Dynamical Systems and Neuroscience

Mario Roca
mario.roca@etu-upsaclay.fr

October 2024

TP1 Binocular Rivalry

Binocular rivalry is a remarkable phenomenon of vision. When two dissimilar images are presented simultaneously to each eye, perception alternates between them. For example, if the left eye views horizontal lines and the right eye views vertical lines, the observer will report seeing vertical lines for a few seconds, then horizontal ones, then vertical ones again, and so on.

Binocular rivalry is a popular tool for studying perception and awareness, because perception changes even though the physical stimulus does not. The rate at which perception alternates between the two images and the relative duration of each images during a perception cycle depend in particular on the intensity (for instance, contrast) of the two images. These dependencies have been formalized by four Levelt's laws. In his paper H.R. Wilson proposed a simple population model for binocular rivalry, made of four differential equations:

$$\tau \dot{E}_L(t) = -E_L(t) + mS(L(t) - aE_R(t) + \epsilon E_L(t) - gH_L(t)) \tag{1a}$$
$$\tau_H \dot{H}_L(t) = -H_L(t) + E_L(t) \tag{1b}$$
$$\tau \dot{E}_R(t) = -E_R(t) + mS(R(t) - aE_L(t) + \epsilon E_R(t) - gH_R(t)) \tag{1c}$$
$$\tau_H \dot{H}_R(t) = -H_R(t) + E_R(t) \tag{1d}$$

Where:

- $L(t) \in [0, 1]$ is the intensity of the stimulus on the left eye.
- $R(t) \in [0, 1]$ is the intensity of the stimulus on the right eye.
- $E_L(t) \geq 0$ is the neuronal activity driven by $L(t)$.
- $E_R(t) \geq 0$ is the neuronal activity driven by $R(t)$.
- $a > 0$ is an inhibitory synaptic weight.
- $\epsilon, g > 0$ are excitatory synaptic weights.

- $m > 0$ describes the excitability of each neuronal population.
- $H_L(t), H_R(t) \geq 0$ represent slow hyperpolarizing currents.
- $\tau, \tau_H > 0$ are time constants.
- S is the activation function defined as $S(x) = \max(0, x)$ for all $x \in \mathbb{R}$.

Question 1 - Justify non-negativity

Justify the fact that, for non-negative initial states $(E_{L0}, H_{L0}, E_{R0}, H_{R0})$, it holds that $E_L(t) \geq 0$, $H_L(t) \geq 0$, $E_R(t) \geq 0$, $H_R(t) \geq 0$ at all times $t \geq 0$.

---

a) Analysis of $E_L(t)$

The equation for $E_L(t)$ is:

$$\tau \dot{E}_L(t) = -E_L(t) + mS(L(t) - aE_R(t) + \epsilon E_L(t) - gH_L(t)).$$

Rewriting the derivative:

$$\dot{E}_L(t) = \frac{-E_L(t)}{\tau} + \frac{m}{\tau} S(L(t) - aE_R(t) + \epsilon E_L(t) - gH_L(t)).$$

We have two terms:

- The first term, $\frac{-E_L(t)}{\tau}$, is always non-positive (≤ 0) since $E_L(t) \geq 0$ and τ is positive.
- The second term, $\frac{m}{\tau} S(\dots)$, is non-negative because $S(\dots)$ is non-negative by definition and both the parameters m and τ are positive.

Case 1: $E_{L0} > 0$

For an initial state with $E_{L0} > 0$, in the worst-case scenario where the negative term dominates, $\dot{E}_L(t)$ will be negative. Hence $E_L(t)$ is decreasing shrinking its value toward 0.

