Student names: . . . (please update)

Instructions: Update this file (or recreate a similar one, e.g. in Word) to prepare your answers to the questions. Feel free to add text, equations and figures as needed. Hand-written notes, e.g. for the development of equations, can also be included e.g. as pictures (from your cell phone or from a scanner). ***This lab is not graded. However, the lab exercises are meant as a way to familiarise with dynamical systems and to study them using Python to prepare you for the final project.*** *This file does not need to be submitted and is provided for your own benefit. The graded exercises will have a similar format.*

The file `lab#.py` is provided to run all exercises in Python. When a file is run, message logs will be printed to indicate information such as what is currently being run and and what is left to be implemented. All warning messages are only present to guide you in the implementation, and can be deleted whenever the corresponding code has been implemented correctly.

In this lab, you will be modifying `lab4.py`, `network.py` and `simulation_parameters.py`.

Coupled firing rate neurons to model a swimming CPG segment

The lab of today is based on networks of single/coupled firing rate neuron models with adaptation that describes the coarse-grained activity of a population of neurons. You will use similar models to build a controller in the zebrafish project. These models that were studied in the following papers:

- *Seely, J., & Chow, C. C. (2011). Role of mutual inhibition in binocular rivalry. Journal of neurophysiology, 106(5), 2136-2150.*
- *Curtu, R., Shpiro, A., Rubin, N., & Rinzel, J. (2008). Mechanisms for frequency control in neuronal competition models. SIAM journal on applied dynamical systems, 7(2), 609-649.*

This model can be used to study the generation of the swimming rhythm by the CPGs in fishes, such as the lamprey or the zebrafish, and the dependence on several biophysical parameters, such as input drive from the brain, excitatory and inhibitory weights, etc.

Model equations

The equations of a single firing rate unit modeling a single hemisegment oscillator that you will need to implement is described by the following pair of equations

$$\begin{aligned} \tau\dot{r} &= -r + F(I + \alpha r - ba) \\ \tau_a\dot{a} &= -a + r, \end{aligned} \tag{1}$$

where r represents the firing rate of the neuron, and a the firing rate adaptation (fatigue). The biophysical meaning of the parameters in system 1 is reported in Table 1, respectively. We assume that the system operates in the slow-fast regime, i.e. τ much smaller than τ_a. This means that the neuron's spiking dynamics is faster than that of the firing adaptation, which is what is typically observed experimentally.

The function $F = F(x)$ in equation 1 is called gain function (also known as activation function, or transfer function), and it is the input-output non-linearity capturing the neural properties of the system. Here we will consider and study the role of three different types of gain functions (see Figure 1). We will consider a sigmoid gain

$$F(x) = F_{\lambda,\theta}(x) = \frac{1}{1 + e^{\lambda(\theta - x)}}, \tag{2}$$

a rectified linear unit (ReLu) gain

$$F(x) = max(x, 0), \tag{3}$$

Parameter	Meaning
$\tau > 0$	Neuron timescale
$\tau_a > 0$	Adaptation timescale
$b > 0$	Adaptation strength
$\alpha > 0$	Self-excitation strength
$I > 0$	Input current (i.e. from higher brain centers)
w	Coupling strength ($w > 0$=inhibitory connection, $w < 0$=excitatory connection)

Table 1: Biophysical meaning of the parameters of the firing rate network

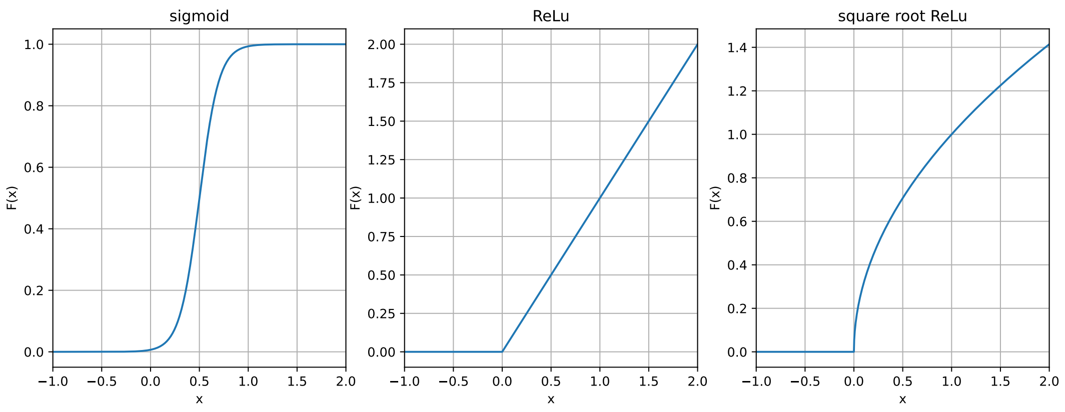

Figure 1: Gain function profiles.

