Models of Higher Brain Function
Computer Course
Week 4

Lecturer: Henning Sprekeler — Assistant: Owen Mackwood

Perceptual Bistability

May 19, 2016

The solutions for these exercises (comprising source code, discussion and interpretation as an IPython Notebook) should be handed in before **May 26 at 8am** through the Moodle interface (in emergency cases send them to owen.mackwood@bccn-berlin.de). The solutions will be discussed in the computer course on Monday, May 30.

Exercise 1

In this exercise we will explore a model of binocular rivalry presented by Laing and Chow (2002). While their paper included a spiking network model, we will concern ourselves only with their reduced population rate model.

1. Write a function *laing_chow* which implements eqs. (1-4) from Laing and Chow (2002) and which can be given to *scipy.integrate.odeint* as its first argument. The parameters of the model should be additional arguments of *laing_chow* such that you can investigate the behavior of the model for different parameter settings. Reproduce qualitatively Fig. 9 of Laing and Chow (2002) by integrating the differential equation defined by *laing_chow* using *odeint(laing_chow, x0, t, args=(I1, I2))* where *x0* should contain the initial values $u_1 = 1,\ a_1 = 0.1\, u_2 = 0,\ a_2 = 0.25$, t should range from 0 to 500, I are the inputs with $I_1 = 0.43,\ I_2 = 0.5$ and all other parameters of the model should be set by default to $\alpha = 0.2,\ \beta = 0.4,\ \phi = 0.4,\ \tau = 20,\ g_1 = g_2 = 1$ in *laing_chow*.

2. Let us define population 1 as active if $u_1 > u_2$ and equivalently for population 2. Determine the dominance durations D_1 during which population 1 is active, from your simulations. To do this write a function *dominance_durations(x1, x2, t)* where *x1* and *x2* are arrays containing the simulated states and t is an array of the same length containing the corresponding times. The result should be an array containing the duration of each dominance period in the data. Why are the dominance durations different from the analytical T_1 (eq. 9) in the paper?

3. The paper states that, if the inputs are larger than $-\alpha + \beta + \phi$, both populations become active and oscillation will stop. Can you confirm this? What about intermediate input values? Do the analytical dominance periods still fit the experimental results? How do you interpret these results?

Exercise 2

Next we will investigate a model of perceptual bistability developed by Moreno-Bote et al. (2007). The model is an attractor network that relies on noise to drive transitions between network states. Once again, while they also present a spiking network model, we will consider only their mean-field firing rate model.

1. Why can't you use *scipy.integrate.odeint* to implement the model in Moreno-Bote et al. (2007)?

2. You will have to write your own integration function. We will use the Euler-Maruyama method. Euler-Maruyama is an extension of the simple Euler method to stochastic differential equations where we here only consider Gaussian noise processes. Euler-Maruyama integrates differential equations of the form

$$dX(t) = f(X(t))dt + g(X(t))dW(t)$$

here $f(X(t))$ is the deterministic part of the differential equation and $g(X(t))dW(t)$ is the stochastic part where $dW(t)$ is a random variable. Euler-Maruyama consists of the following update which your function has to implement:

$$X_t = X_{t-1} + f(X_{t-1})\Delta t + g(X_{t-1})\sqrt{\Delta t}W_t$$

Each W_t is a sample from a standard normal distribution. For more information about Euler-Maruyama see Higham (2001), or Wikipedia.

Write a function *euler_maruyama(ffun, gfun, x0, t, **args)* which implements this equation and which behaves like *odeint*. Note that we have defined args as a dictionary which differs from *odeint* where args is a tuple. This makes it easier to change individual parameters of the model functions later. Test your function by using *ffun=laing_chow* and *gfun=0* (hint: you can implement *gfun* compactly using Python's lambda notation). For small enough time steps ($\Delta t = 0.1$ should be sufficient) you should get the same results as with *odeint*.

3. Make *ffun* return 0 and *gfun* return $1/\tau$. Simulate this stochastic differential equation between $t = 0$ and $t = 5$ with $\Delta t = 0.01$ for $\tau = 0.1$ and for $\tau = 1$. Further, simulate the stochastic differential equation with *gfun* as before and *ffun*$=-x/\tau$. Compare the results of the three simulations. What is similar? What is different? (hint: look up Wiener processes and Ornstein-Uhlenbeck processes. Your answer should incorporate these terms.)

4. Finally, write functions *f_moreno_bote* and *g_moreno_bote* which implement equations (A1, B5, B6, B7) of Moreno-Bote et al. (2007). (hint: note the stochastic part of the equations: only that goes into *g_moreno_bote*.) Simulate for t from 0 to 20 with $\Delta t = 0.01$ time units for the model using the same parameter settings as in the paper: $g_A = g_B = 0.05$, $\alpha = 0.75$, $\beta = 0.5$, $\gamma = 0.1$, $\phi = 0.5$, $\tau = 0.01$, $\tau_a = \tau_b = 2$, $\tau_s = 0.1$, $\sigma = 0.03$, $\eta = 0.5$, $\theta = 0.1$, $k = 0.05$ and initial conditions $r_A = 1$, $r_B = 0.01$, $\alpha_A = 0$, $\alpha_B = 0.04$, $n_A = n_B = 0$. Visualize the results using three subplots: rates, hyperpolarizing currents and noise.

5. Now simulate for t up to 500 (again with $\Delta t = 0.01$). Determine the dominance durations (if you implemented it correctly, you can simply use function *dominance_durations* from exercise 1). Plot a histogram of the dominance durations. What is their mean?

6. What is the influence of γ in the model? How and why does the histogram change if you choose $\gamma = 0$? Repeat your simulation with $\gamma = 0$ and estimate the mean dominance duration.

7. Try to find a setting of parameters with $\gamma = 0$ which has a mean dominance duration close (within 0.2 time units) to the first simulation from exercise 2.5. Show all three duration distributions in one histogram plot. How do they differ?