Case 2: $E_{L0} = 0$

In general, when $E_L(t) = 0$, the equation becomes:

$$\dot{E}_L(t) = \frac{m}{\tau} S(L(t) - aE_R(t) + \epsilon E_L(t) - gH_L(t)).$$

Since $S(x) \geq 0$, we have:

$$\dot{E}_L(t) \geq 0.$$

Thus, when the function reaches zero $E_L(t) = 0$, it cannot decrease further and remains non-negative. Therefore, $E_L(t) \geq 0$ for all $t \geq 0$.

b) Analysis of $H_L(t)$

The equation for $H_L(t)$ is:

$$\tau_H \dot{H}_L(t) = -H_L(t) + E_L(t).$$

Rewriting the derivative:

$$\dot{H}_L(t) = \frac{-H_L(t)}{\tau_H} + \frac{E_L(t)}{\tau_H}.$$

We have two terms:

- The first term, $\frac{-H_L(t)}{\tau_H}$, is always non-positive (≤ 0).
- The second term, $\frac{E_L(t)}{\tau_H}$, is non-negative because $E_L(t) \geq 0$.

Case 1: $H_{L0} > 0$

Also in this case, for an initial state with $H_{L0} > 0$, in the worst-case scenario where the negative term dominates, $\dot{H}_L(t)$ will be negative. Hence $H_L(t)$ is decreasing shrinking its value toward 0.

Case 2: $H_{L0} = 0$

In general, when $H_L(t) = 0$, the equation becomes:

$$\dot{H}_L(t) = \frac{E_L(t)}{\tau_H}.$$

Since $E_L(t) \geq 0$, we have:

$$\dot{H}_L(t) \geq 0.$$

Thus, $H_L(t)$ cannot decrease below zero.
Also for $H_L(t)$, the function may decrease, but it cannot become negative. Therefore, $H_L(t) \geq 0$ for all $t \geq 0$.

c) Analysis of $E_R(t)$ and $H_R(t)$

The same reasoning applied to $E_L(t)$ and $H_L(t)$ holds for $E_R(t)$ and $H_R(t)$ due to the symmetry of the system of equations, ensuring that $E_R(t) \geq 0$ and $H_R(t) \geq 0$ for all $t \geq 0$.

Question 2 - Find the equilibrium point

Given constant inputs $L(t) = L^* \in [0, 1]$ and $R(t) = R^* \in [0, 1]$, determine the equilibria $(E_L^*, H_L^*, E_R^*, H_R^*)$ of the system for which $E_i^* > 0$ for each $i \in \{L, R\}$. To simplify computations, we may let $c = 1 - m\epsilon + mg$.

Given constant inputs $L(t) = L^* \in [0, 1]$ and $R(t) = R^* \in [0, 1]$, we are looking for the equilibria of the system for which $E_L^* > 0$ and $E_R^* > 0$. At equilibrium, the derivatives are zero, i.e., $\dot{E}_L = 0$, $\dot{H}_L = 0$, $\dot{E}_R = 0$, and $\dot{H}_R = 0$. This gives us the following system of equations:

$$0 = -E_L^* + mS(L^* - aE_R^* + \epsilon E_L^* - gH_L^*) \tag{1}$$
$$0 = -H_L^* + E_L^* \tag{2}$$
$$0 = -E_R^* + mS(R^* - aE_L^* + \epsilon E_R^* - gH_R^*) \tag{3}$$
$$0 = -H_R^* + E_R^* \tag{4}$$

At equilibrium, we have $H_L^* = E_L^*$ and $H_R^* = E_R^*$ from the second and fourth equations. Substitute these into the first and third equations:

$$E_L^* = mS(L^* - aE_R^* + (\epsilon - g)E_L^*) \tag{5}$$
$$E_R^* = mS(R^* - aE_L^* + (\epsilon - g)E_R^*) \tag{6}$$

Given that $E_L^* > 0$ and $E_R^* > 0$ the arguments of $S(\cdot)$ must be positive, so the activation function $S(x) = x$. This simplifies the system to:

$$E_L^* = m\,(L^* - aE_R^* + (\epsilon - g)E_L^*) \tag{7}$$
$$E_R^* = m\,(R^* - aE_L^* + (\epsilon - g)E_R^*) \tag{8}$$

Solving for E_L^* and E_R^*

From the first equation, we isolate E_L^*:

$$E_L^* = \frac{m(L^* - aE_R^*)}{c}$$

where $c = 1 - m(\epsilon - g)$.