and the square ReLu gain

$$F = \sqrt{max(x,0)}. \tag{4}$$

The equations of a couple of connected firing rate units the following equations (similar to equation 1 but with the addition of a connectivity term with strength w)

$$\begin{aligned} \tau \cdot \dot{r}_1 &= -r_1 + S(I + \alpha r_1 - w r_2 - b a_1) \\ \tau \cdot \dot{r}_2 &= -r_2 + S(I + \alpha r_2 - w r_1 - b a_2) \\ \tau_a \cdot \dot{a}_1 &= -a_1 + r_1 \\ \tau_a \cdot \dot{a}_2 &= -a_2 + r_2. \end{aligned} \tag{5}$$

This crossed coupling term represents the commissural projections found in swimming animals.

Metrics

To evaluate the model we provided your the following three metrics of the neural network (stored in the dictionary `network.py::FiringRateController().metrics` and computed in `util/metrics.py`):

- **frequency**: returns the frequency of the first element of the state array `network.py::FiringRateController().state[:,0]`. Use it to compute the frequency of the firing rate variable r of systems 1 and 5. Internally, the code computed the frequency from the last two threshold crossings of `network.py::FiringRateController().state[:,0]`.
- **amp**: return the amplitude of `network.py::FiringRateController().state[:,0]`
- **sync**: computes a mesure of synchronization for the coupled units (only valid for system 5). It has values between 0 and 1, where 0/1=inphase synchronization and 0.5=antiphase synchronization.

Numerical integration

In all simulations, we consider only steady-state dynamics (we exclude transient dynamics) and starting from uniformly random initial conditions in all the system's variables. Simulations can be run using two options:

- `util.run_open_loop.py::run_simulation(pars)` - runs a single simulation with simulation parameters specified by the `simulation_parameters.py::SimulationParameters()` class.
- `util.run_open_loop.py::run_multiple(pars_list)` - runs multiple simulation in parallel using your computer cores from a list of `simulation_parameters.py::SimulationParameters()` classes. Use this function to speed up computations in parameter searches.

Exercises

4.a Implement the equations of a single firing rate neural controller unit with sigmoid gain function and adaptation (equation 1 and equation 2). This network represents the central pattern generator in a single hemisegment in a fish or lamprey (as seen in the lectures). Run systematic simulations when varying the self-excitation strength $\alpha \in [0, 3]$. How is the behavior of the single unit changing? Is this in line with what we know from the lamprey/fish literature?

Hemisegment oscillations: If the self-excitation is sufficiently high a single unit with adaptation can generate self-sustained oscillations (Figure 2). Similarly, research on lamprey and other fish(and tadpoles) demonstrated that single hemisegments can generates rhythmic oscillations with constant input levels.

4.b Implement the equations for the coupled firing rate controllers 5 with sigmoid gain function, no self-excitation and $w = 2$ (low inhibition). What does the output looks like? Run a systematic parameter sweep varying the input $I \in [0, 10]$. What dynamical behavior can the model generate for different values of I?

coupled units: see Figure 3

4.c Repeat the same study as in Exercise 4.b (same parameters and specifications as in Exercise 4.b) for higher levels of inhibition $w = 3.8$. Do new states emerge when varying $I \in [0, 10]$ as in exercise 4b?

coupled units: see Figure 4

4.d Repeat the same study as in Exercise 4.b/c replacing the sigmoid gain by a ReLU function (equation 3) and by the square root of the ReLu gain (equation 4). Discuss the simlarities/differences with these previous cases.

