# Final Master Thesis

## Self-regulation of cell population in microbial consortia

University Master in Industrial Engineering

Marta González Larequi



Supervisors:

Mario di Bernardo

Davide Fiore

Naples, July 2020

# Abstract

In recent years, multicellular applications have been increased in the field of synthetic biology. However, this has led to the appearance of a new problem; ensure the stable co-existence of the populations in microbial consortia. As different cell types have different growth conditions, they tend to compete with each other for resources, leading to the disappearance of one of the populations.

This issue has been studied in recent years by multiple research groups. However, in all these previous works the final model proposed either needed some external inputs or implied a mutual dependence between the two populations. In thesis we present a novel approach to the problem of stable coexistence of different cell populations in microbial consortia, proposing a design in which the cells are able of self-regulating their relative number in the consortium. This is achieved by embedding a bistable memory mechanism, a genetic toggle switch, inside each cell, so that the current internal state defines the population the cell belongs to. On one hand, taking as reference previous works, the mathematical model of the system, as well as a simplified version of it, will be developed. On the other hand, the numerical validation of the model will be performed including an analysis of the parameters role in the system, a study of the capability of tuning the final ratio and a robustness analysis to cell heterogeneity.

# Index

# Figure index

# 1. Introduction

Biological systems are dynamical systems that have the capacity to reproduce, replicate, grow, adapt and evolve. Cells are the basic building blocks of these living systems. The complexity of this systems is explained for different reasons. First, cells are composed of a very large number and variety of elements interacting in space and time and its dynamic functioning is of a nonlinear nature. On the other hand, cell-biological systems are difficult to observe and are subject to continuous change [1].

Since this complexity of the cellular systems was discovered, researchers had tried to control and manipulate them to productive ends leading to the emergence of synthetic biology, a subfield of bioengineering in which researchers use genetic engineering to develop novel biomolecular pathways [2].

Over the past years, synthetic biology technologies and applications have experimented a considerable progress making this field become a widely recognized branch of biological research. The recent innovative technologies and engineering approaches have enabled a huge advance towards practical applications in biotechnology and medicine [3].

Some of the current applications of synthetic biology are the production of biofuels and other valuable chemicals, molecular computation and logic, medical diagnostics and artificial microbial communities. Nevertheless, there exists an extent list of possible applications that could be develop in the future such as the correction of nutritional deficiencies with engineered gut bacteria, the production of medicines from rare or endangered plants in distributed fermenters or the production of biofuels sustainably on demand in programmable fermentation tanks [2].

Until now, a substantial part of the synthetic biology researcher's efforts was focused on the implementation of synthetic gene circuits in single cells. Nevertheless, single cells biological circuits present several throwbacks, such as non-compatible chemical reactions, an excessive metabolic burden or retroactivity. This limits the complexity of the genetic circuits that can be embedded into the cells, limiting hence the new functionality that can be designed. In order to overcome these issues and design more advanced functionalities researchers started working on multicellular synthetic gene circuits [4].

Engineering a synthetic microbial consortium allows the design of more complex genetic circuits providing more sophisticated functionalities. However, this approach requires the cohabitation of different species of bacteria with different growth rates that will compete for the same resources, so the stable co-existence of the different cell populations must be ensured [5], [6].

In pursuance of maintaining the stable co-existence of the cell populations in a microbial consortium, researchers are working to find strategies to control their relative population numbers (i.e. their ratio). This problem has been recently investigated by different researchers' groups. In [4] Xinying Ren *et al.* designed a synthetic control circuit that regulates the total cell population and relative ratio between two different cell strains developing a dual feedback control strategy. The

Imperial College team, competitor in the iGEM competition, worked also on this topic and engineered a genetic circuit, composed by three modules (communication, comparator and growth regulation) that allows the ratiometric control of populations in a co-culture [6]. In [7], the authors constructed a synthetic ecosystem based on predator-prey systems where two E.coli populations communicate bi-directionally through quorum sensing and regulate each other's gene expression. Another approach to this issue was proposed in [8], where the design of a synthetic feedback controller implemented across two different populations was investigated. In [9] the authors constructed a synthetic microbial consortium composed of two cell strains that when cultured together show the circuit topology of a synthetic dual-feedback oscillator operating within a single strain. Other important results were obtained using bistable genetic toggle switches embedded in cells that give the cells a reversible role that can be switched from one population to another, allowing them to balance the cell populations in their microbial consortiums [10], [5], [11].

The aim of this final master project is to study the problem of maintaining a stable coexistence of different cell populations in microbial consortia and develop new multicellular strategies for the regulation of the relative population numbers using control engineering tools, but with a novel approach; built cells capable to self-regulate their ratios. The approach that we propose here is to embed a bistable memory mechanism (genetic toggle-switch) inside each cell, whose current internal state defines which of the two possible populations the cell belongs to.

Other systems designed to control the population ratio in microbial consortia, are based on a dynamic equilibrium encoded in the synthetic design. In those approaches, if one of the two population suddenly dies the other one can either die because there is a symbiotic dependence between the strains [4] or grow uncontrollably at full rate [6]. Unlike these solutions, with our approach if one of the populations dies out the cells on the cell population of the strain will remain stable. Another difference with previous works, [5], [11], is that, in our system, is not necessary to externally control the ratio with external inputs, because the consortia is capable of self -regulate its own cell population.

This new scenario involves several challenges. First, we must ensure the communication between all the cells in the consortia, as each cell must detect the size of its own population as well as the size of the other one. At the same time, each cell must be capable of comparing this population numbers in order to realize if there is an unbalance in the consortia and act consequently encouraging the change of role by means of the toggle switch, if necessary.

This thesis has been developed as a part of the COSY-BIO project (2017-2020) during an Erasmus Exchange in the Università degli Studi di Napoli Federico II. COSY-BIO is a multidisciplinary project fund by the European Union's Horizon 2020 research and innovation program. The goal of this project is to develop a theoretical framework and innovative technological tools to engineer reliable biological systems that are robust despite their individual components being not by translating principles of control engineering to molecular and cell biology [12].

# 2. Literature review

## 2.1. Biology background

The cell is the smallest unit of life, which constitutes the elemental unit of all living beings. Cells are made of several thousand types of proteins which are produced in response to every different signal the cell is capable to sense. The cell responds to a signal by making the right amount of each protein, which is achieved by mapping signals and genes, Figure 1. For this aim, the cell does an internal representation using special proteins called transcription factors, which are design to transit rapidly between active and inactive molecular states. Each active transcription factor can bind the DNA and regulate the rate at which genes are read, acting as Activators if they increase the rate or as Repressors if they decrease it. [13].

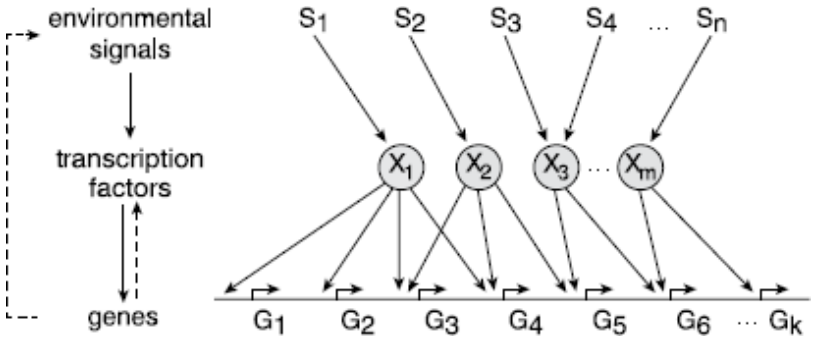


*Figure 1:Internal representation of the signals and genes mapping inside a cell [13]*

The transcription network describes the interaction between transcription factors and genes and controls the expression levels of mRNA and their corresponding proteins, Figure 2. This gene regulation process is composed by two main actions: Transcription and Translation. During the Transcription the gene is copied into the mRNA (messenger RNA) by the RNAp (RNA polymerase) and in the Translation, the mRNA is then translated into the protein mainly by the ribosomes [13].

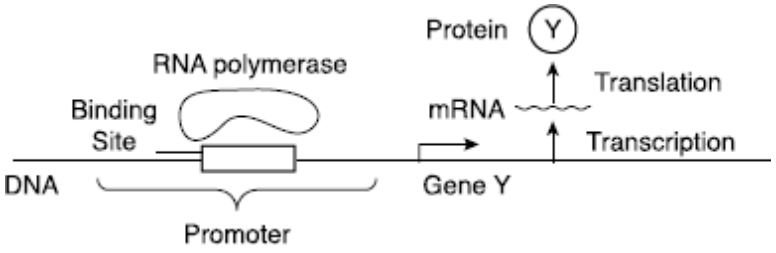


*Figure 2:Transcription network representation [13]*

## 2.2. Gene input functions

Typically, the mathematical function used to describe gene input functions is the Hill function. This function can be modeled in two different ways depending on the role of the transcription factor (Activator or repressor) [13]. Considering the production rate of a protein Y controlled by a transcription factor X these functions can be written as follows:

Hill function for an activator:

$$f(x^*) = \alpha \cdot \frac{(x^*)^n}{K^n + (x^*)^n} \tag{1}$$

Hill function for a repressor:

$$f(x^*) = \alpha \cdot \frac{K^n}{K^n + (x^*)^n} \tag{2}$$

Where $\alpha$, K and n in (1) and (2) are the maximal promoter activity, the activation coefficient, and the Hill coefficient respectively. For an activator the Hill input function is an increasing curve while for a repressor is a decreasing curve, Figure 3.