Now, substituting the expression for E_L^* into the equation for E_R^*:

$$E_R^* = m\left(R^* - a\frac{m(L^* - aE_R^*)}{c} + (\epsilon - g)E_R^*\right)$$

Rewriting it we obtain:

$$E_R^* = \frac{m\left(R^* - a\frac{mL^*}{c}\right)}{c - \frac{a^2m^2}{c}}$$

This solves the expression for E_R^*.

Once E_R^* is found, we can substitute it back into the expression for E_L^* into:

$$E_L^* = \frac{m(L^* - aE_R^*)}{c}$$

The equilibrium values for E_L^* and E_R^* are obtained in terms of the parameters and input values L^* and R^*.

Question 3 - Instability and Bifurcation

Using the fact that $\tau \ll \tau_H$, study the stability of those equilibria by considering H_L and H_R as constants. More precisely, give a condition on the parameters m, a, and ϵ ensuring instability of these equilibria. Considering a as a bifurcation parameter, show that a bifurcation occurs and give the value of a at which it occurs.

---

a) Studying the stability

Assuming that $\tau \ll \tau_H$ and treating H_L and H_R as constants, the system simplifies to two linear equations after removing the activation function $S(x)$:

$$f(E_L, E_R) = -E_L + mL^* - amE_R + \epsilon mE_L - mgH_L$$
$$g(E_L, E_R) = -E_R + mR^* - amE_L + \epsilon mE_R - mgH_R$$

The Jacobian matrix J is :

$$J = \begin{pmatrix} \frac{\partial f}{\partial E_L} & \frac{\partial f}{\partial E_R} \\ \frac{\partial g}{\partial E_L} & \frac{\partial g}{\partial E_R} \end{pmatrix}$$

Thus, the Jacobian matrix is:

$$J = \begin{pmatrix} -1 + \epsilon m & -am \\ -am & -1 + \epsilon m \end{pmatrix}$$

The eigenvalues λ are found by solving the characteristic equation:

$$\det(J - \lambda I) = 0$$

This leads to the quadratic equation:

$$(-1 + \epsilon m - \lambda)^2 - (am)^2 = 0$$

That can be simplified as:

$$-1 + \epsilon m - \lambda = \pm am$$

Solving this yields:

$$\lambda_{1,2} = -1 + \epsilon m \pm am$$

For instability, one of the eigenvalues must be positive. For example:

$$-1 + \epsilon m + am > 0$$

Thus, the condition for instability is:

$$a > \frac{1 - \epsilon m}{m}$$

b) Bifurcation

A bifurcation occurs when the system transitions from stability to instability, i.e., when one eigenvalue from negative becomes positive. This happens when:

$$a = \frac{1 - \epsilon m}{m}$$

Question 4 - Time evolution of $E_L(t)$ and $E_R(t)$

Implement the model on Matlab-Simulink with $L(t) = 1$, $R(t) = 0.95$, $a = 3.4$, and $\varepsilon = 0$. Then, create a file `main.m` providing all parameter values, defining the initial state $\mathbf{x_0} = [0\ 0\ 0\ 0]$, and defining T_s and the simulation duration `StopTime` $= 20$ s.
Plot the time evolution of $E_L(t)$ and $E_R(t)$. Comment the results.

---

a) Result

Figure 1 shows the implementation of the Simulink model with all its input parameters allowing for adjustment of values for reusability in different scenarios.