coupled units: see Figure 5 and Figure 6

4.e Now study the role of excitatory and inhibitory coupling. What happens to the pair if the weight are now excitatory ($w < 0$)? Run a parameter sweep of $w \in [-2, 5]$ Use the sync metric to describe your findings. Experiments revealed the existence of crossed inhibitory connections are dominant for explaining the swimming rhythm. Do your results corroborate these findings?

coupled units: see Figure 7

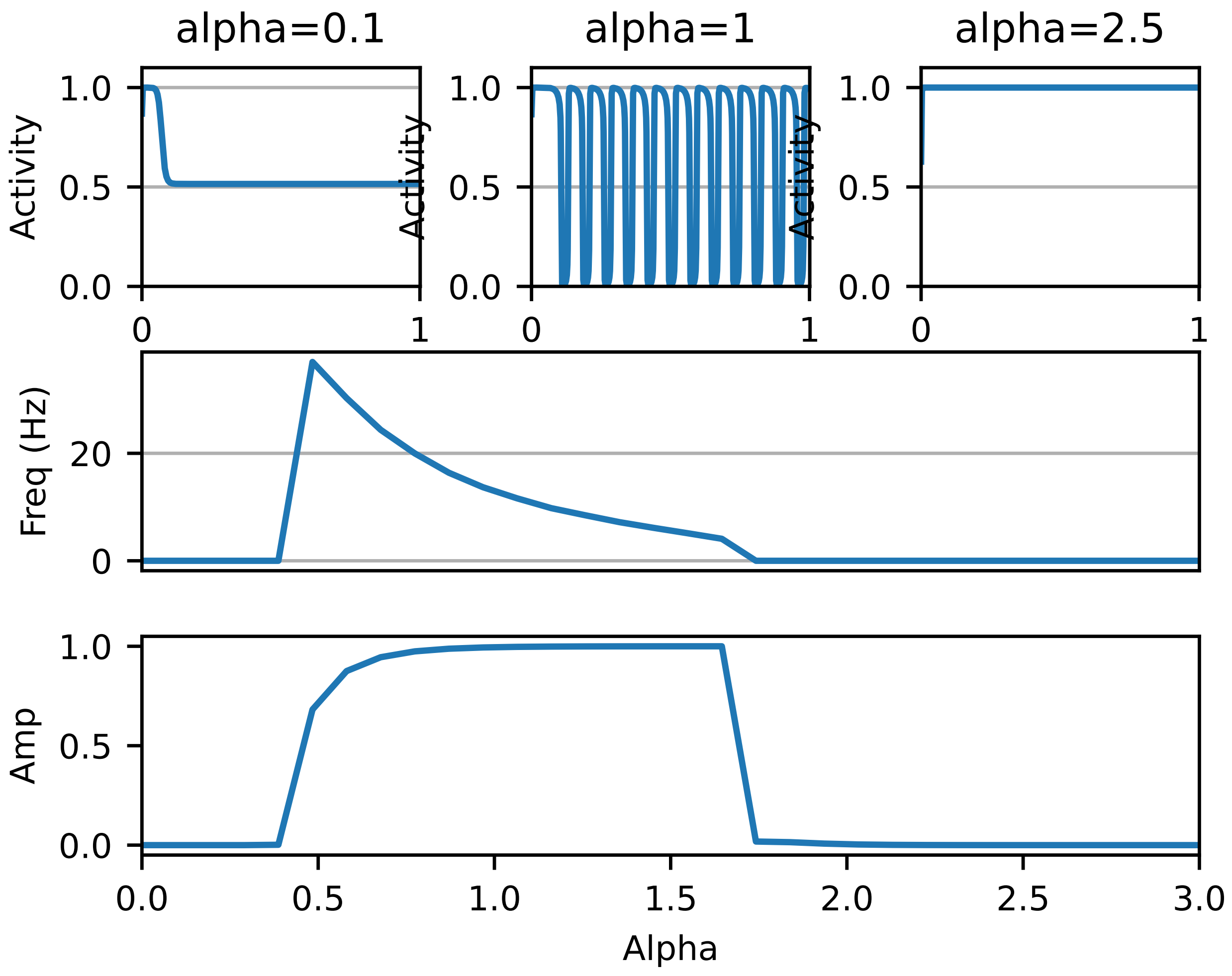

Figure 2: Single unit. With no self-excitation ($\alpha = 0$) a single firing rate unit cannot generate oscillations. With higher α values the network starts to spontanously oscillate. Interestingly, the frequency decreases with α. The top three panels show the unit's activity rate r at selected values of α. The bottom panels show measures at varying α.