In these equations, X* represents the concentration of the active transcription factor X in the cell. For the activation case, that is when X is an activator for the expression of gene Y, (Equation 1), with high activator concentrations, that is X*>>K, the fraction term tends to 1 obtaining the maximal promoter activity, $\alpha$, as result. This makes sense because X* binds the promoter with higher probability. However, with low values of X*, that is X*≈0, the equation will tend to 0 (as the fraction term tends to 0).

In the opposite way, (Equation 2), with high activator concentrations the equation will tend to 0 (as the fraction term tends to 0) and with low values of X* it is obtained the maximal expression rate (as the fraction term tends to 1).

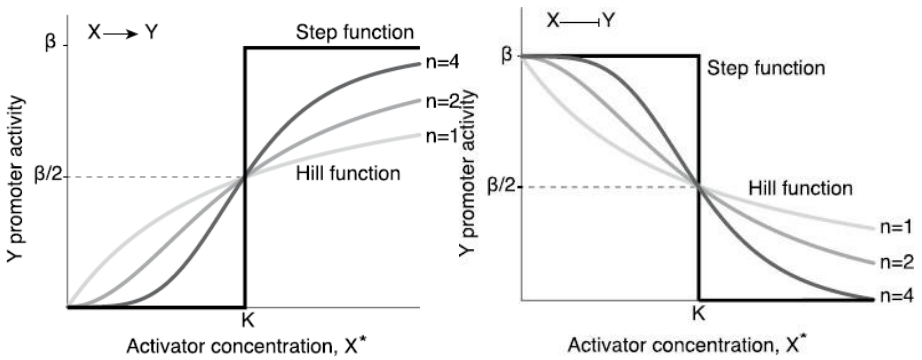


*Figure 3: From left to right representation of the Hill functions for an activator and for a repressor respectively depending on the Hill coefficient [13]*

On the other hand, it will be also taken into account the fact that the expression rate of protein Y is also balanced by protein degradation (destruction by specialized proteins in the cell) and dilution (the reduction in concentration due to cell volume growth and division). Many proteins, called stable proteins, are not actively degraded in growing cells. The production of this proteins is balanced by dilution due to the increasing volume of the growing cell and for them the response time, $T_{1/2}$ (the time to reach halfway between the initial and final levels in a dynamic process), is equal to one cell generation time, τ. This way, when a cell produces a protein if suddenly the production stops, the cell grows and when it doubles its volume splits into two cells [13]. Thus, after one cell generation time, the protein concentration decreases by half:

$$T_{1/2} = \frac{\log(2)}{\gamma} = \tau \tag{3}$$

Furthermore, many genes have nonzero minimal expression level which is called the gene's basal expression level [13]. For example, the transcription rate of a gene Y with basal expression level degradation and dilutions rates could be written as follows:

For an activator:

$$f(x^*) = \alpha_0 + \alpha \cdot \frac{(x^*)^n}{K^n + (x^*)^n} - x^* \cdot (d + \gamma) \tag{4}$$

For a repressor:

$$f(x^*) = \alpha_0 + \alpha \cdot \frac{K^n}{K^n + (x^*)^n} - x^* \cdot (d + \gamma) \tag{5}$$

Being $\alpha_0$, $d$, $\gamma$ in (3) y (4) the basal expression rate, the degradation rate and the dilution rate, respectively.

## 2.3. Genetic toggle switch

The fundamental element in the self-regulatory system we propose in this thesis is a genetic toggle switch. A genetic toggle switch is a synthetic, bistable memory gene-regulatory network. By applying some inputs (the inducers) we can induce the toggle switch to flip between two different stable states exhibiting a nearly ideal switching threshold [14].

This first genetic toggle switch was proposed in [14] and is composed by two repressors and two promoters, Figure 4. Repressor 1 that is promoted by Promoter 2 is in charge of repressing Promoter 1, which promotes Repressor 2 that repress Promoter 2, closing the loop.

Without taking into account the Inducers, the toggle switch present only two stable states; either Promoter 1 promotes Repressor 2, or Promoter 2 promotes Repressor 1. When the Inducers are introduced the switching becomes possible. The Inducer represses the repressing action of the corresponding repressor enabling the other Repressor to be maximally transcribed until the systems stabilizes in the opposite state.

This behavior is described by the following equations:

$$\dot{u} = \frac{\alpha_1}{1+v^{\beta}} - u \tag{6}$$

$$\dot{v} = \frac{\alpha_2}{1+u^{\gamma}} - v \tag{7}$$

Where $u$, $v$, $\alpha_1$, $\alpha_2$, $\beta$ and $\gamma$ in (3) and (4) are the concentration of repressor 1, the concentration of repressor 2, the effective rate of synthesis of repressor 1, the effective rate of synthesis of repressor 2, the cooperativity of repression of promoter 2 and the cooperativity of repression of promoter 1, respectively. The first term of the equations represents the cooperative repression of constitutively transcribed promoters and the second term the degradation/dilution of the repressors [14].

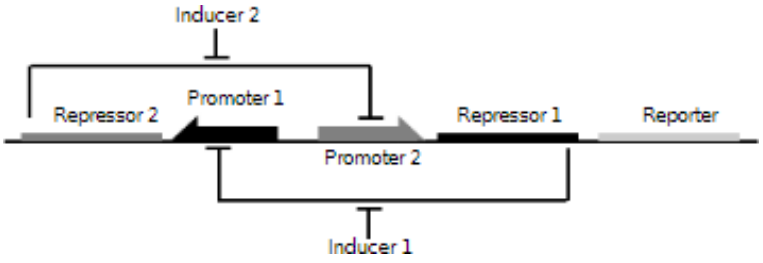


*Figure 4: Toggle switch design [14]*

To behave as described above, the toggle-switch dynamics needs to satisfy some conditions. First of all, the structure of the toggle network should create two basins of attraction. Secondly, the nullclines must intersect three times (have 3 equilibrium points), two stable equilibrium points, and one saddle point, whose stable manifold is the separatrix between the regions of attraction of the other two equilibria. If the initial condition of the system is above the separatrix it will settle to state 1 and in the opposite way it will settle to state 2 if the initial condition is below this separatrix. Finally, the rates of synthesis of both repressors should be balanced in order to ensure the 3 intersection points [14]. If the toggle switch model designed satisfies all the above conditions the phase portrait of the toggle network should be similar to Figure 5.

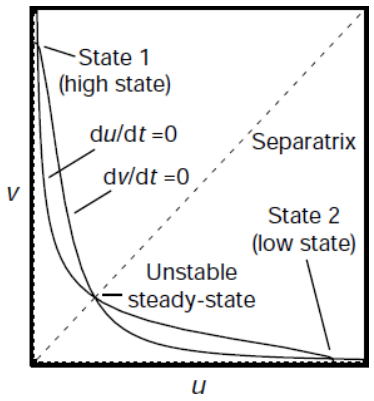


*Figure 5: Bistable toggle switch network satisfying all the conditions [14]*

In this thesis, the toggle switch is embedded in the cells to provide them the capacity of changing to what population they belong to in order to regulate the ratio of the two populations. A similar idea was presented in [5], where it was studied the design of three different external feedback control strategies able to steer the inducer molecules inputs to control the populations' ratio. The authors presented a different approach in means of how to induce the synthetic cell to switch between the two stable states. This work and others will be discussed later in the Stable co-existence in microbial consortia section.

However, the use of toggle-switches embedded in cells, is not limited to the application of ratio regulation. In the following subsections, two control strategies, with alternative applications, involving the use of a toggle-switch will be reviewed.

### 2.3.1. Balancing cell populations endowed with a synthetic toggle switch via adaptative pulsatile feedback control

One recent research where a synthetic toggle switch was embedded in the cells with the aim of controlling cell populations was presented in [15]. In this research they propose the use of adaptative pulsatile feedback control strategies to improve the robustness and balancing performance of the system by selecting in real-time the amplitude and the duty-cycle of the inducer molecules (aTc and IPTG), which affect the behavior of the toggle switch. This way, adapting these input signals in real-time but maintaining their periodic nature, they take advantage of the beneficial effects of periodic forcing inputs while enhancing the stability, coherence and robustness of the system [15].