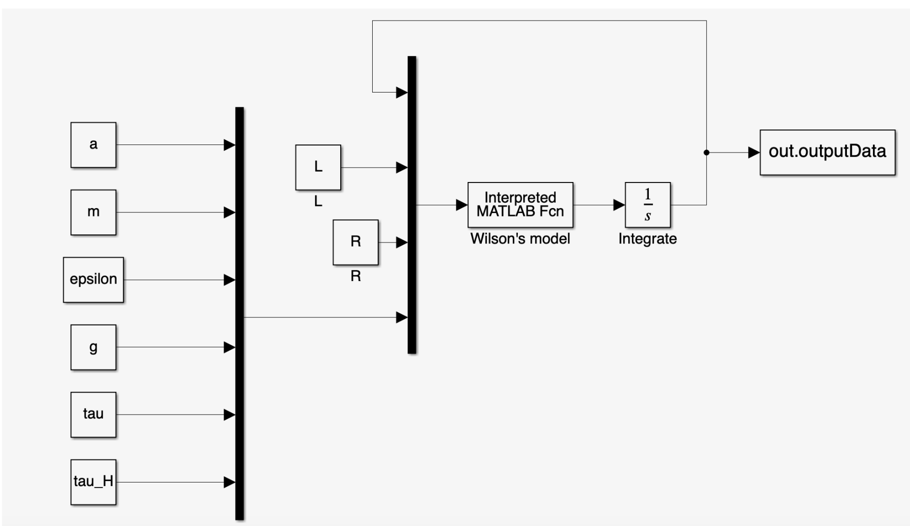

Figure 1: Simulink model

The function defined in the `Interpreted MATLAB Fcn` block is defined in the MATLAB code section below.

In Figure 2 you can find the plot of the time evolution of $E_L(t)$ and $E_R(t)$ over a simulation time of 20 seconds. In the plot, the neuronal activities driven by the intensity of the stimulus on the two eyes alternate in dominance over time. In particular, we can observe that when the intensity of the stimulus is higher, as is the case for the left eye with $L(t) = 1$ and $R(t) = 0.95$, the neural activity generated lasts longer compared to the one generated by the eye with lower stimuli, which is the right eye in this case. $E_L(t)$ and $E_R(t)$ exhibit this alternating pattern, where one rises as the other falls, creating a clear cycle of perceptual switching between the left and right eye's dominance.

