CASE STUDY REPORT - GROUP 5

Team members: Vyshnavi Arunkumar(03803119), Deepa Ramakrishnan Subathra(03799043), Om Mihani(03803669)

INDEX

Task 1

In this task, a comparison between the performance of Euler's method with MATLAB's built-in ODE solver ode45 to solve a gene expression model is done. The model explains how protein R_1 is synthesized and degraded in response to an inhibitor (R_2) and a constant input (u(t)). We also performed a parameter study to analyze the quality of Euler's method with sinusoidal input, varying the frequency parameter (p) in the range [0.1, 20]. The ODE in question is described by Equation 1.

$$\dot{R}_1 = k_{s1} \frac{u(t)}{1 + (R_2/K_2)^n} - k_1 R_1 \tag{1}$$

Task 1.1: Euler's Method vs ode45 Solver

In Task 1.1, a model1.m function file was created that calculates the RHS of equation 1 which is also compatible with MATLAB's ODE solvers. It accepts t,y and p as input parameters and the state variable y represents R_1 in the equation. It returns the difference between the synthesis and degradation(dy/dt), which is the expected output.

Then we implemented Euler's method to solve the ODE for protein R_1 using the given model1 and compared the results with MATLAB's ode45 solver. We used a constant input (u(t) = 1) and a constant inhibitor (R_2 = 1), with an initial condition for (R_1(0) = 3). The time interval was set from 0 to 10, and the number of intervals for Euler's method was set to N = 40.

The code for solving the ODE using both methods is as follows:

1. Euler's Method: The function euler_ode_solv was implemented, which uses Euler's method to solve the ODE by updating the state variable at each time step.

2. MATLAB ode45 Solver: The built-in ode45 solver was used to solve the same ODE. Both methods were compared in the task1_main.m file and then the results were plotted for the concentration of R1 over time.

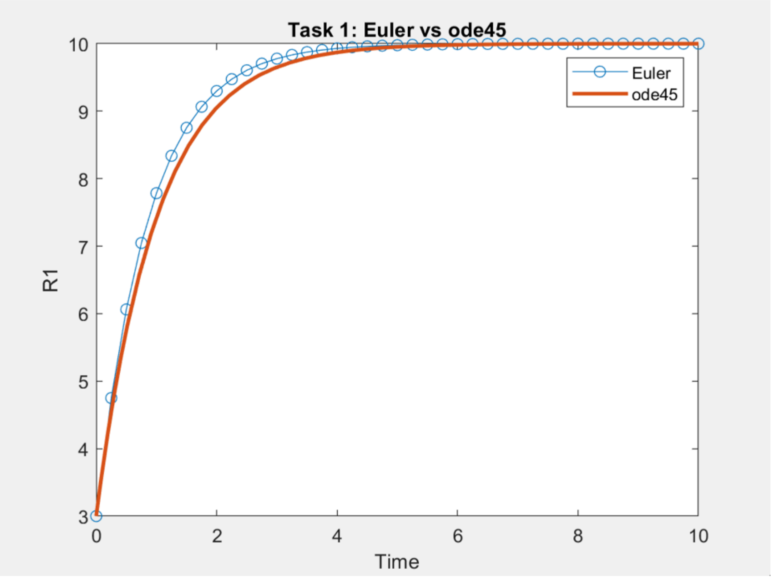

Fig 1: Euler Vs ode45 comparison graph

The comparison plot clearly shows that ode45 provides a much smoother and more accurate solution compared to Euler's method. Euler's method, with a relatively coarse step size (N = 40), produces an approximation that diverges more as time progresses.

Task 1.1 Observations

When we compare Euler's method with ode45, we found that Euler's method yields a less accurate answer, especially for longer time intervals or when the system is changing quickly. Being an adaptive solution, ode45 manages the system;s dynamics considerably better. Because of the greater step size, Euler's approach produced data with apparent oscillations and inaccuracies, whereas ode45 produced smooth curve that more closely matched the intended answer.

Q) Can you improve your result by modifying N?

Yes, increasing the number of intervals (N) will improve the accuracy of Euler's method. A larger N will reduce the step size, improving the approximation, but Euler's method will still not be as accurate as ode45, which uses adaptive step sizes.

