Speed Experiment Simulation Demo

09-18-2019

Fei Wang

Aim. The simulation aims to fit the neuron response elicited by stimulation by dividing them into slightly different stimuli. The difference can be site, direction of motion, distance between stimuli and participant or color and so on.

Background. The responses of stimulation are fitted by von Miser function, which is given by("von Miser distribution,"):

$$f(x|\mu,\kappa) = \frac{e^{\kappa \cos(x-\mu)}}{2\pi I_0(\kappa)} \tag{1}$$

where x is the motion direction of the unidirectional stimulus, μ is the direction where the tuning curve reaches its peak (the neuron's preferred direction), and $I_0(\kappa)$ is the modified Bessel function of order 0.

Suppose R_1, R_2 are the firing rates evoked by stimulation delivered by component 1 and component 2 of overall stimulation, respectively, the von Miser function can be simplified as:

$$R_1(\theta_1) = A_1 + B_1 e^{C_1 \cos(PD-\theta_1)} \tag{2}$$

$$R_2(\theta_2) = A_2 + B_2 e^{C_2 \cos(PD-\theta_2)} \tag{3}$$

where θ_1, θ_2 are the stimulation's direction of component 1 and 2, respectively, and PD is the preferred angle of the neuron. B and C determine the magnitude and bandwidth of the tuning curve, respectively, and A is a constant. The indexes of parameters A, B, C indicate different segmentations of stimulation.

The model constructed to fit the overall response of combined stimulation added a nonlinear interaction to the weighted linear sum model, which is referred to as the summation plus nonlinear interaction (SNL) model. Assume that R_{12} is the firing rate of the overall stimulation, then the model will be given by (Vokoun, Huang, Jackson, & Basso, 2014):

$$R_{12}(\theta_1,\theta_2) = \omega_1 R_1(\theta_1) + \omega_2 R_2(\theta_2) + bR_1(\theta_1)R_2(\theta_2) + c \tag{4}$$

where ω_1, ω_2 are response weights and b represents the weight for nonlinear interaction. The constant c is always set to zero in this simulation.

In the simulation, R_1 and R_2 are generated by users' input parameters. Gaussian noise is applied to the firing rates, whose mean and standard deviation can also be determined by users.

User Interfere. *Simulation* is a Graphical User Interface (GUI) within which simulation can be defined, executed, and monitored. The interface can be divided into three parts: *Parameters Settings*, *Figure Plotting*, *Results and Save*. *Parameters Setting* area is dedicated to editing a series of relevant parameters to determine the simulation. *Figure Plotting* area plots the tuning curves whose y-axis represents firing rate. In this area, users are also able to choose some display alternatives. *Results and Save* area display the values and verifications of the parameters and data. After "Save Data" button is pressed, results will be saved in MATLAB document, titled by the filename provided by users. "Save Figure" button save the axes in Portable Network Graphic (.PNG), which is the only format it supports for the time being.

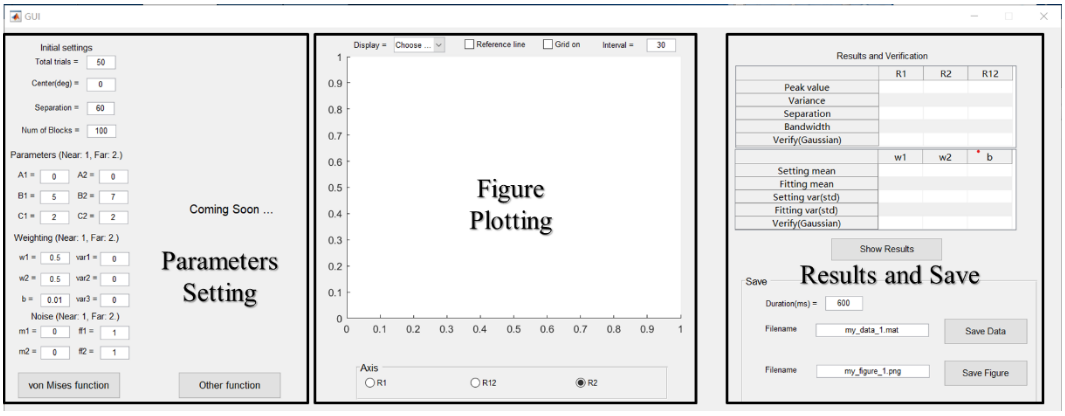


Figure 1. User Interfere

Parameters Settings. In this area, users can edit Initial Settings, Parameters for von Miser function, Weighting Parameters Generation and Noise Generation.