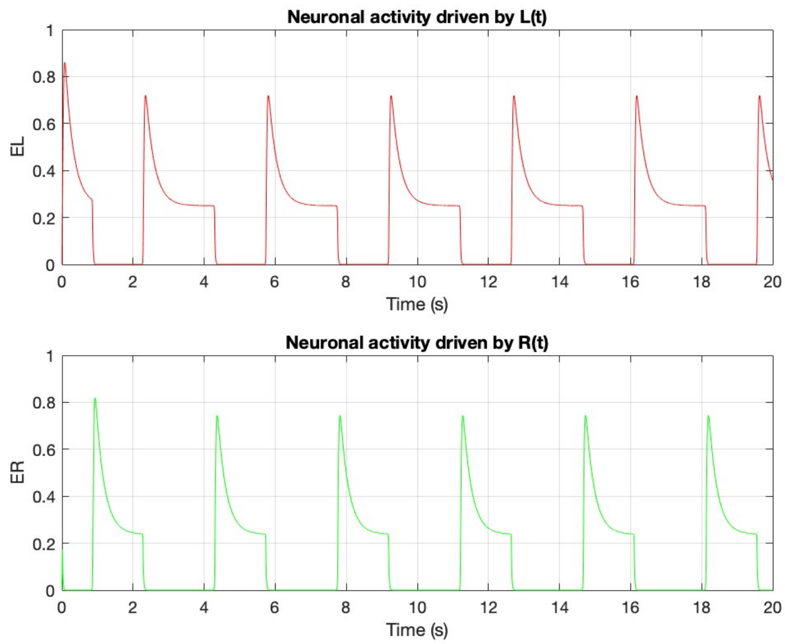

Figure 2: Time evolution of $E_L(t)$ and $E_R(t)$

b) MATLAB code

The following MATLAB code contains the definition of the function `binocularRivalryModel(x)` used int the `Interpreted MATLAB Fcn` block of the Simulink model.

```
% File: binocularRivalryModel.m
% Author: Mario Roca - mario.roca@etu-upsaclay.fr

function dx_dt = binocularRivalryModel(x)
    % Extract the state variables from x
    EL = x(1);  % Neuronal activity driven by L(t)
    HL = x(2);  % Hyperpolarizing current of the left eye
    ER = x(3);  % Neuronal activity driven by R(t)
    HR = x(4);  % Hyperpolarizing current of the right eye

    L = x(5);
    R = x(6);
```

```
    a = x(7);
    m = x(8);
    epsilon = x(9);
    g = x(10);
    tau = x(11);
    tau_H = x(12);

    % Define the activation function S(x) = max(0, x)
    S = @(y) max(0, y);

    % The four equations of the binocular rivalry model
    dEL_dt = (-EL + m * S(L - a * ER + epsilon * EL - g * HL)) / tau;
    dHL_dt = (EL - HL) / tau_H;
    dER_dt = (-ER + m * S(R - a * EL + epsilon * ER - g * HR)) / tau;
    dHR_dt = (ER - HR) / tau_H;

    % Return derivatives as a vector
    dx_dt = [dEL_dt, dHL_dt, dER_dt, dHR_dt];
end
```

The following MATLAB code contains the first part of the main with the definition of the parameters, initial state and answer to question 4 running the simulation of the model for 20 seconds and plotting the time evolution of $E_L(t)$ and $E_R(t)$.

```
% File: main.m
% Author: Mario Roca - mario.roca@etu-upsaclay.fr

% Parameter values
tau = 15e-3;        % Time constant
tau_H = 1;          % Time constant
m = 1;              % Excitability
a = 3.4;            % Inhibitory synaptic weight
g = 3;              % Excitatory synaptic weight
epsilon = 0;        % Excitatory synaptic weight
L = 1;        % Constant input to the left eye
R = 0.95;     % Constant input to the right eye

% Initial state [E_L(0), H_L(0), E_R(0), H_R(0)]
x0 = [0, 0, 0, 0];

% Sampling time and simulation duration
Ts = 1e-3;          % Sampling time (1ms time steps)
StopTime = 20;      % Simulation time in seconds

% QUESTION 4

out = sim('binocularRivalry.slx', 'StopTime', '20', 'FixedStep', '1e-3');

time = out.outputData.time;            % Time vector
stateData = out.outputData.data;       % State data matrix

% Plotting the state variables
```

```
figure;

subplot(2, 1, 1);
plot(time, stateData(:, 1), 'r');
xlabel('Time␣(s)');
ylabel('EL');
title('Neuronal␣activity␣driven␣by␣L(t)');
grid on;

subplot(2, 1, 2);
plot(time, stateData(:, 3), 'g');
xlabel('Time␣(s)');
ylabel('ER');
title('Neuronal␣activity␣driven␣by␣R(t)');
grid on;
```

Question 5

For the same values of the stimuli intensity $L(t)$ and $R(t)$, draw the bifurcation diagram of the system, considering the synaptic weight a between the two eyes as the bifurcation parameter. To that end, for values of $a \in [0, 3.5]$, measure the maximum and minimum steady-state values of $E_L(t)$ and $E_R(t)$. Comment on the obtained curve. Justify the fact that the approximation made in Question 3) was indeed adequate.

---

a) Result

In Figure 3 you can find the bifurcation diagram of the system. As the plot shows, for $a < 1$, the system exhibits stability characterized by low and constant neural activity, with a consistent dominance of the left eye, since $L = 1$ and $R = 0.95$. However, when a crosses the critical value of 1, the system undergoes a mathematical instability, which leads to perceptual instability manifested as oscillatory alternation between the neural activities of the left and right eyes. This perceptual instability does not imply that the entire system is unstable; rather, it results in regular shifts in dominance while maintaining stable periodic dynamics. Thus, while the bifurcation diagram reveals critical transitions, it is essential to differentiate between mathematical instability and the robust oscillatory behavior that characterizes the system's perceptual alternations.

The approximation made in Question 3 was appropriate, as the mathematical result aligns with the behavior observed in the simulations. The instability condition derived in Question 3 was $a > \frac{1-\epsilon m}{m}$, which holds in this case since $\epsilon = 0$ and $m = 1$.