In Figure 6a the dynamics of the toggle switch are represented where the two genes LacI and TetR are bound with RFP and GFP, respectively, and repress each other's expression while the external inducer molecules, IPTG and aTc, modulate the strength of those repressions. Following these dynamics and taking as reference other studies they define the model of the synthetic toggle switch as follows

$$\frac{dmRNA_{LacI}}{dt} = k_L^{m0} + \frac{k_L^m}{1+\left(\frac{TetR}{\theta_{TetR}}\cdot\frac{1}{1+\left(\frac{aTc}{\theta_{aTc}}\right)^{n_{aTc}}}\right)^{n_{TetR}}} - g_L^m \cdot mRNA_{LacI} \tag{8}$$

$$\frac{dmRNA_{TetR}}{dt} = k_T^{m0} + \frac{k_T^m}{1+\left(\frac{LacI}{\theta_{LacI}}\cdot\frac{1}{1+\left(\frac{IPTG}{\theta_{IPTG}}\right)^{n_{IPTG}}}\right)^{n_{LacI}}} - g_T^m \cdot mRNA_{TetR} \tag{9}$$

$$\frac{dLacI}{dt} = k_L^p \cdot mRNA_{LacI} - g_L^p \cdot LacI \tag{10}$$

$$\frac{dTetR}{dt} = k_T^p \cdot mRNA_{TetR} - g_T^p \cdot TetR \tag{11}$$

$$\frac{daTc}{dt} = \begin{cases} k_{aTc}^{in}(u_{aTc} - aTc), & if\ u_{aTc} > aTc \\ k_{aTc}^{out}(u_{aTc} - aTc), & if\ u_{aTc} \leq aTc \end{cases} \tag{12}$$

$$\frac{dIPTG}{dt} = \begin{cases} k_{IPTG}^{in}(u_{IPTG} - IPTG), & if\ u_{IPTG} > IPTG \\ k_{IPTG}^{out}(u_{IPTG} - IPTG), & if\ u_{IPTG} \leq IPTG \end{cases} \tag{13}$$

From a control point of view, in Figure 6b, they illustrate how the control inputs, $u_{aTc}$ and $u_{IPTG}$, are first filtered by the cell membrane and then enter the toggle switch through nonlinear Hill-like functions. Then, in Figure 6c, they sketch one of the pulsatile inputs applied to control the toggle switch being the other input its mirror image as they are mutually exclusive [15].

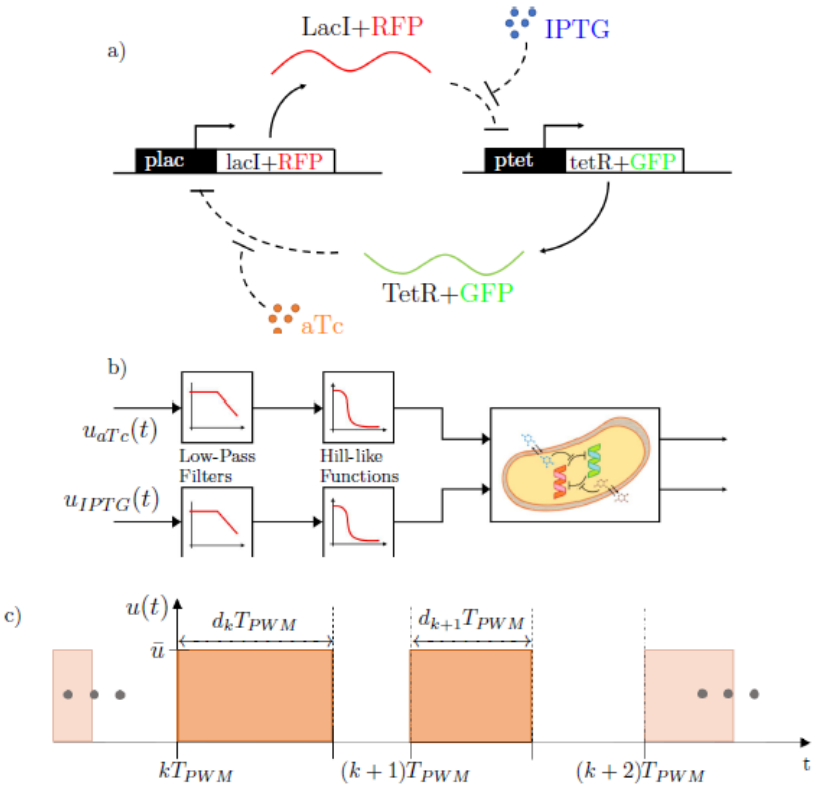


*Figure 6: a) Schematic of the Genetic Toggle Switch network structure. The two genes (LacI and TetR) –respectively bound with RFP and GFP-mutually repress each other; the external inducers, IPTG and aTc, modulate the strengths of the repression exerted by LacI and TetR on each other. b) The toggle switch as a multi-input, multi-output (MIMO) control system. The control inputs are first filtered because of diffusion through the cell membrane and then enter the toggle switch nonlinearly through Hill-like functions. c) Sketch of one of the pulsatile inputs applied to control the toggle switch. Being mutually exclusive, the other input is chosen as its mirror image. TPWM is the period of the inputs, dk represents the fraction of the k-th period during which the input is switched ON (the other being OFF) [15].*

In order to prove the need of a close-loop control strategy, they first analyze the effect of applying the inputs in an open-loop fashion by using precomputed amplitudes, $\bar{u}_{aTc}$ and $\bar{u}_{IPTG}$, and duty-cycle, $d_k$. As expected, this test showed that the strategy fails to achieve the balancing goal in the presence of diffusion, cell-to-cell variability and noise, as the evolution of the average values over the population of LacI and TetR expressed by the cells does not converge to the desired values ( Figure 7) [15].

In view of the results, they design a feedback control approach based on using two mutually exclusive periodic inputs and also able to adapt and change their duty-cycle and amplitude to achieve the desired behavior. The control strategy, illustrated in Figure 8, is based on two control actions; A feedforward action which pre-computes the ideal value of the amplitude and duty-cycle required to achieve the control goal without perturbations and diffusion effects and a feedback action that is in charge of adapting the duty-cycle of the periodic inputs as a function of the current cell behavior [15].

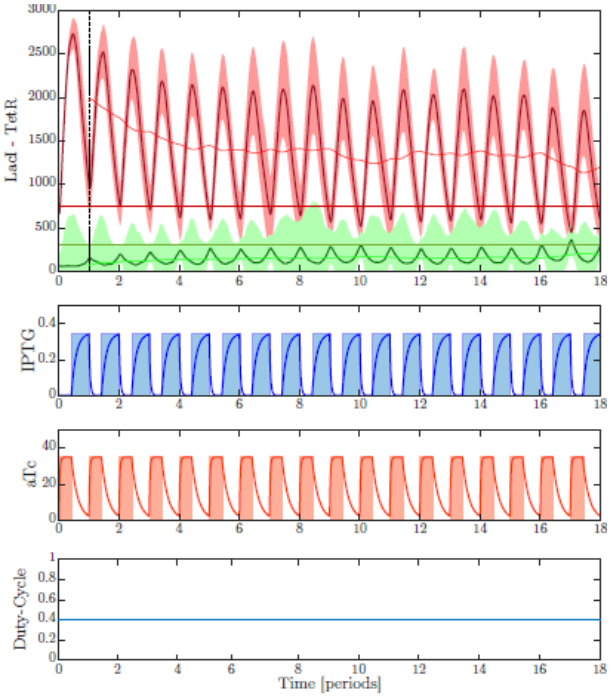


*Figure 7: Open-loop periodic forcing. Evolution of TetR and LacI when cells are subject to mutually exclusive pulsing inputs whose amplitude and duty-cycle were precomputed off-line. Top: Dashed red and green lines: Desired setpoint. Solid dark red and green lines: Evolution of the average values over the population of LacI and TetR as expressed by the cells. Shaded areas: Value of the standard deviation from each mean value, computed at each time instant. Solid light red and green lines: Evolution of the mean value of the oscillations evaluated with a moving window of period equal to $T_{PWM}$. The lack of convergence to the desired set-point shows the limits of open-loop, pre-computed inputs in achieving the control objective when diffusion and other effects are appropriately modeled. Middle:Evolution of the pulsing inputs applied to the system. Bottom: Duty-cycle that is kept constant over time [15].*

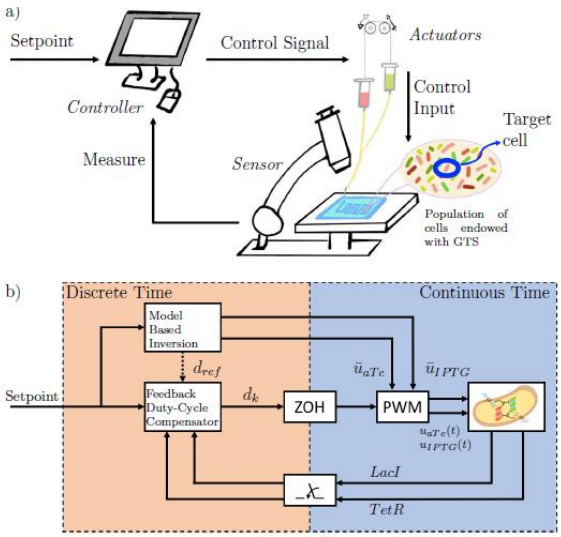


*Figure 8: a) External Control Architecture. A population of E. coli endowed with the Genetic Toggle Switch is hosted in a microfluidic device. A fluorescence microscope takes pictures of the cells, whose average RFP and GFP values are evaluated through segmentation algorithms. This information and the setpoint, is sent to the controller that computes online the inputs to be applied to the cells. The actuators, receive the control signal and produce the action needed to feed the population of cells with the required inputs. b) Block diagram of the proposed closed-loop hybrid control strategy. The population of cells, together with the PWM inputs, evolve in continuous-time. The controller is designed in discrete-time, computing the control input at each time period $T_{PWM}$. A feedforward Model Based Inversion block evaluates the amplitudes $\bar{u}_{aTc}$ and $\bar{u}_{IPTG}$, of the pulse wave inputs, on the basis of the setpoint $[LacI_{Ref}\ TetR_{Ref}]$. The feedback controller evaluates and adapts in real time the duty-cycle of the inputs as a function of the desired setpoints and the outputs of the system. The (ZOH) keeps the duty-cycle, computed by the compensator at the beginning of each period, constant during the rest of the period $T_{PWM}$ [15].*

Then, they compare the effectiveness of two feedback control strategies: A Proportional-Integral controller that drives a pulse width modulation block (PI-PWM) and a Model Predective Controller (MPC) that optimizes a desired cost function to select the input duty-cycle dynamically. To this end, they carry out deterministic (Figure 9) and stochastic (Figure 10) in-silico experiments whose results show that both strategies accomplish the final goal of controlling the toggle switch to the desired output value despite the presence of perturbations and uncertainties. In the deterministic case (Figure 9) the MPC shows better performance in terms of dynamic regulation (lower overshoot and transient duration) and also better steady-state regulation of the set-point. In the stochastic case (Figure 10) the results lead to the same conclusions [15].

