementation 2: discretized algorithms.

The results does not show much difference between these 2 implementations.

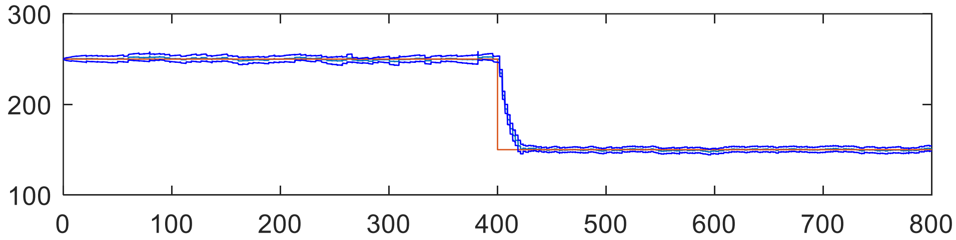
2. Simulation Results Analysis

Experiment 1: Spatial Receptive Field Analysis

4. Manipulations on observation intervals



Observation time: 20ms



Observation time: 8ms

- ➢ Faster tracking performance with smaller observation intervals.

Q&A