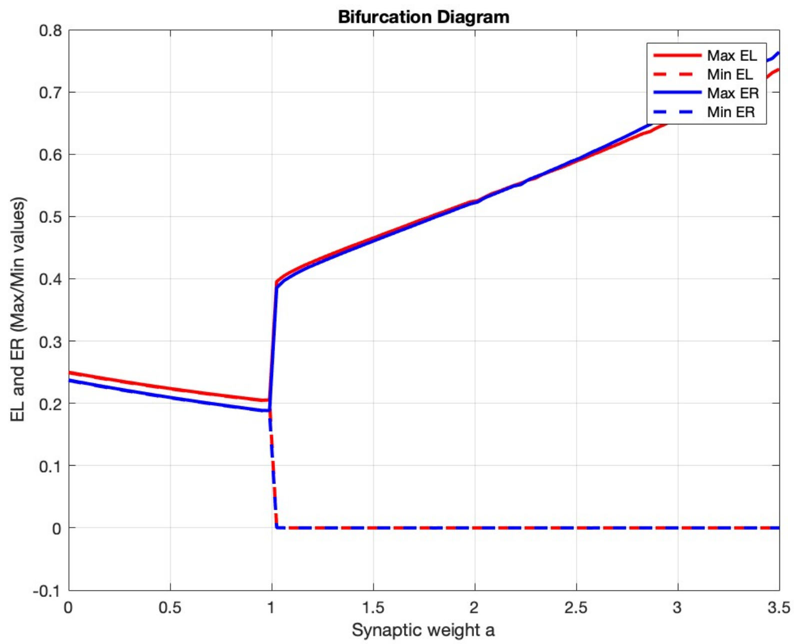

Figure 3: Bifurcation Diagram

b) MATLAB code

The following MATLAB code contains the solution to Question 5 defining 100 values in the range [0, 3.5], running the simulations and plotting the bifurcation diagram with the max and min values of $E_L(t)$ and $E_R(t)$.

```

% QUESTION 5

% Define the bifurcation parameter range
a_values = linspace(0, 3.5, 100);

% Preallocate arrays to store max and min values of EL and ER
max_EL = zeros(1, length(a_values));
min_EL = zeros(1, length(a_values));
max_ER = zeros(1, length(a_values));
min_ER = zeros(1, length(a_values));

% Loop over different values of 'a'
for i = 1:length(a_values)
    % Set the current value of 'a'
    a = a_values(i);

    % Run the simulation for the current value of 'a'
    out = sim('binocularRivalry.slx', 'StopTime', '20', 'FixedStep'
        , '1e-3');
```

```

    % Extract the data
    stateData = out.outputData.data;
    time = out.outputData.time;
    EL_data = stateData(:, 1);
    ER_data = stateData(:, 3);

    % Discard the transient part (first 5 seconds)
    steady_state_indices = find(time > 5);
    EL_steady = EL_data(steady_state_indices);
    ER_steady = ER_data(steady_state_indices);

    % Find the max and min values in the steady state
    max_EL(i) = max(EL_steady);
    min_EL(i) = min(EL_steady);
    max_ER(i) = max(ER_steady);
    min_ER(i) = min(ER_steady);
end

% Plot the bifurcation diagram
figure;
plot(a_values, max_EL, 'r-', 'LineWidth', 2); hold on;
plot(a_values, min_EL, 'r--', 'LineWidth', 2);
plot(a_values, max_ER, 'b-', 'LineWidth', 2);
plot(a_values, min_ER, 'b--', 'LineWidth', 2);
xlabel('Synaptic weight a');
ylabel('EL and ER (Max/Min values)');
title('Bifurcation Diagram');
legend('Max EL', 'Min EL', 'Max ER', 'Min ER');
grid on;
```

Question 6

Levelt's second law predicts that "increasing the intensity of the stimulus at one eye tends to decrease the perceptual dominance of the other eye." Check whether the proposed model successfully captures this trend. To that aim, for $L = 1$, $a = 3.4$, $\epsilon = 0$, $m = 1$, and R taking constant values in $[0.86, 0.99]$, draw the steady-state duration of perception of the left and right stimuli as a function of the right stimulus intensity R.