Additionally, they carried out robustness test by introducing parametric variations in the cell population. The results depicted from these tests confirm that PI-PWM is more robust to small perturbations while larger perturbations worsen its performance making MPC preferable for in-vivo implementation [15].

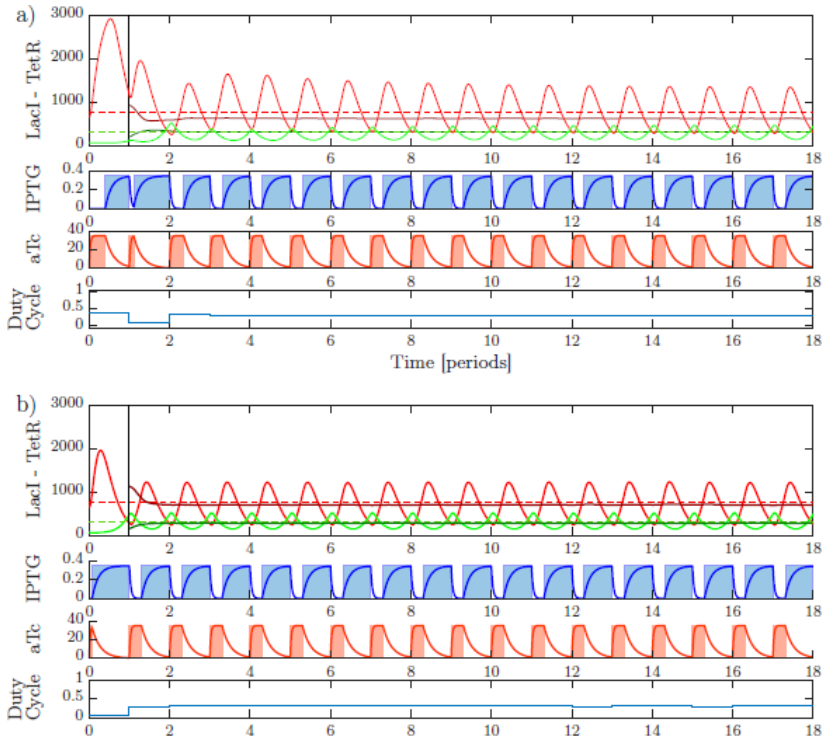


*Figure 9: In-silico deterministic experiment: comparison of the performance of the PI-PWM (a) and MPC (b) control strategies via deterministic simulations. For the PI-PWM, the duty-cycle starts from $d_{ref}$ = 0.4 and is then adapted by the controller after the first period, while the MPC computes the duty-cycle from the start (solving the optimization problem). Top panels: Dashed red and green lines: Setpoint of the experiment, respectively $LacI_{Ref}$ and $TetR_{Ref}$. Solid lines: Evolution of promoter proteins for LacI (red) and TetR (green). Dark solid lines, starting from t = $T_{PWM}$: Mean values of the state in the time period, evaluated with a moving window of period $T_{PWM}$. Middle panels: Evolution of the pulsing inputs applied to the system. Bottom panels: Evolution of the duty-cycle over time [15].*

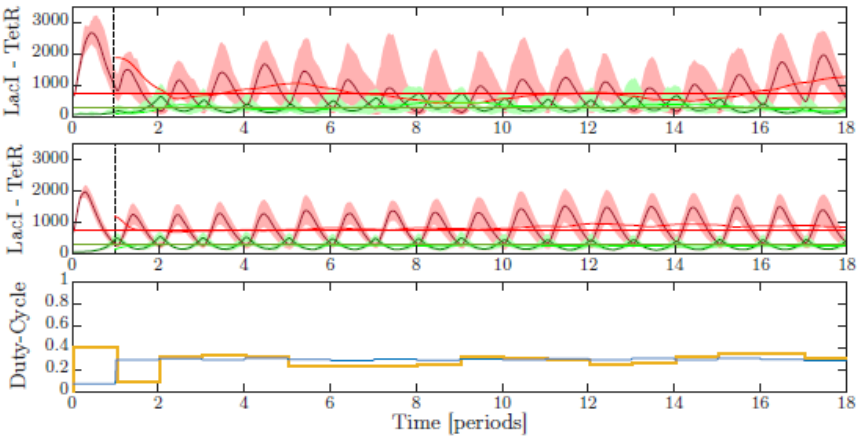


*Figure 10: In-silico stochastic experiment: comparison of the performance of the PI-PWM (top panel) and MPC (central panel) control strategies. Dashed lines: Setpoint of the experiment, for $LacI_{Ref}$ (red) and $TetR_{Ref}$ (green). Solid red and green lines: Average evolution of LacI and TetR over the population. Darker solid lines: Evolution of the mean trajectory in the period, evaluated with a moving window as in the deterministic case. Shaded areas: Values of the standard deviation from the means, at each time instant. Bottom panel: Duty-cycle evolution over time when the MPC (blue) or the PI-PWM (yellow) strategies are used [15].*

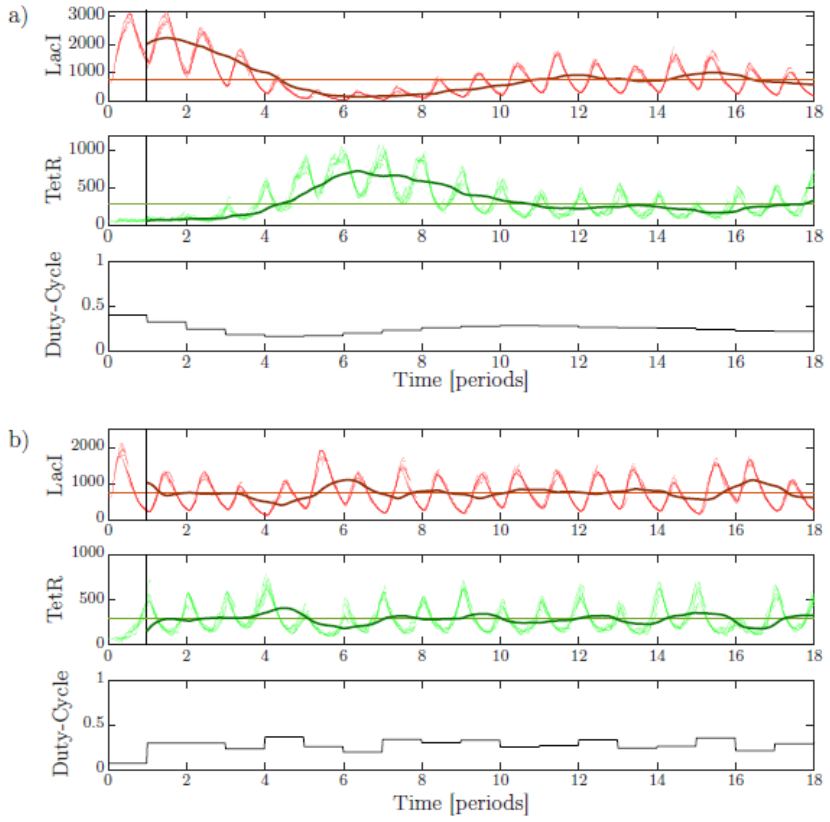


*Figure 11: Agent-based simulation in BSim 2.0 of the PI-PWM (a) and MPC (b) control strategies. E.colli cells are considered growing in a single chamber of a "mother machine"-like microfluidic device: the simulations start with a single cell located at the bottom of the chamber; as the cell grows and duplicates, it pushes outside of the chamber new cells that exceed the maximum capacity of the chamber. The top panel shows the evolution over time of LacI; the dashed line representing the setpoint $LacI_{Ref}$, while lighter lines the evolution of the state for each cell in the simulation, and the darker solid line the mean trajectory computed over the population, evaluated through a moving window of period $T_{PWM}$. The middle panel shows the evolution over time of TetR; the dashed line representing the setpoint $TetR_{Ref}$, lighter lines are the evolution of the state for each cell in the simulation, while the dark solid line represents the evolution of the mean trajectory across the population in the period, evaluated using a moving window of period $T_{PWM}$. The bottom panel shows the evolution of the duty-cycle over time [15].*

Finally, they perform agent-based simulations using BSim (Figure 11) confirming the viability and effectiveness of the strategies for in-vivo experiments. After that, they introduce a 10% variation of all the parameters of each cell in the agent-based model to reproduce cell-to-cell variability, whose results ratify again that the MPC control strategy shows a better performance in the presence of such uncertainties [15].