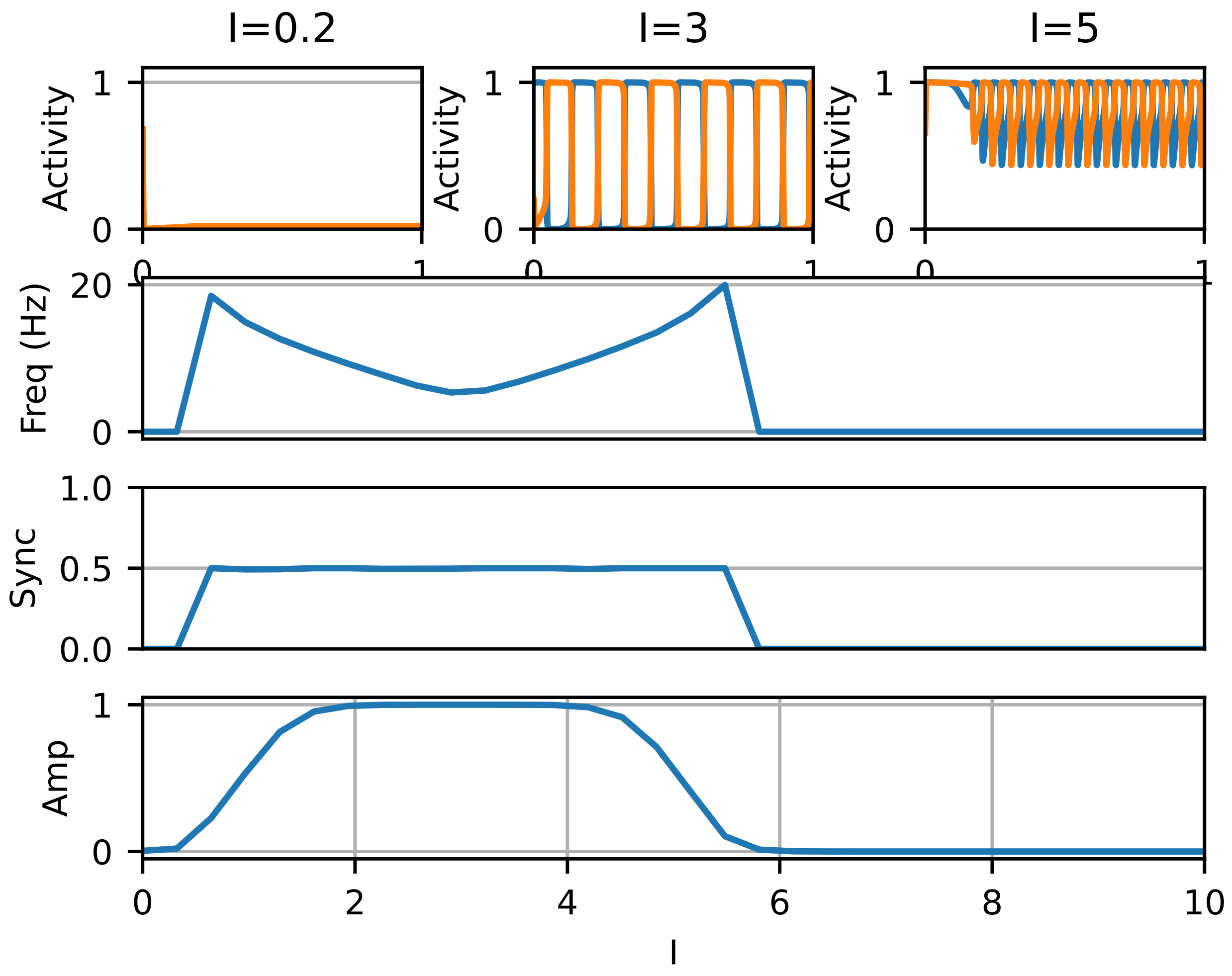

Figure 3: Mutually inhibitory units with sigmoidal gain - low inhibition. The top three panels show the activities of the two units (blue and orange curves). The bottom panels show the metrics at varying inputs I. For low inputs the units both units are at low activity (below sigmoid threshold). At intermediate inputs the network generates oscillations in alternating fashion (sync ~ 0.5*). At high inputs the oscillations are replaces by high saturated activity. Note the nonmonotic input-frequency relationship: once oscillations are established the frequency first decreases and then increases, accoriding to mechanisms of release and escape behaviors as explained in Reference 1.*