For Initial Settings:

- Total trials: represents the number of duplicated trials. In each trial, the same stimuli will be applied to eliminate random errors;
- Center (degree): the preferred angle of neuron;
- Separation: the angular separation between θ_1 and θ_2;
- Num of blocks: one block concludes trials set by Total trials. This value will be used to verify whether the generation of ω_1, ω_2 and b conform to Gaussian function. Note that large value may cause delay in plotting and showing results, and small value may cause mistakes in verification due to the limited samples.

For Parameters for von Miser function:

- A1, A2: constants which are positive in most cases. The default value is set to zero. Note that 1, 2 means different component of the stimulation, same as below.
- B1, B2: parameters which determine the magnitude of the tuning curves.
- C1, C2: parameters which determine the bandwidth of the tuning curves. Note that C1, C2 also influence the magnitude.

For Weighting Parameters Generation:

- w1, std1: mean value and standard deviation of weight of component 1 to be generated, complied with Gaussian function.
- w2, std2: mean value and standard deviation of weight of component 2 to be generated, complied with Gaussian function.
- b, stdb: mean value and standard deviation of weight of the nonlinear interaction of two components to be generated, complied with Gaussian function.

For Noise Generation:

- mean1, mean2: mean value of noise in generation of component 1 and 2, respectively, complied with Gaussian function.
- ff1, ff2: coefficients of variation of noise in generation of component 1 and 2, respectively, complied with Gaussian function.

If users do not edit any parameters, the simulation will be determined by the default values shown in the Interfere, as Figure 2 shows.

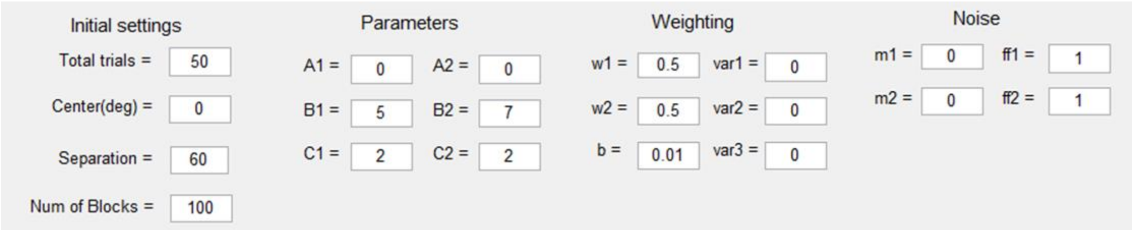


Figure 2. Parameters Setting.

Figure Plotting. Press the button “von Miser function” to plot the figure.

There are several options for the figure plotting:

- Display Mode: There are four modes to display. In Mode 1, tuning curve of R1 is always on the left, while Mode 2 is exactly the opposite. In Mode 3 and 4, it will show the single tuning curve of R1. Note that the default mode is Mode 1.
- Reference line: the tuning curve when w1, w2 equal 0.5 and b equals zero.
- Grid: show the grid of figure.
- Axis: choose whose axis to show.

Note that in default condition, x-axis will always be from -180 to 180 degree whose interval is 30 degree, regardless of the neuron’s preferred degree. If users need to view the actual x-axis of any curve, please press the corresponding button. The interface is shown as Figure 3.