All these results evidence the viability of their approach of using pulsatile inputs computed online by means of a feedback control strategy to achieve a robust stabilization of the toggle switch showing also a better performance of the Model Predictive Control in all the in-silico experiments [15].

### 2.3.2. Multicellular feedback control of a genetic toggle-switch in microbial consortia

An interesting application of the toggle-switch in microbial consortia was proposed in [11]. This time, the authors engineered a synthetic microbial consortium consisting of three different cell populations; two populations, the Controllers, able to respond to some reference input and induce the switch of the toggle switch embedded in a third population, the Targets, so as to activate or deactivate some additional functionalities in the cells. As usual, the communication between the three populations takes place by means of orthogonal quorum sensing molecules.

As represented in Figure 12, the controller cells, Activator and Deactivator, sense the concentration in the environment of the reference signal Ref and the Targets output $y$, which is high ($y$ =1) when the Targets are active. Then, the controllers generate the control signals $u_1$ and $u_2$ based on the following logic functions:

$$u_1 = Ref\ AND\ (NOT\ y) \tag{14}$$

$$u_2 = (NOT\ Ref)\ AND\ y \tag{15}$$

This way, the Activators activate the Targets when at the same time they sense a reference signal in the environment and the Targets are inactive, on the contrary, the Deactivators inhibit the activity of the Targets when those are active and there is no reference signal in the environment [11].

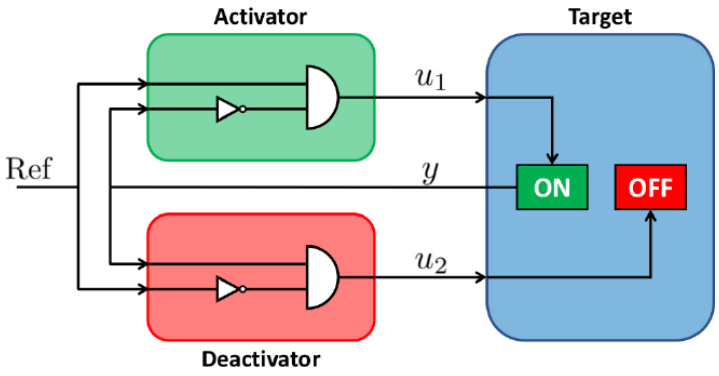


*Figure 12: Representation of the relationship between the cell populations in the consortium and their molecular signals. The Target receives ON ($u_1$ = 1) only if Ref = 1 AND y = 0 and OFF ($u_2$ = 1) only when Ref = 0 AND y = 1. This way, the input signals $u_1$ and $u_2$ are equal to 1 only when there is disagreement between Ref and y [11].*

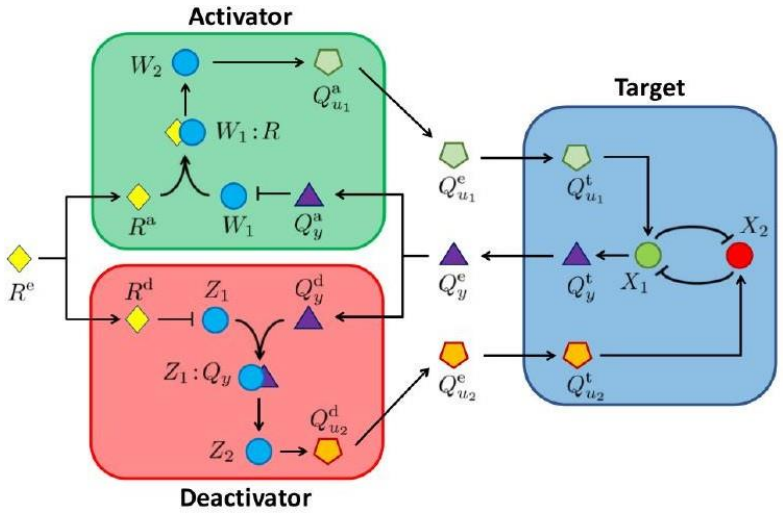


*Figure 13: Abstract biological implementation of the multicellular control system. The Controllers compare the concentrations of the signaling molecules R and $Q_y$ using an antithetic motif and produce $Q_{ui}$. This, diffuses inside the Target and promotes the activation of $X_i$, which makes the Target changes its state. Circles represent internal molecular species and polygons signaling molecules and the reference signal [11].*

They illustrated the biological implementation of the model in Figure 13, where the subscripts e, t, a, d denote quantities in the environment, Target cells, Activator cells and Deactivator cells, respectively. As said before, the Target cells have an inducible toggle switch embedded inside, which consists in two proteins $X_1$ and $X_2$, each repressing each other's expression. The full expression of $X_1$ represents the active state of the Target and the full expression of $X_2$ the inactive state. The behavior of this toggle switch is given as:

$$\dot{x}_1 = \alpha_{x1}^0 + \frac{\alpha_{x1}}{1+\left(\frac{x_2}{\theta_{x2}}\right)^{n_{x2}}} - \gamma^t \cdot x_1 + u_1 \tag{16}$$

$$\dot{x}_2 = \alpha_{x2}^0 + \frac{\alpha_{x2}}{1+\left(\frac{x_1}{\theta_{x1}}\right)^{n_{x1}}} - \gamma^t \cdot x_2 + u_2 \tag{17}$$

Where $\alpha_{xi}^0$, $\alpha_{xi}$, $\theta_{xi}$, $n_{xi}$ and $\gamma^t$ are the basal and maximal expression rates of species $X_i$, the disassociation coefficient, the hill coefficient and the degradation rate of the species $X_i$ in the Target.

Also, $x_1$ and $x_2$ are the concentrations of molecules $X_1$ and $X_2$ inside the cell and $u_1$ and $u_2$ the inputs which describe the promoting action of the signaling molecule $Q_{ui}$ on the expression of $X_i$, for i=1 ,2,by:

$$u_i := \beta_i \cdot \frac{(q_{ui}^t)^{n_{ui}}}{\theta_{ui}^{n_{ui}} + (q_{ui}^t)^{n_{ui}}} \tag{18}$$

Where $\beta_i$, $q_{ui}^t$, $n_{ui}$ and $\theta_{ui}^{n_{ui}}$ are the maximal promoter activity, the concentration of molecule $Q_{ui}$ inside the Target and the Hill and activation coefficients.

As illustrated in Figure 13, in the Deactivators, $Q_{u2}$ is generated by $Z_2$ that is promoted by $Z_1$: $Q_y$, being $Q_y$ the orthogonal quorum sensing molecule produced proportional to $X_1$. For the formation of this complex, the expression of $Z_1$ is required, that is also inhibited by the reference signal $R$. For this reason, the control signaling molecule $Q_{u2}$, produced at a rate proportional to the concentration of $Z_2$, will be produced at high rate and released only when R inside the Deactivator is low and the $Q_y$ is high [11].

Then, being $z_1$and $z_2$ the concentrations of the species $Z_1$: $Q_y$ and $Z_2$ in the Deactivators, their dynamics are described as:

$$\dot{z}_1 = \left(\alpha_{z1,r}^0 + \alpha_{z1,r} \cdot \frac{\theta_{r,z1}^{n_{r,z}}}{\theta_{r,z1}^{n_{r,z}} + (\gamma^d)^{n_{r,z}}}\right) \cdot \left(\alpha_{z1,q}^0 + \alpha_{z1,q} \cdot \frac{(q_y^d)^{n_{q,z}}}{\theta_{q,z1}^{n_{q,z}} + (q_y^d)^{n_{q,z}}}\right) - \gamma^d \cdot z_1 \quad (19)$$

$$\dot{z}_1 = \alpha_{z2}^0 + \alpha_{z2} \cdot \frac{z_1^{n_z}}{\theta_{z1}^{n_z} + z_1^{n_z}} - \gamma^d \cdot z_2 \quad (20)$$

With a similar reasoning, being $w_1$and $w_2$ the concentrations of the species $W_1$: $R$ and $W_2$ they defined the Activator dynamics as:

$$\dot{w}_1 = \left(\alpha_{w1,r}^0 + \alpha_{w1,r} \cdot \frac{\theta_{q,w1}^{n_{q,w}}}{\theta_{q,w1}^{n_{q,w}} + \left(q_y^a\right)^{n_{q,w}}}\right) \cdot \left(\alpha_{w1,r}^0 + \alpha_{w1,r} \cdot \frac{(r^d)^{n_{r,w}}}{\theta_{r,w1}^{n_{r,w}} + (r^d)^{n_{r,w}}}\right) - \gamma^a \cdot w_1 \quad (21)$$

$$\dot{w}_2 = \alpha_{w2}^0 + \alpha_{w2} \cdot \frac{w_1^{n_w}}{\theta_{w1}^{n_w} + w_1^{n_w}} - \gamma^w \cdot w_2 \quad (22)$$

Then, they defined the intracellular communication describing the evolution of the concentrations in the signaling molecules inside a type j cell for i=1,2 as:

$$\dot{r}^j = \eta \cdot \left(r^e - r^j\right) - \gamma^j \cdot r^j \quad (23)$$

$$\dot{q}_{ui}^j = f_{ui}^j + \eta \cdot \left(q_{ui}^e - q_{ui}^j\right) - \gamma^j \cdot q_{ui}^j \quad (24)$$

$$\dot{q}_y^j = f_y^j + \eta \cdot \left(q_y^e - q_y^j\right) - \gamma^j \cdot q_y^j \quad (25)$$