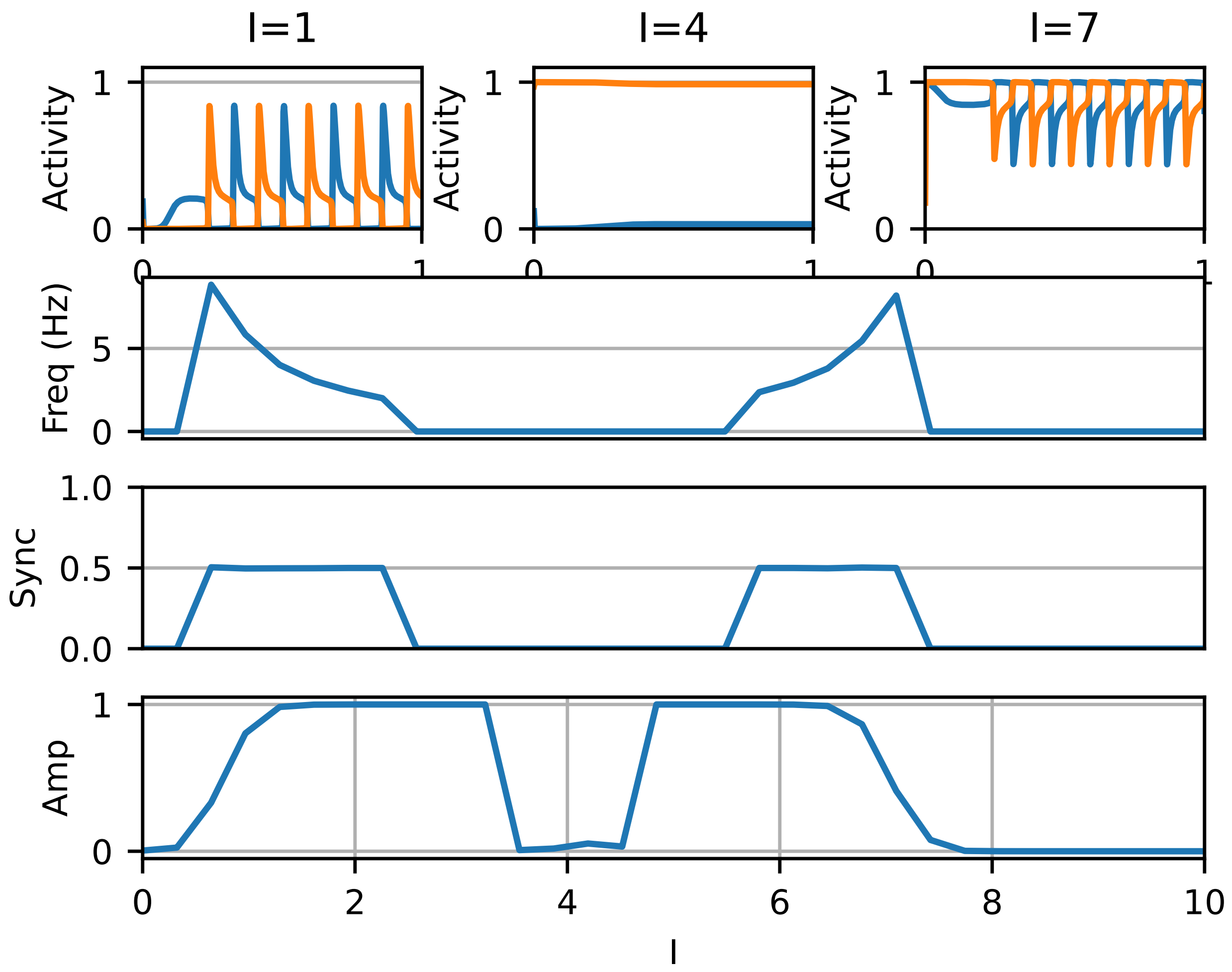

Figure 4: Mutually inhibitory units with sigmoidal gain - high inhibition. The plots are as in Figure 2 but for higher values of inhibition $w = 3.8$. A new state (winner-take-all) appears, where one unit is active while the other is suppressed and cannot recover from the suppressed state (i.e. for $I = 4$).