---

a) Result

In Figure 4, the steady-state duration of perception for the left and right stimuli is shown as a function of the increasing right stimulus intensity R, taking constant values in $[0.86, 0.99]$. As illustrated in the graph, Levelt's second law holds: increasing the intensity of the stimulus in one eye tends to decrease the perceptual dominance of the other eye.

To obtain these curves, I defined the function `count_spikes(..)` as shown in Section b), where I iterate through the signal to count the number of spikes and compute the duration for which the signal remains on. The plotted dominance time was calculated as the average on-duration divided by the number of spikes for each respective signal.

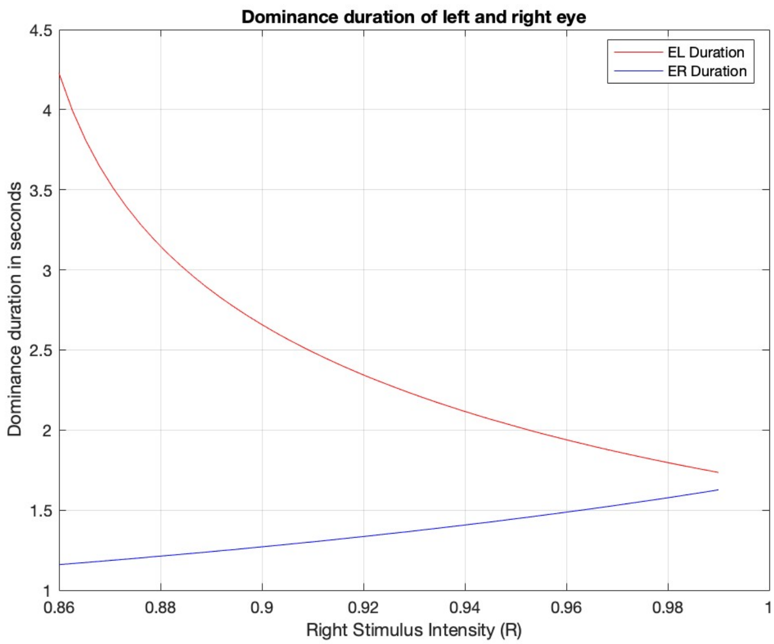

Figure 4: Dominance duration

b) MATLAB code

```
% File: count_spikes.m
% Author: Mario Roca - mario.roca@etu-upsaclay.fr

function [number_spikes, on_duration] = count_spikes(E, threshold)
% Iterate through the signal, count the number of "spikes", and compute how
% long the signal stays "on"

    % Initialize variables
    number_spikes = 0;
    on_duration = 0;
    i = 1;
    len = length(E);

    % Skip initial values below threshold
    while i <= len && E(i) >= threshold
        i = i + 1;
    end

    % Main loop through the signal
```

```
    while i <= len
        % Wait for the rising phase
        while i <= len && E(i) < threshold
            i = i + 1;
        end

        % Count time above threshold (on duration)
        start_on = i;
        while i <= len && E(i) >= threshold
            i = i + 1;
        end
        if i <= len && E(i) < threshold  % Only count if there was a full rise and fall
            on_duration = on_duration + (i - start_on);
            number_spikes = number_spikes + 1;
        end
    end
end
```

```

% QUESTION 6

a = 3.4;
% Right stimulus intensity range [0.86, 0.99]
R_values = linspace(0.86, 0.99, 50);  % 50 steps between 0.86 and 0.99

EL_dominance = zeros(1, length(R_values));
ER_dominance = zeros(1, length(R_values));

% Loop over different values of R
for i = 1:length(R_values)
    R = R_values(i);

    % Simulate the system for the given R
    out = sim('binocularRivalry.slx', 'StopTime', '20', 'FixedStep', '1e-3');

    % Extract the data
    stateData = out.outputData.data;
    time = out.outputData.time;
    EL_data = stateData(:, 1);
    ER_data = stateData(:, 3);

    % Discard the transient part (first 5 seconds)
    steady_state_indices = find(time > 5);
    EL_steady = EL_data(steady_state_indices);
    ER_steady = ER_data(steady_state_indices);

    threshold = 0.1;

    [n_spikes_EL, EL_ON_duration] = count_spikes(EL_steady, threshold);
    [n_spikes_ER, ER_ON_duration] = count_spikes(ER_steady, threshold);

    EL_dominance(i) = (EL_ON_duration / n_spikes_EL) * Ts;
```

```
    ER_dominance(i) = (ER_ON_duration / n_spikes_ER) * Ts;
end

% Plot the results
figure;
plot(R_values, EL_dominance, '-r', 'DisplayName', 'EL Duration');
hold on;
plot(R_values, ER_dominance, '-b', 'DisplayName', 'ER Duration');
xlabel('Right Eye Stimulus Intensity (R)');
ylabel('Dominance duration in seconds');
legend('Location', 'best');
title('Dominance duration of left and right eye');
grid on;
```