Where the production rates $f_{ui}^{j}$ and $f_{y}^{j}$ are defined as:

$$f_y^t := k_y \cdot x_1, \qquad k_y > 0 \tag{26}$$

$$f_{u2}^d := k_{u2} \cdot z_2, \quad k_{u2} > 0 \tag{27}$$

$$f_{u1}^a := k_{u1} \cdot w_2, \quad k_{u1} > 0 \tag{28}$$

$$f_{ui}^t, f_{u1}^d, f_y^d, f_{u2}^a, f_y^a = 0 \tag{29}$$

Finally, the concentrations of the reference signal molecule and of the quorum sensing molecules secreted by the three populations in the environment are described as:

$$\dot{r}^e = r_{in}(t) + \eta \cdot (r^a - r^e) + \eta \cdot (r^d - r^e) + \eta \cdot (r^t - r^e) - \gamma^e \cdot r^e \tag{30}$$

$$\dot{q}_{ui}^e = \eta \cdot (q_{ui}^a - q_{ui}^e) + \eta \cdot \left(q_{ui}^d - q_{ui}^e\right) + \eta \cdot (q_{ui}^t - q_{ui}^e) - \gamma^e \cdot q_{ui}^e \tag{31}$$

$$\dot{q}_y^e = \eta \cdot \left(q_y^a - q_y^e\right) + \eta \cdot \left(q_y^d - q_y^e\right) + \eta \cdot \left(q_y^t - q_y^e\right) - \gamma^e \cdot q_y^e \tag{32}$$

Where $r_{in}(t)$ and $\gamma^e$ represent the concentration of the reference signal provided externally and the degradation rate in the environment, respectively [11].

Once they had all the model defined, they identified several required conditions needed to ensure the correct behavior of the system and then describe the in-silico experiments implemented to validate the effectiveness of the design using BSim, a realistic agent-based simulator of bacterial populations. The results of this experiments can be seen in Figure 14 and Figure 15, where the controller cells are depicted in red and green and the Target cells in blue. These results prove the efficacy of the system as, it can be clearly seen that, the Controller cells successfully toggle the Target cells from their active state to their inactive state and vice versa [11].

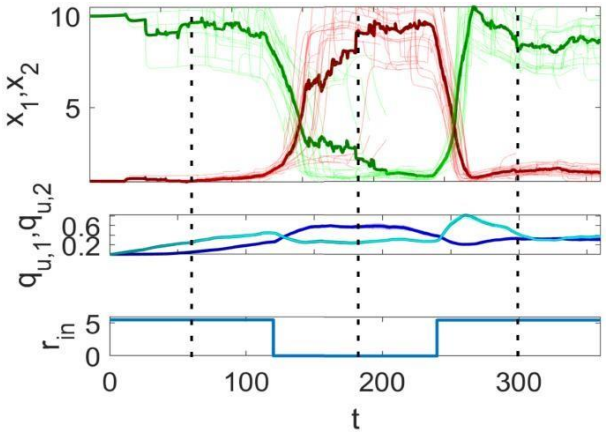


*Figure 14: Evolution of the average (thick lines) and single cell (thin lines) values of the concentrations of $x_1$ (green) and $x_2$ (red) in the Target population (top panel) when the reference signal $r_{in}(t)$ (bottom panel) is switched from high to low and vice versa. The middle panel shows the average and individual values of the concentrations of quorum sensing molecules $q_{u,1}^t$ (light blue) and $q_{u,2}^t$ (dark blue) inside the Target cells [11].*

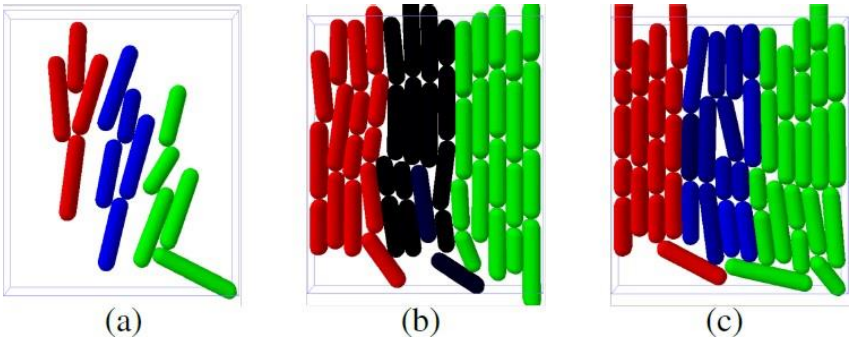


*Figure 15: Snapshots of an agent-based simulation at different time instants (highlighted in Figure 8 with dashed vertical lines). Specifically, panel (a) corresponds to t = 60, panel (b) to t = 180 and panel (c) to t = 300. Activator cells are shown in green, Deactivator cells in red and Target cells are depicted in blue when they are active and black when they are inactive [11].*

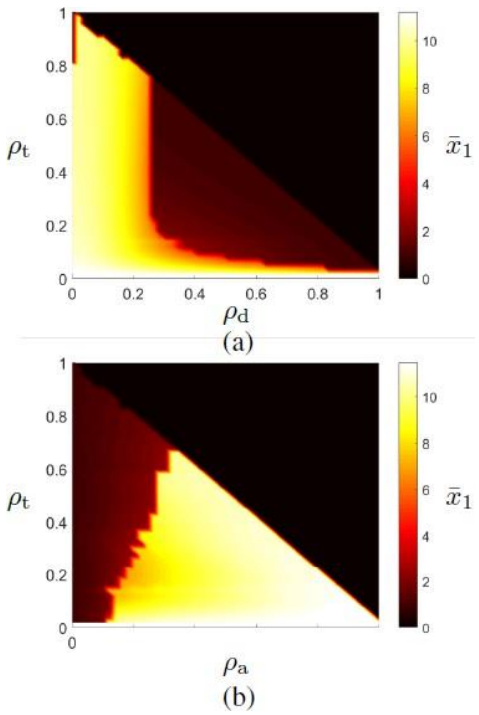


*Figure 16:* (a): Steady-state values of x1 as a function of the ratios of Targets $\rho_t$ and Deactivators $\rho_d$, with Targets starting from ON state and setting $r_{in} = r^{OFF}$. (b): Steady-state values of x1 as a function of the ratios of Targets $\rho_t$ and Activators $\rho_a$, with Targets starting from OFF state and setting $r_{in} = r^{ON}$ [11].

Finally, they performed robustness analysis in Matlab. First, they evaluated the effects the variation of the ratio between Targets and Controllers in the consortium, representing in Figure 16 the steady-state values of $x_1$ when switching Targets OFF (Figure 16a) and ON, (Figure 16b) by the action of the reference signal $r_{in}$ while the ratios of the cell populations in the consortium were being varied. These results demonstrated that, for a wide range of population densities (red region for Deactivators in Figure 16a, white region in Figure 16b for Activators), the Controllers are able to switch the state of Target cells.

Then, they evaluated the effects of the increase in the heterogeneity among cells in each population, perturbating the nominal values of all the parameters. As they showed in Figure 17, even with considerable parameter mismatch, the Controllers are able to induce the switch of state of a large fraction of the Target cells, showing the Activators an even better performance [11].

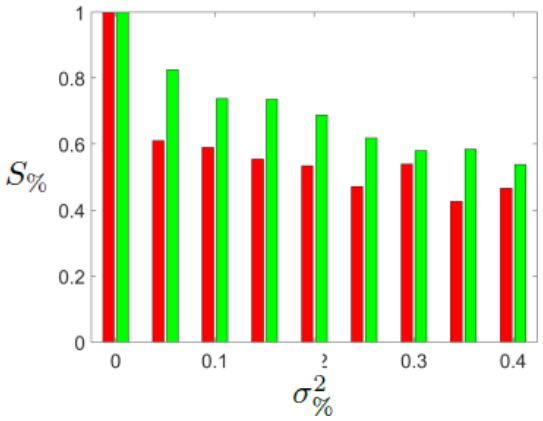


*Figure 17: Percentage of successfully switched Targets (S%) in a balanced population as a function of heterogeneity $\alpha^2_{\%}$. The bar plot in red (green) represents the percentage at steady state of Targets that, starting from ON (OFF) state, are turned OFF (ON) following the reference input rin being switched to $r_{in}^{OFF}$ ($r_{in}^{ON}$) [11].*

## 2.4. Stable co-existence in microbial consortia

Being able to engineer synthetic ecosystems, multicellular synthetic gene circuits, implies a huge advance in synthetic biology. The creation of synthetic microbial consortia, where synthetic populations grow and work together, enables the design of novel, complex and advanced functionalities. However, it also implies the emergence of new challenges.

The main problem that appears working with synthetic microbial consortia is the difficulty to determine the conditions under which different cell populations will survive. Each type of cell has certain conditions under which they grow best. When different cell populations, with diverse suitable conditions, come together in the same microbial consortium, if the chosen conditions for the whole consortium are not carefully balanced, the populations will tend to compete against each other until finally one of them dies. For this reason, the stabilization of the population ratios is a key factor when dealing with engineered microbial consortia.

The issue of creating a stable co-existence in microbial consortia, has been addressed by different research groups with diverse approaches in the last years. In the following subsections three different approaches of this issue will be reviewed.