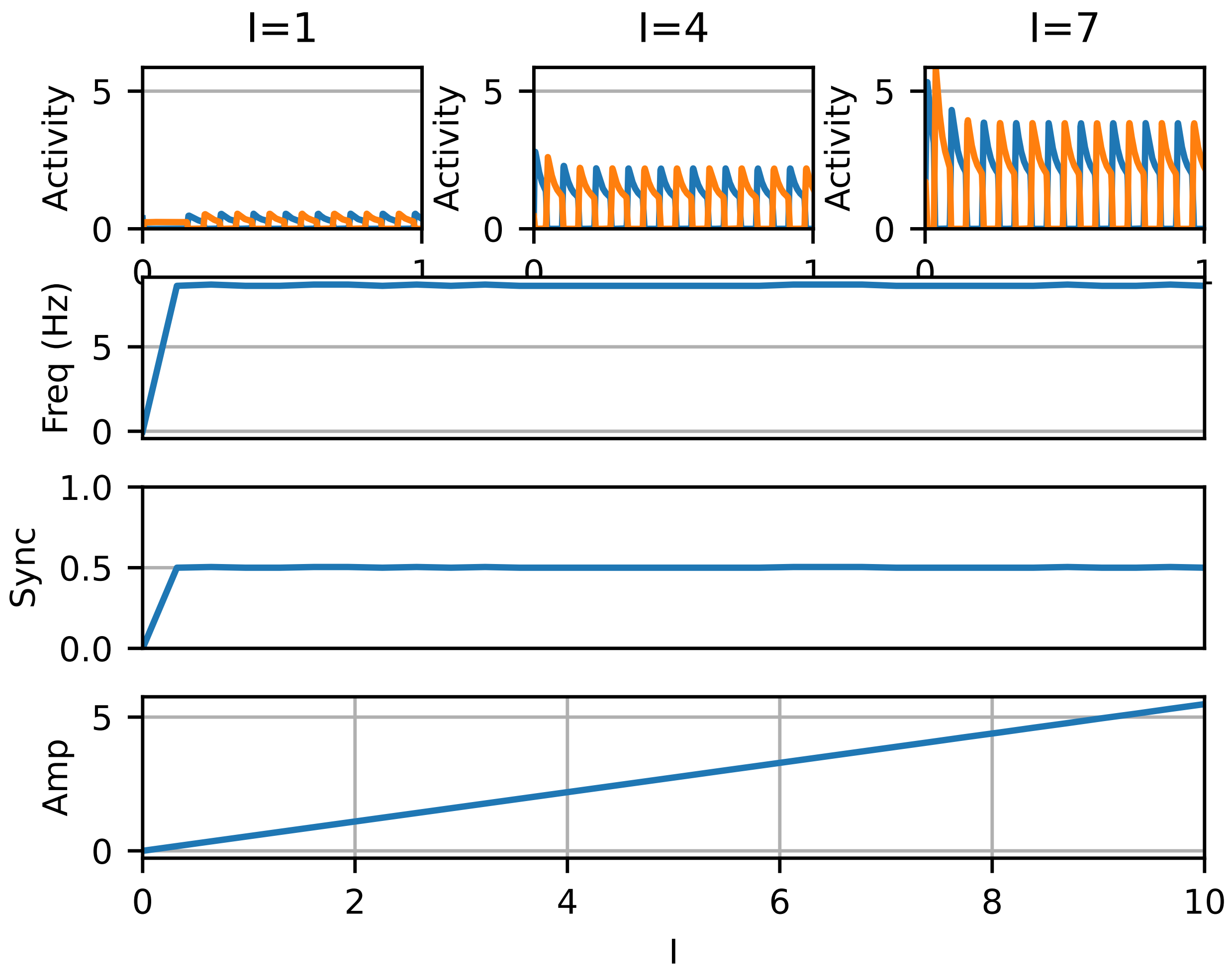

Figure 5: Mutually inhibitory units with ReLU gain. The network now generates only oscillations at all inputs. The frequency is constant, while the amplitude grows linearly, as expected from the linearity of the gain function.

