Architectures of Intelligence
Assignment 4/Part2

Group 13
Mirko Maragaira & Matteo Wohlrapp

December 15, 2021

Introduction to Nengo

Values

Figure 1 shows a plot of the value of ensemble c. The blue line represents the ideal value, whereas the green graph shows the actual value, derived with 50 neurons. The amplitude and period are quite similar, but the graph is shifted slightly to the right.

Functions

In the following section, we will explain three functions of the model in more detail.

RUN_NEURONS

run_neurons basically runs the neuron model. That means that for every neuron in the ensemble it computes if it fires. As inputs, it takes a list of ints, where the int at index i describes the input for the i^{th} neuron. The values given are comprised of the input to the ensemble, the gain and the bias. It also receives a list of floats, where the i^{th} entry is the voltage for the i^{th} neuron. Furthermore, a list of floats with the refractory values is passed to the function, describing how long each neuron can't fire. The function then iterates over every neuron, calculates the delta voltage and adds it to the current-voltage of the neuron. Afterwards, if the neuron is not in the refractory period, the voltage is checked and if it exceeds a threshold, which indicates the neuron has fired, true is added to a list and the voltage is reset, otherwise, the neuron is not considered as fired and false is added. This list of boolean, which holds information about which neurons fired is then the output of the function.

COMPUTE_RESPONSE

compute_response measures the spiking rate of an ensemble for a fixed x value and returns it in a list of floats with the rate measured in hertz. As input parameters, the function takes the value x, an encoder for the ensemble, gain, bias and the time limit which specifies the duration of computing the spikes in the neurons. First, the input is calculated based on the gain, bias, the x value and the specified encoder, then the voltage is initialized randomly. Afterwards, the function iterates over a certain time, in each iteration calculating the spikes with the *run_neurons* function described above. If one neuron spikes, the spike count of that neuron is increased. In the end, before returning the count, it is divided by the time to convert to hertz.

COMPUTE__TUNING__CURVES

The function computes the in part one of the assignment described tuning curves. That is, it iterates over a range of x values and calculates the number of spikes of each neuron for that value. To do this, the above described *compute_response* function is used. The resulting list of spike rates is appended to a matrix and later used to plot the curve. The function takes the encoder of the ensemble, the gain and the bias as the input and returns the before mentioned matrix of spike rates and the corresponding x values.

Bonus

Figure 2 shows a plot of the value of ensemble d, where dimension 0 is connected to the stimulus and dimension 1 is connected to ensemble c. The blue graph shows the stimulus, which is a sin-function, the green and red the ideal and actual value of ensemble d in dimension 1 according to ensemble c. When comparing **Figure 2** to **Figure 1**, one can see the similarity between the graph of C in **Figure 1** and the graph of D in **Figure 2**. The amplitude, as well as the period, are nearly identical, they only appear different because the ranges of the two graphs differ.