### 2.4.1. Population regulation in microbial consortia using dual feedback control

The researchers of the Department of Control and Dynamical Systems from the California Institute of Technology in Pasadena, presented an interesting control strategy that combines the control of the cell strain ratio as well as the total population size in a two-strain system [4].

To achieve that goal, their population regulation circuit design is composed by two separated feedback control loops; the global regulation loop and the co-regulation loop. First of all, the global regulation loop, illustrated in Figure 18a, is in charge of the control of the total population size and consists of three modules. The cell dynamics module, which relies on the cell growth and division processes activated by species G, the communication module, based on the global quorum sensing system and the feedback controller module that compares the output and the reference and acts consequently on the cell growth process to decrease the error. The reference signal is set by internal tuning of the induction rate of species G, which can be sequestered by species D in the controller module to decrease cell growth rate. On the other hand, in the communication module, cells both produce and sense Sg which concentration is proportional to the total cell population. These global quorum sensing signaling molecules diffuse into the cells and activate reactions that produce species D, which binds with species G slowing down the cell growth [4].

Moreover, the co-regulation loop, illustrated in Figure 18b, controls the relative population ratio between the two cell strains (Cell 1 and Cell 2) and is also composed by three different modules. This time the dynamics module regulates the cell rate with the production of toxin T, the communication module is composed by two orthogonal quorum sensing systems S1 and S2, and the feedback controller module compares the two populations and actuates the antitoxin A production. Cell 1 and Cell 2 produce S1 and S2 respectively. In Cell 1 the activation of S1 produces

T which kills Cell1, while A is produced by S2. Antitoxin A sequesters T repressing the death process. This way if one cell strain has a larger population the T production rate inside this cell strain is bigger than the A production rate stopping its cell growth until they are equal [4].

Finally, in the dual loop, illustrated in Figure 18 c, both regulation loops are coupled and the total and relative population sizes are independently set by two reference signals [4].

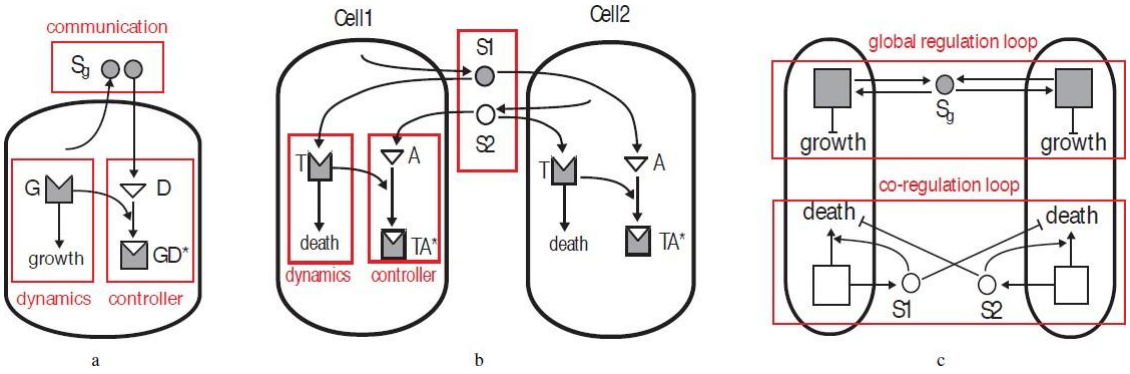


*Figure 18: Biological design of the global regulation loop, of the co-regulation loop, and of the dual loop. a. Global regulation that controls the total population. The reference signal is set by internal tuning of the induction rate of species G. Cells release and sense Sg proportional to the total population. Sg activates the production of D, which binds to G to inhibit cell growth. b. Co-regulation that controls the relative population ratio between two cell strains. Cell strain Cell1 releases and senses S1 that activates the production of toxin T in Cell1. Cell1 also senses S2 released by cell strain Cell2 to activate the production of antitoxin A, which binds with T to inhibit the killing process in Cell1. The negative feedback controller in Cell1 regulates the death dynamics to track the population of Cell2. Same for Cell2.c. It describes the dual loop regulation that couples both global and co-regulation [4].*

Being C, G, D, Sg the cell population, the species that affect cell growth, the species that sequesters it and the global signaling molecules, the model of the control loop that regulates the total size of the populations is defined by the following equations:

$$\dot{C} = k_c \cdot G \cdot \left(1 - \frac{C}{C_{max}}\right) \cdot C - \gamma_C \cdot C \tag{33}$$

$$\dot{G} = g_G - k^+ \cdot G \cdot D - d_G \cdot G \tag{34}$$

$$\dot{D} = g_D + k_D \cdot \frac{S_g^\beta}{K_S + S_g^\beta} - k^+ \cdot G \cdot D - d_D \cdot D \tag{35}$$

$$\dot{S}_g = c_s \cdot C - (d_s + \gamma_S) \cdot S_g \tag{36}$$

Where $k_c$, $C_{max}$, $\gamma_C$ , $g_x$, $k^+$, $d_x$, $k_D$, $\beta$, $K_s$, $c_s$, $d_s$ and $\gamma_S$ represent the cell growth rate constant, the carrying capacity for cell growth, the cell dilution rate constant, the basal production rate of species X (X= G, D), the binding rate of effective annihilation, the dilution rate of species X (X= G, D), the maximal production rate of D, the Hill function coefficient, the dissociation constant for Sg, the synthesis rate of Sg, the degradation rate of Sg and the dilution rate of Sg, respectively [4].

Considering the cell strains Cell 1 and Cell 2 in mixed culture, for {i, j} = {1,2}, being $C_i$, $T^{(i)}$ and $A^{(i)}$ the cell population, the toxin concentration and anti-toxin concentration of cell strain i and $S_i$ and Sj the signaling molecules released by cell strain i and j, the mathematical model of the co-regulation loop is defined by the following equations:

$$\dot{C}_i = k_{ci} \cdot \left(1 - \frac{C_1 + C_2}{C_{max}}\right) \cdot C_i - d_C \cdot T^{(i)} \cdot C_i - \gamma_C \cdot C_i \tag{37}$$

$$\dot{T}^{(i)} = g_T - k_T \cdot \frac{S_i^{\beta}}{K_s + S_i^{\beta}} - k^+ \cdot T^{(i)} \cdot A^{(i)} - d_T \cdot T^{(i)} \tag{38}$$

$$\dot{A}^{(i)} = g_A - k_A \cdot \frac{S_j^{\beta}}{K_s + S_j^{\beta}} - k^+ \cdot T^{(i)} \cdot A^{(i)} - d_A \cdot A^{(i)} \tag{39}$$

$$\dot{S}_i = c_s \cdot C_i - (d_s + \gamma_S) \cdot S_i \tag{40}$$

Where $k_{ci}$, $C_{max}$, $d_C$, $\gamma_C$ , $g_x$, $k_x$ , $\beta$, $K_s$, $k^+$, $d_x$, $c_s$, $d_s$ and $\gamma_S$ represent the cell growth rate constant of each strain, the carrying capacity for cell growth, the cell death rate constant, the cell dilution rate constant, the basal production rate of species X (X= T, A), the maximal production rate of species X (X= T, A), the Hill function coefficient, the dissociation constant for Si, the binding rate of effective annihilation, the dilution rate of species X (X= T, A), the synthesis rate of Si, the degradation rate of Si and the dilution rate of Si, respectively [4].

The simulation results for the dual control loop obtained by Xinying Ren *et al* were very satisfying. They first proved the controller performance and robustness assessing the behavior of the system in response to set-point references of the total population and the population ratio by defining several performance metrics. Assuming some initial conditions, they simulated their model introducing a perturbation in the cell growth rate at some point of the simulation and demonstrated that the lag compensator of the global regulation maintained correctly the relative cell strain ratio in steady state. They also varied the amplitude of the introduced perturbation and measured the population steady states of the cell strains and of the total populations proving that the steady state of the total population always recovers the initial value and that the relative ratio also stabilizes for perturbations of less than 80%, demonstrating the robustness of the dual loop and it capacity of adaptation to perturbations in the cell growth rate, Figure 19 [4].

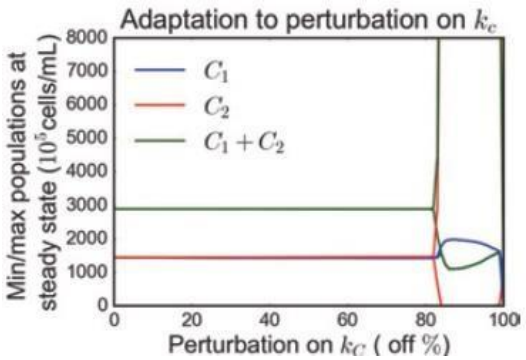


*Figure 19: Perturbations on the growth rate of one cell strain. The dual loop controller shows adaptation to perturbations on the cell growth rate for growth rate perturbations lower than 80% [4]*

Finally, they demonstrated that is possible to independently tuning the total population and the relative strain ratio, showing with Figure 20, that both regulation functions are performed with small steady state errors.