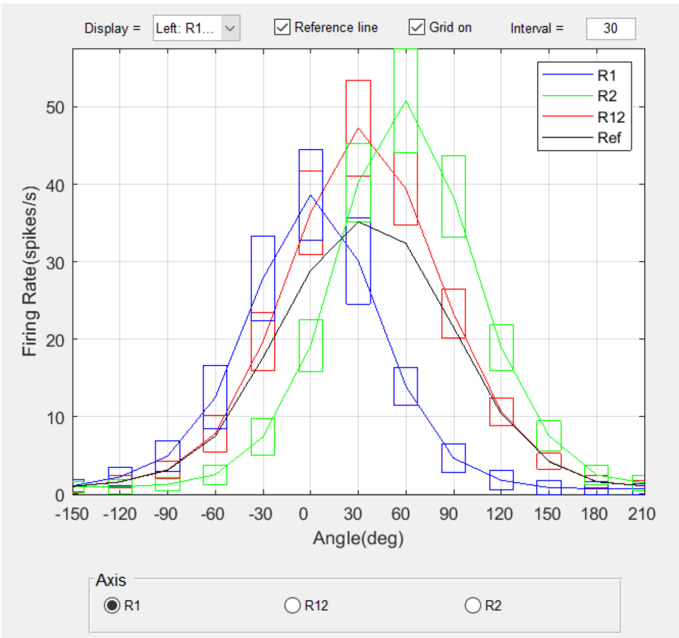
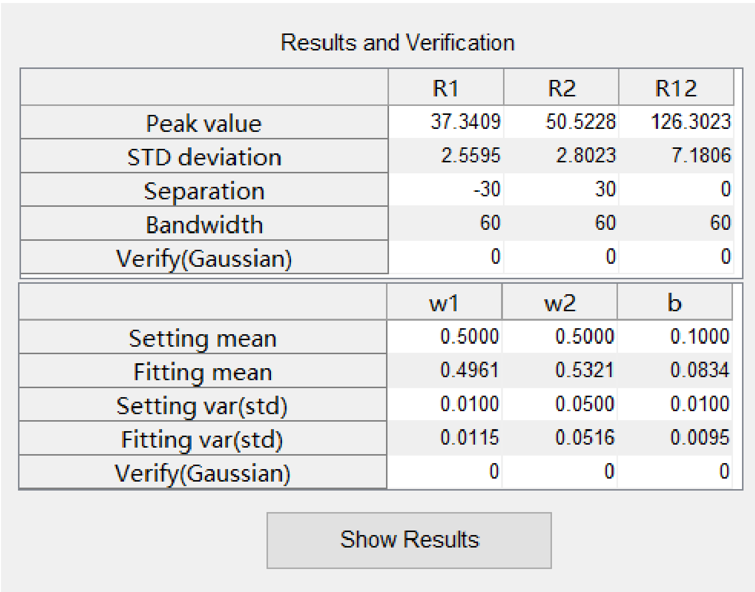


Figure 3. Figure Plotting

Results and Save. Press "Show Results" button to show results and verification, as Figure 4 shows. Note that if the generation conform to Gaussian distribution, the related value in Verify (Gaussian) will be zero, otherwise one. In Save interface, name of file should be text into the bar. The duration is used to calculate the number of spikes to compare with the real data, which will be saved in the same MATLAB document with simulation results. After press the "Save Data" button and "Save Figure" button, a MATLAB document and a PNG figure can be saved, as Figure 5 shows. Note that the filename won't display when choosing the save path of "Save Data", and you can type any key to go on the procession.


Results and Verification

	R1	R2	R12
Peak value	37.3409	50.5228	126.3023
STD deviation	2.5595	2.8023	7.1806
Separation	-30	30	0
Bandwidth	60	60	60
Verify(Gaussian)	0	0	0

	w1	w2	b
Setting mean	0.5000	0.5000	0.1000
Fitting mean	0.4961	0.5321	0.0834
Setting var(std)	0.0100	0.0500	0.0100
Fitting var(std)	0.0115	0.0516	0.0095
Verify(Gaussian)	0	0	0

Show Results

Figure 4. Results and Verification

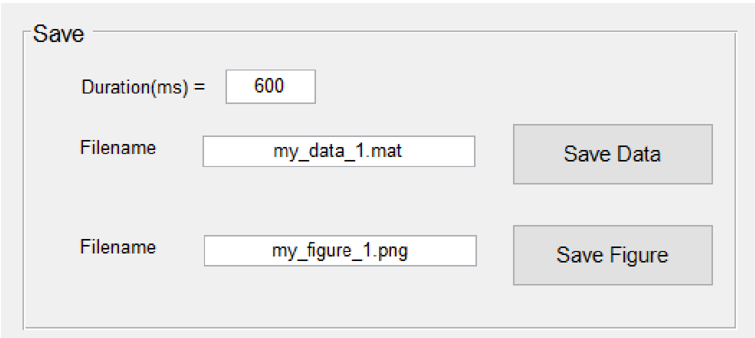


Figure 5. Save Data and Figure

Edit Instruction. If you want to edit the interface, first type "guide" in MATLAB command, then open existed file "GUI.fig". In this MATLAB provided interface, you are able to add new functions in *Simulation*.

Reference

Vokoun, C. R., Huang, X., Jackson, M. B., & Basso, M. A. Response Normalization in the Superficial Layers of the Superior Colliculus as a Possible Mechanism for Saccadic Averaging. *The Journal of Neuroscience*, *34*(23), 7976-7987. doi:10.1523/jneurosci.3022-13.2014.