A. Assignment 4/Part 2

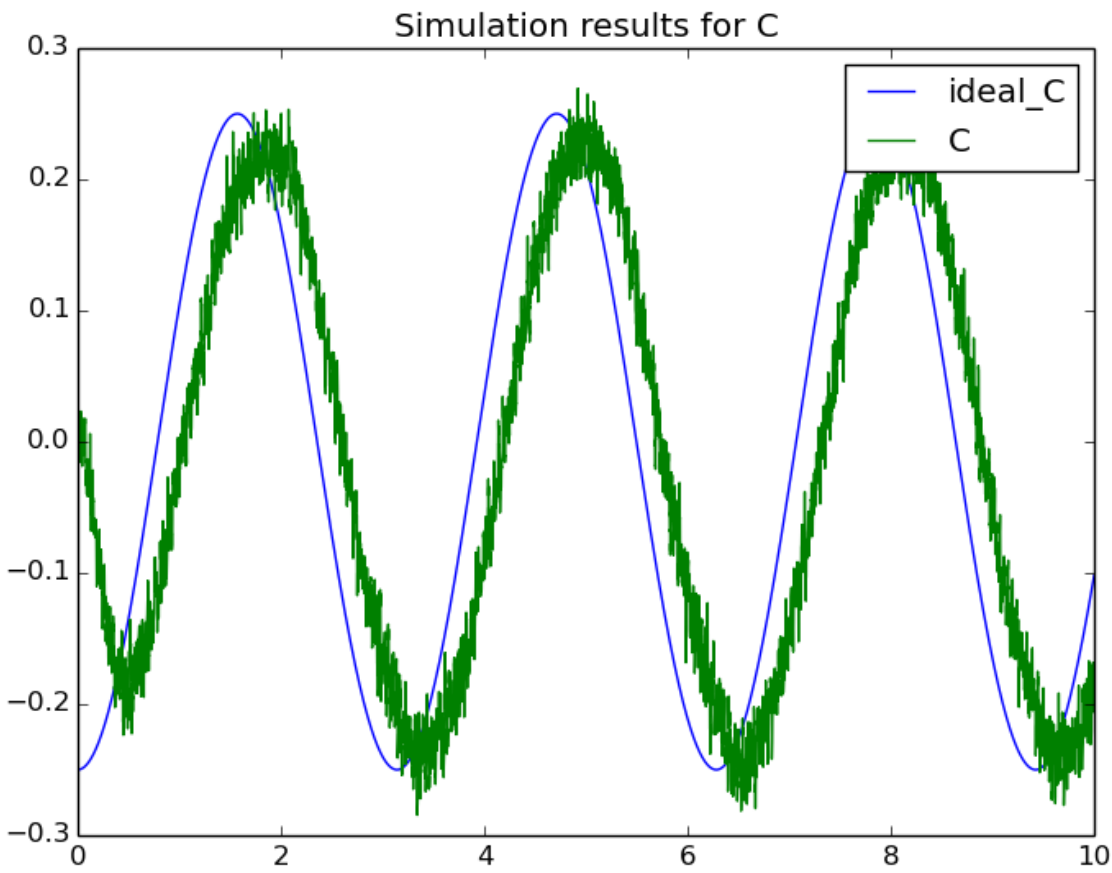

Figure 1: *Plot of the value of ensemble c.*

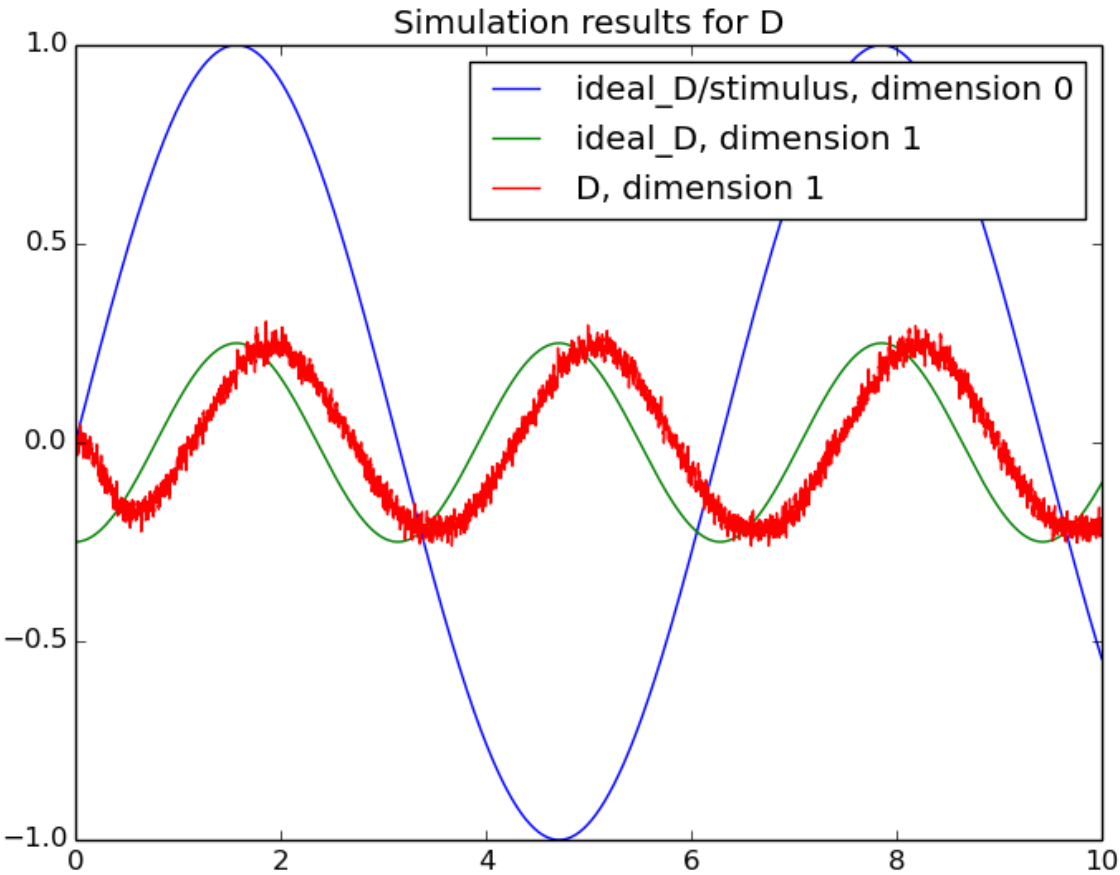

Figure 2: *Plot of the value of ensemble d, where dimension* 0 *is connected to the stimulus and dimension* 1 *is connected to ensemble c*