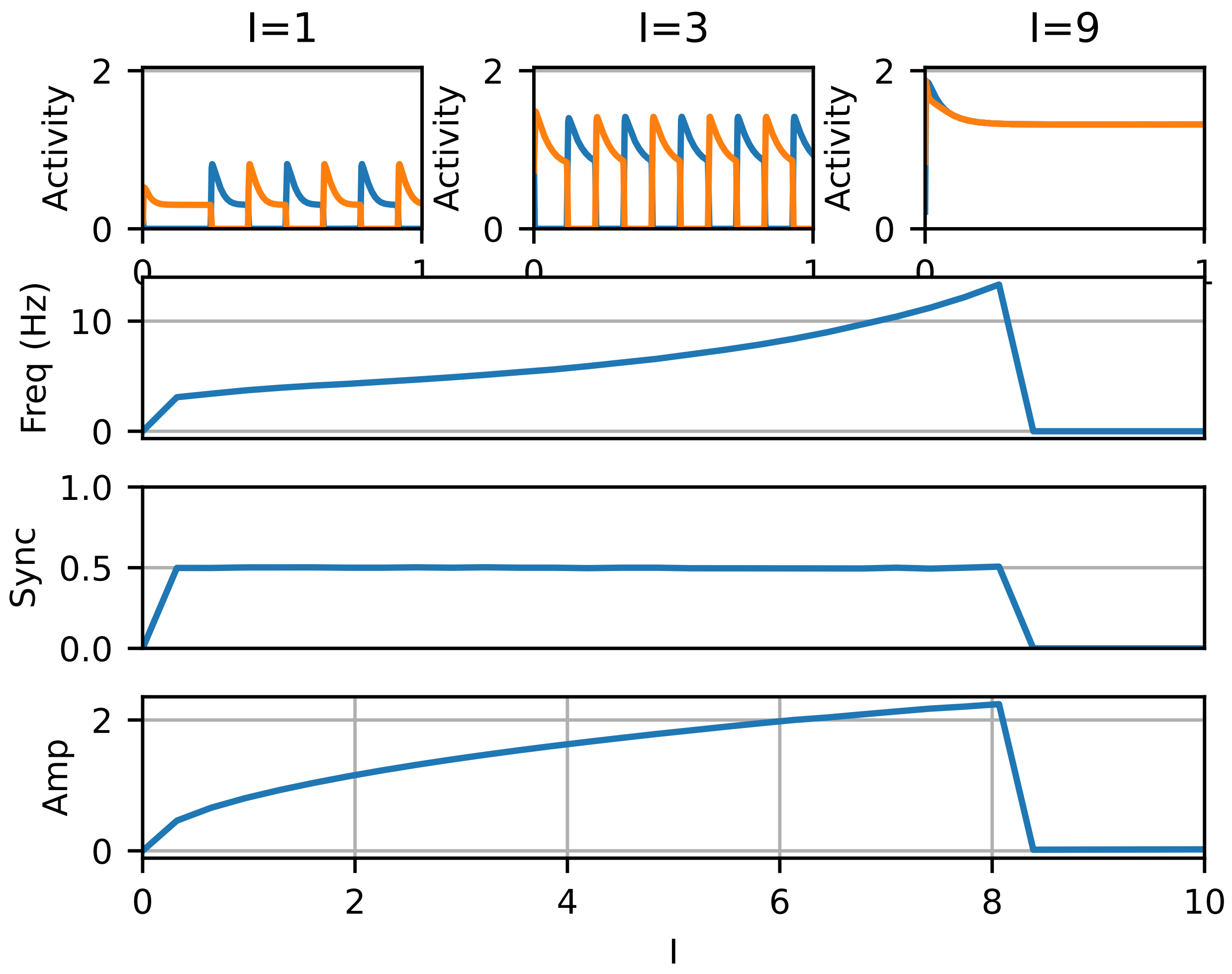

Figure 6: Mutually inhibitory units with square root ReLU gain. The network now generates a monotonically increasing current-frequency relationship, which is in line with experiments (check lectures).