Task 1.2: Sinusoidal Input Parameter Study

In Task 1.2, we analyzed the quality of Euler's method by performing a parameter study with a sinusoidal input, where the input function is implemented using input_u function where p.freq adjusts the frequency is given by Equation 2:

$$u(t) = \sin(p\,t) + 1 \qquad (2)$$

where (p) is a frequency parameter that varies in the range [0.1, 20]. This study aimed to observe how the solution behavior changes with increasing values of (p).

In the code, we implemented a loop to solve the ODE using Euler's method for different values of (p) and plotted the resulting concentrations of R_1 over time through the task1_sinusoidal.m code file. The input function is implemented using the input_u.m function file where p.freq adjusts the frequency which is varied across four values: 0.1, 1, 5, and 20. The results were plotted for each frequency.

Task 1.2 Observations

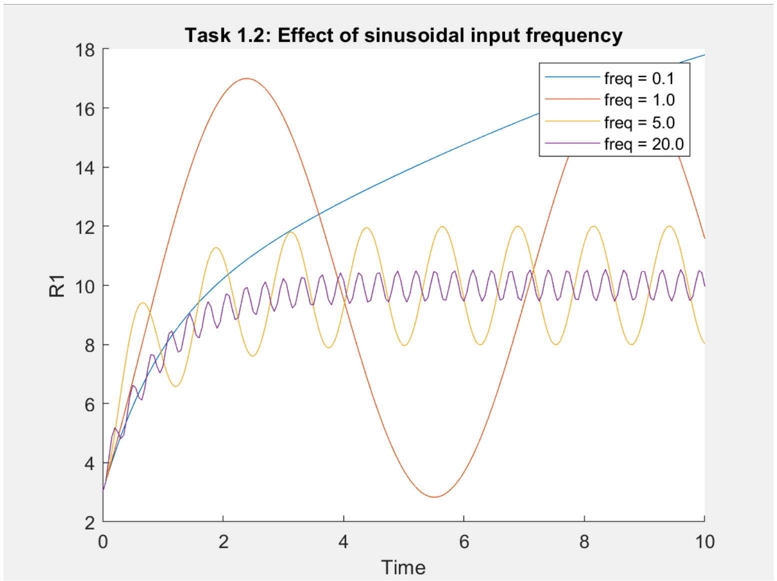

Fig 2: Effect of sinusoidal input frequency on the concentration of R_1 graph

As the frequency parameter (p) increased, the sinusoidal input oscillated more rapidly. Euler's method struggled to keep up with the high-frequency oscillations, resulting in noticeable inaccuracies and oscillations in the solution for higher values of (p) due to the larger step size and inability to capture rapid changes in input. This demonstrates that Euler's method is not well-suited for high-frequency inputs and behaves like a numerical low-pass filter.

Task 2

In Task 2, a two–gene regulatory network is modelled using coupled ordinary differential equations. The system consists of two proteins, R_1 and R_2. Protein R_1 represses the synthesis of R_2, while protein R_2 represses the synthesis of R_1. Both proteins are produced at a regulated rate and degraded proportionally to their concentration.

The Task 2 system consists of two ordinary differential equations describing the dynamics of proteins R_1 and R_2:

$$\begin{aligned}\dot{\boldsymbol{R}}_1 &= \boldsymbol{k}_{s1}\frac{\boldsymbol{u}_1(\boldsymbol{t})}{1+(\boldsymbol{R}_2/\boldsymbol{K}_2)^n} - \boldsymbol{k}_1\boldsymbol{R}_1\\ \dot{\boldsymbol{R}}_2 &= \boldsymbol{k}_{s2}\boldsymbol{u}_2(\boldsymbol{t}) - \boldsymbol{k}_2\boldsymbol{R}_2\end{aligned}$$

Protein R_1 is repressed by R_2 through a Hill-type inhibition function, while protein R_2 is produced independently of R_1 .Both proteins are degraded according to first-order kinetics.

The production of each protein is modelled using a Hill repression function. The parameters k_{s1} and k_{s2} represent the maximum synthesis rates, while K_1 and K_2 denote the half-maximal repression constants. The Hill coefficients n and m determine the steepness of repression. Protein degradation is modeled as a first-order process with degradation rates k_1 and k_2.

Task 2.1 Simulation of the system

The system was simulated numerically using MATLAB's ode45 solver and compared with the Euler's method. The input signals u_1 and u_2 were chosen to be constant. All parameters were kept fixed except for the degradation rate k_2 , which was varied to analyze its influence on system behavior. Initial conditions for both protein concentrations were chosen as non-negative values.