Question 7

Levelt's third law states that "increasing the stimulus strength at one eye increases the perceptual alternation rate." The alternation rate is the inverse of the alternation period. This alternation period can be considered to be the sum of the perceptual durations of each eye. Based on the data obtained in Question 6), compute this alternation period and check whether the model captures this law.

---

a) Result

The Wilson model captures Levelt's third law: increasing the stimulus strength at one eye, in this case R, increases the perceptual alternation rate. Figure 5 shows the alternation rate as a function of the right eye stimulus intensity R. The plot was obtained as the inverse of the alternation period, where the alternation period is the sum of the left and right on-durations, as shown in the code section below.

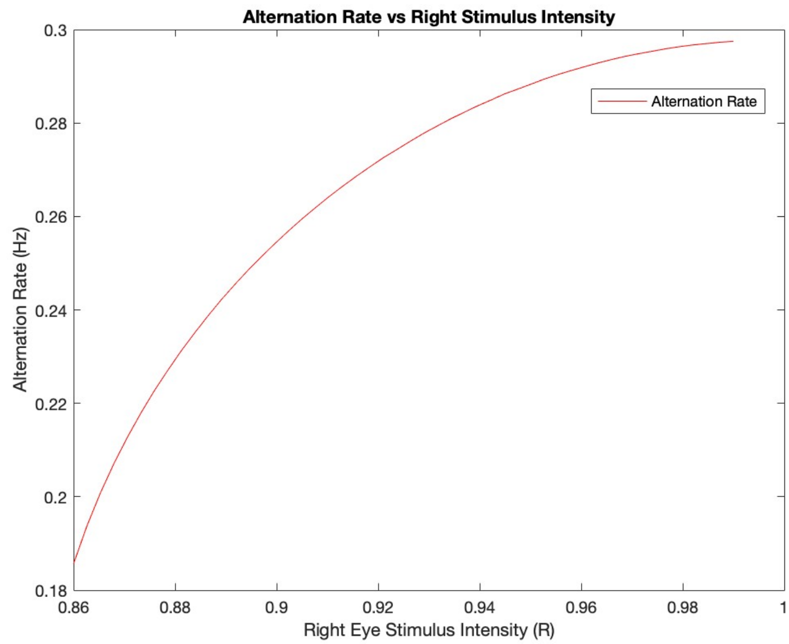

Figure 5: Alternation Rate

b) MATLAB code

```
% QUESTION 7

% Compute the alternation period (sum of left and right eye ON
    durations)
alternation_period = EL_dominance + ER_dominance;

% Compute the alternation rate (inverse of alternation period)
alternation_rate = 1 ./ alternation_period;

% Plot alternation rate vs right eye intensity
figure;
plot(R_values, alternation_rate, '-r', 'DisplayName', 'Alternation␣
    Rate');
xlabel('Right␣Eye␣Stimulus␣Intensity␣(R)');
ylabel('Alternation␣Rate␣(Hz)');
title('Alternation␣Rate␣vs␣Right␣Stimulus␣Intensity');
legend show;
```