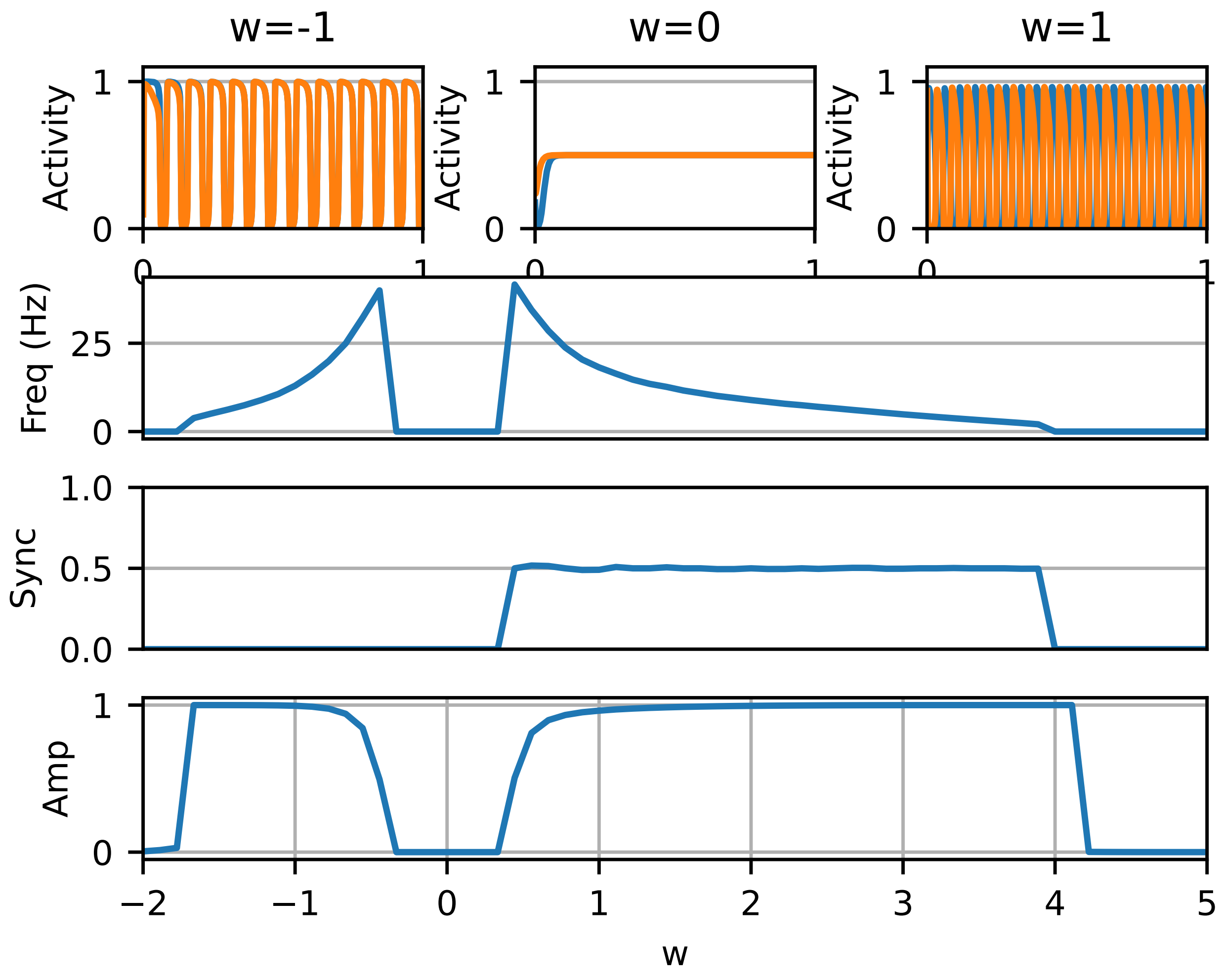

Figure 7: Study the synchronization at varying coupling weight w. For positive weights (inhibition) the network produces alternation, while for negative weights (excitation) the network produces in-phase oscillations. For extreme (high) values of inhibition the network generates a winner-take-all regime. Low w values generate no oscillations (no self-excitation).