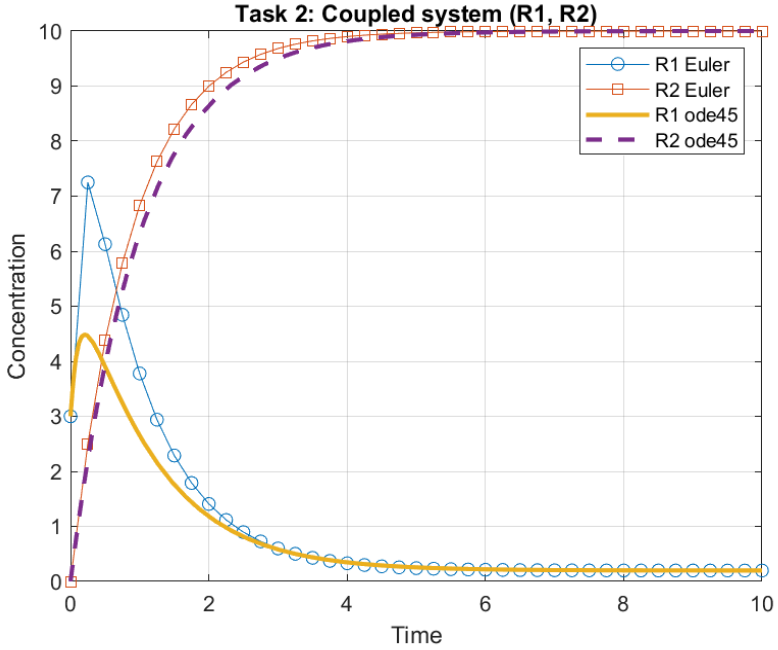

Figure 3: Simulation of the new coupled system with Euler's method and **ode45**

During the simulation(in Figure 3) , the protein R_1 initially increases, reaching a transient peak at early time points. After the peak, R_1 declines and settles down to a low steady state value.This behavior is due to the strong repression of R_1 by R_2. As R_2 accumulates, it inhibits the production of R_1, leading R_1 to decay due to degradation.The Euler's method slightly overestimates the early peak compared to the ODE45 solver, which indicates the numerical approximation error at early time intervals.

The dynamic behavior of R_2 increases monotonically from zero and approaches a high steady state at concentration(~10).This depicts the R_2 production dominates over the degradation once repression from R_1 becomes weak. Here, both Euler and ODE45 solver solutions closely match, indicating the R_2 dynamics are smoother and are very less sensitive to the numerical method.

Keypoint: Both the systems settles into a stable steady state characterized by:

1. Low R_1 concentration
2. High R_2 concentration

This outcome is called **mutual repression**, where R_2 dominates the network under the chosen parameter set.

Task 2.2 Influence of the degradation parameter k_2

The degradation parameter k_2 controls how fast protein R_2 is removed from the system and therefore plays a key role in regulating the repression of R_1.

This figure shows how changing the parameter k_2 affects the time evolution and steady-state level of **protein R_1** in the coupled gene regulatory network.

Biologically:

- **Low k_2** : strong repression (even small R_2 strongly inhibits R_1)
- **High k_2** :weak repression (more R_2 is needed to inhibit R_1)

Case 1: k_2 influence on R_1

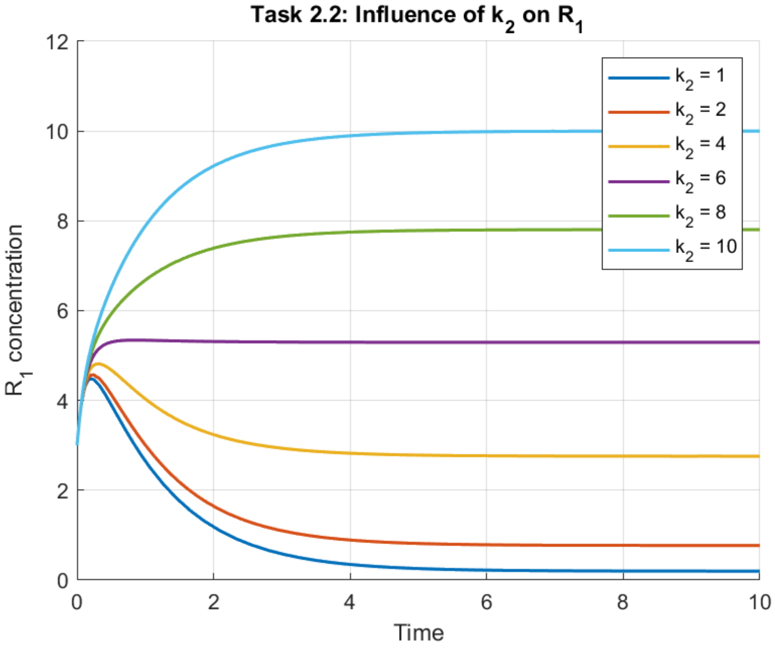

Figure 4: Influence of k2 parameter on R1 protein at time interval between 1 and 10

Figure 4 shows that for all values of k_2, R_1 exhibits a transient increase at early time points, which is followed by convergence to a steady state. As k_2 increases, the steady-state concentration of R_1 rises significantly. At low k_2 values, strong repression by R_2 suppresses R_1, resulting in a low steady-state level. In contrast, higher k_2 values weaken the repression, allowing R_1 to accumulate and approach its maximal production level. These results demonstrate that k_2 plays a critical role in regulating the steady-state expression of R_1 by modulating the strength of repression exerted by R_2.

Case 2: k_2 influence on R_2

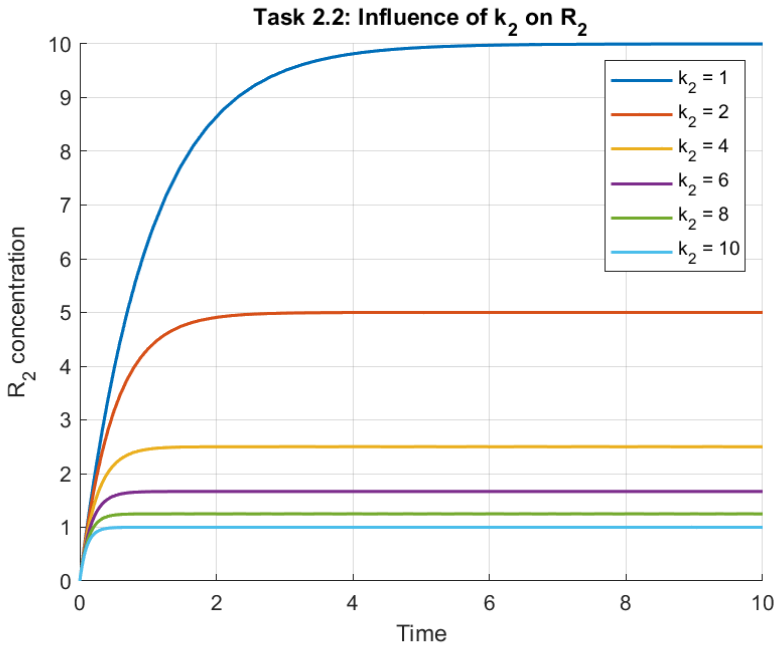

Figure 5: Influence of k_2 parameter on R_2 protein at time interval between 1 and 10

In figure 4, as we can see that for values of R_2 , in all cases, R_2 starts from an initial concentration of zero and increases monotonically until it reaches a steady-state value.

As the degradation parameter k_2 increases, the steady-state concentration of R_2 decreases significantly. For k_2 =1, R_2 reaches the highest steady-state level, while progressively larger values of k_2 results in lower steady-state concentrations.

Therefore, the degradation rate of one protein can significantly influence the behavior of the entire regulatory system.

Task 3

Equation form:

$$\dot{R}_1 = k_{s1} \frac{u_1(t)}{1 + (R_2/K_2)^n} - k_1 R_1$$

$$\dot{R}_2 = k_{s2}\frac{u_2}{1+(R_1/K_1)^{n*}} - k_2 R_2$$

The coupled system describes a mutual inhibition (“toggle switch”) between R_1 and R_2, where each protein represses the production of the other. When the parameters are chosen appropriately, the system becomes bistable: depending on the initial values of R_1 and R_2 or on the effective production rates, it settles into one of two states, with one protein at a high level and the other at a low level. At steady state (saturation), the production and degradation terms balance for each protein, so the net rate of change of both R_1 and R_2 is zero.

Task 3.1

With the initial values of R_1 and R_2 being 3 and 0, the plot looks like this:

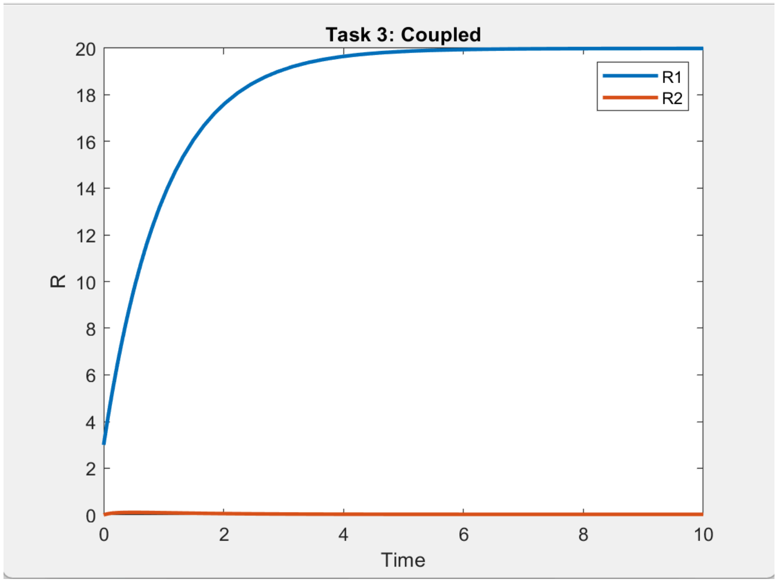

Figure 6: Base case for coupled toggle switch at stable state R_1 high

Both equations use the same Hill coefficient, so the strength and steepness of the mutual inhibition are symmetric. At the same time, the production rate constant k_{s1} and the initial concentration of R_1 are larger than those of R_2, which biases the system toward R_1 dominance. Consequently, once the system relaxes to a steady state, R_1 settles at a high level while R_2 remains low.

Now, lets see this case in the R_2 vs R_1 space:

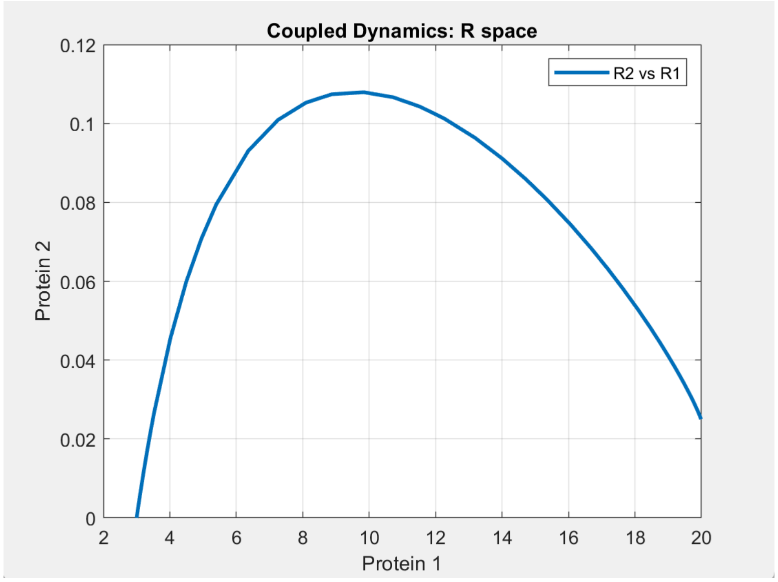

Figure 7: Base case for coupled toggle switch at stable state R_1 high in R space

Observe that the second protein increases very slightly and then decreases. This very slight increase can be attributed to the fact that there is no consumption at t = 0. Now, let's compare with other initial values:

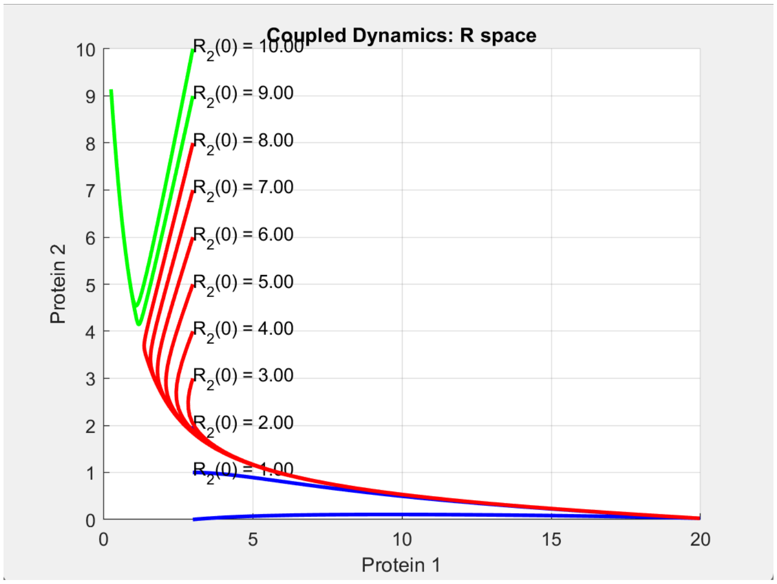

Figure 8: Multiple domain cases for coupled toggle switch with varying R_2 starting concentration in R space

The phase-plane plot reveals three main dynamical domains and a fourth hypothetical one.

A. Blue domain: R_1 increases monotonically while R_2 decreases, so trajectories move toward a steady state with high R_1 and low R_2
B. Red Domain: R_1 initially decreases because inhibition by R_2 dominates, but later its production overtakes degradation and inhibition; the system again converges to a state with high R_1 and decreasing R_2.
C. Green Domain: R_1 keeps decreasing due to strong inhibition by R_2; R_2 first decreases slightly and then rises, driving the system toward a state with low R_1 and high R_2
D. Hypothetical fourth domain (very small R_1 (0)): If the system starts with almost no R_1, R_2 increases while R_1 may rise a bit and then decay back toward zero, corresponding to a basin where the "R_2-high / R_1-low" state is strongly favored.

Task 3.2

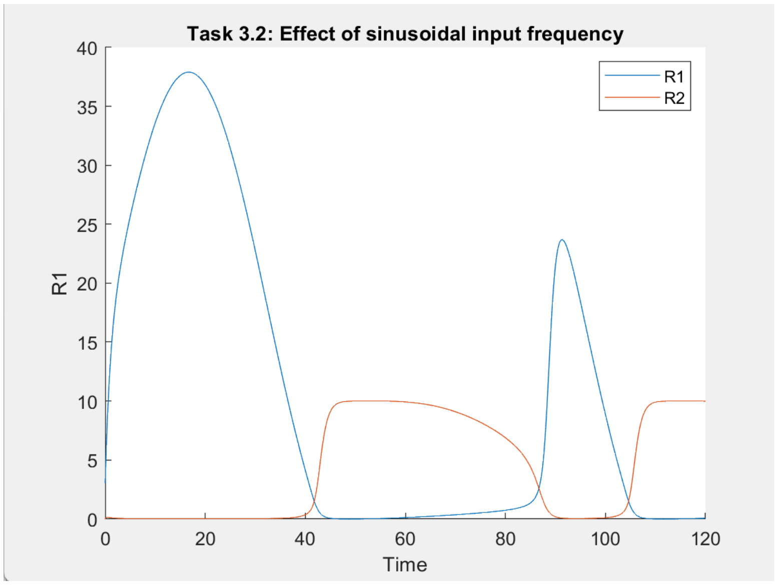

Figure 9: Coupled toggle switch with sinusoidal generation function for R_1 displaying swinging state

In task 3.2, we apply a sinusoidal input of the form $u(t) = \sin(p_1 t) + p_2$ to the coupled toggle switch system and look for parameter values that cause repeated switching between the two stable states. The behaviour is observed only for a low input frequency of 0.1 and an offset in the range of 0.9 to 1. This can be explained by the fact that if the offset is too large or too small, then one of the proteins will completely dominate over the other, irrespective of the sinusoidal function. The frequency is chosen keeping the target of three switches in mind.

For these values, the input changes slowly enough to generate a quasi-steady state. The frequency allows a value of sin(pi/2) to be reached in about 15 time units, allowing the maximum generation of R_1. After this, the production slowly drops and keeps decreasing till about 45 minutes. This allows the production of R_2 to increase and dominate. When the sin curve turns around again, R_1 dominates over R_2. Observe that interestingly, the first peak of R_1 is higher and wider than the second peak. This can be explained by the fact that initially, we start with zero R_1 and zero R_2. However during the second peak, there is some R_2 already present, which shapes the second peak to be lower and narrower.

Thus, the sin curve with constant offset allows a self switching toggle system.