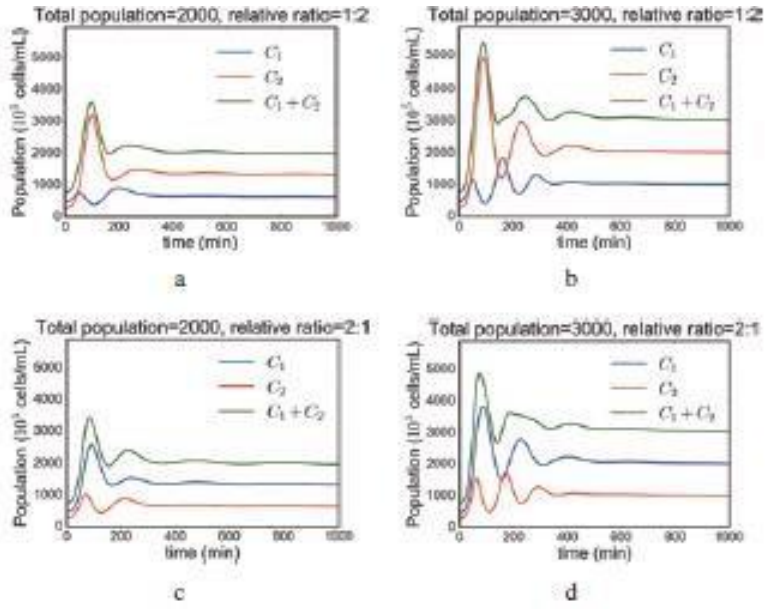


*Figure 20: The dual loop controller dynamics with independently tuned values of the total population and of the relative ratio of the cell strains [4].*

Even if the simulation results demonstrated that the mathematical model proposed in [4] meets the set objectives it also presents certain limitations. The most relevant limitation that shows the model is that there is a symbiotic dependence between the strains, as one produces the antitoxin for the other one, this way if one of the two cell populations suddenly dies out, the other one will irremediably die.

### 2.4.2. Ecolibrium project

Another approach of the ratiometric problem was presented by the students from the Imperial College of London for the 2016 iGEM competition. In their project, called Ecolibrium, they developed a genetic circuit system, a G.E.A.R (Genetically Engineered Artificial Ratio) system, composed by three modules, see Figure 21. First, a communication module which uses two orthogonal quorum sensing systems to allow the bacteria to detect both their own population density and the density of the other population. Secondly, the comparator module, which is in charge of linking the quorum sensing signals to RNA logic enabling the cells to compare the two populations. Third, a growth regulation module that allows the cells to respond the signals from the comparator module. This way, if its own population is bigger than the one of the other cells, a growth inhibiting protein is expressed, promoting the stabilization of the population's ratio [6].

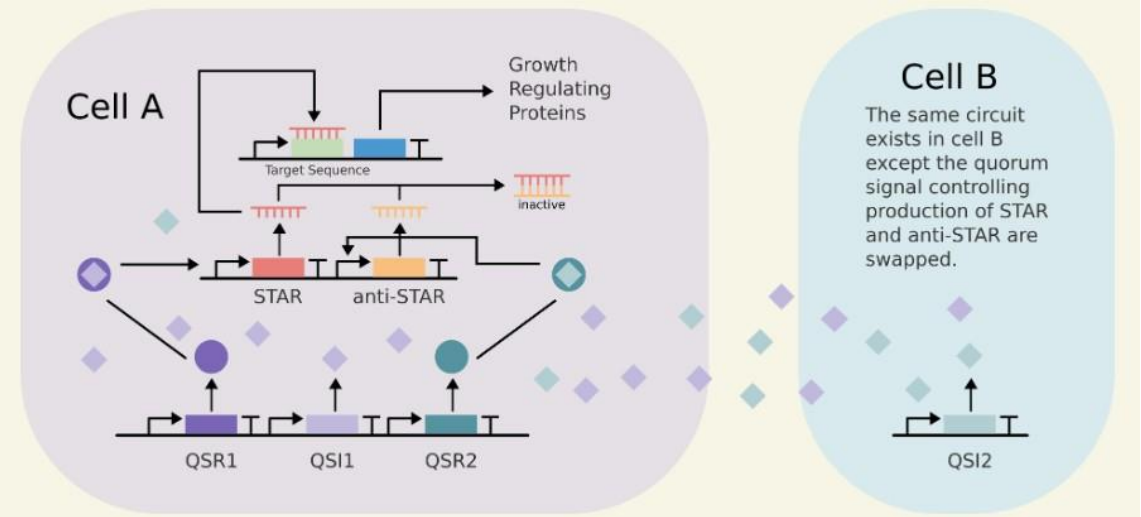


*Figure 21: Representation of the full circuit with the three modules [6]*

In order to fulfil all the characteristics explained above, they first design a single cell level model which described the intracellular interactions, their system reactions can be seen in the section Single Cell Model of [6]. Then a population level model was created, based on the single level model, which not only predicts the ratio between two populations, but also allows the prediction of the ratio between two populations when modifying some key controllable parameters. The model consists of two compartments, the two populations, housed within a bigger compartment that represents the extracellular environment. The communication between the circuits within each population is encapsulated via the quorum sensing molecules diffusion between the populations and the environment. Each population has also the single cell model inside scaled and their growth rate is determined by their internal dynamics defined by the growth regulation protein. Finally, the growth of the populations is modelled by the Competitive Lotka-Volterra equations. Their whole system reactions and equations can be seen in the Population Model section in [6].

In their simulations, they compared the behavior of a co-culture of two populations with different growth rates inoculated at a 1:1 ratio with and without the GEAR system. In Figure 22, we can see the system dynamics without the GEAR system. It is observable that one of the populations, the one with a higher growth rate, rapidly grows clearly outcompeting the other which disappears having 0 cells, losing the 1:1 ratio. When the default GEAR system is implemented, without tuning the ratio with the key parameter, an important improvement of the ability of maintaining the original 1:1 ratio is observed (Figure 23). Finally, when a key tunable parameter is optimized, the effectiveness of the GEAR system is even more notorious as the system recovers its initial 1:1 ratio faster (Figure 24) [6].

Despite the good results of their simulations, this model presents limitations similar of those of [4]. This synthetic design also encodes a dynamic equilibrium between the two populations. This time, if one of the two populations dies out the other one will experiment an uncontrolled growth.

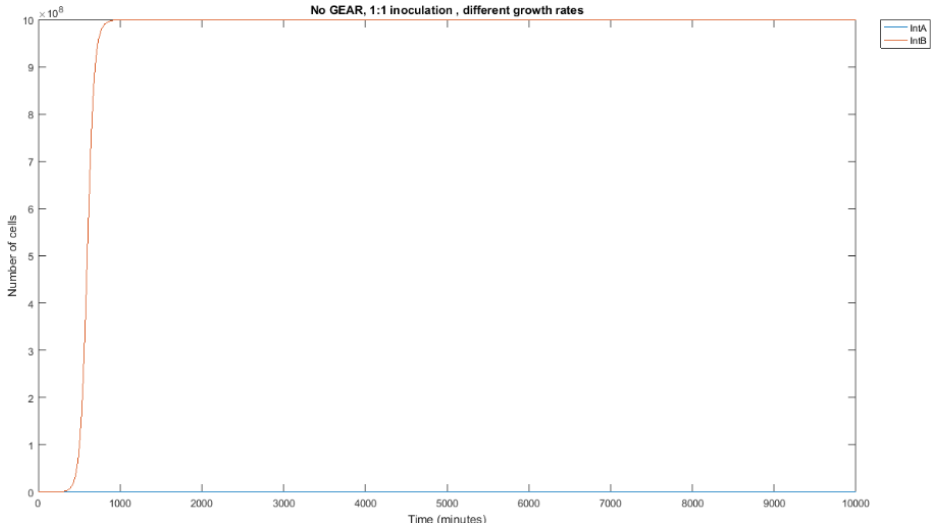


*Figure 22:Population dynamics without a GEAR system implemented [6]*

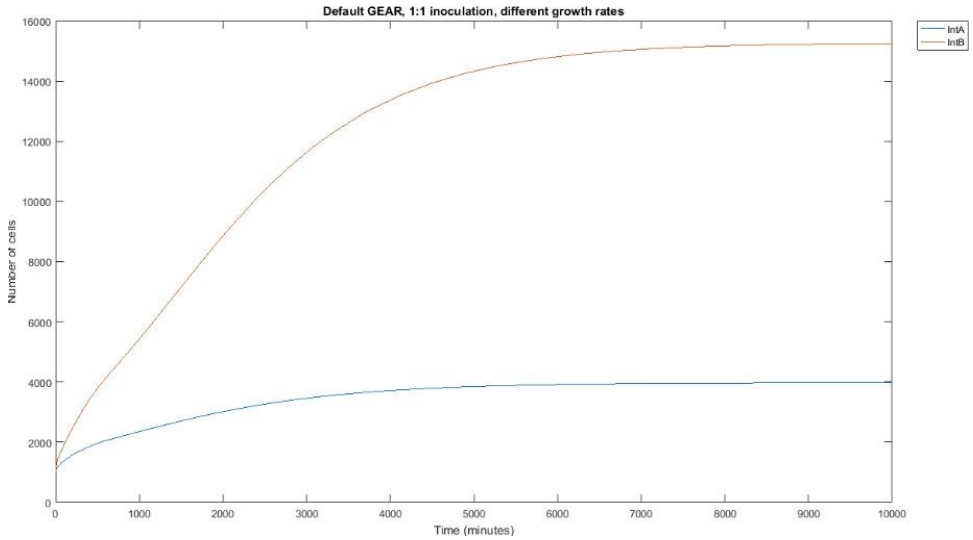


*Figure 23:Population dynamics with a default GEAR system [6]*

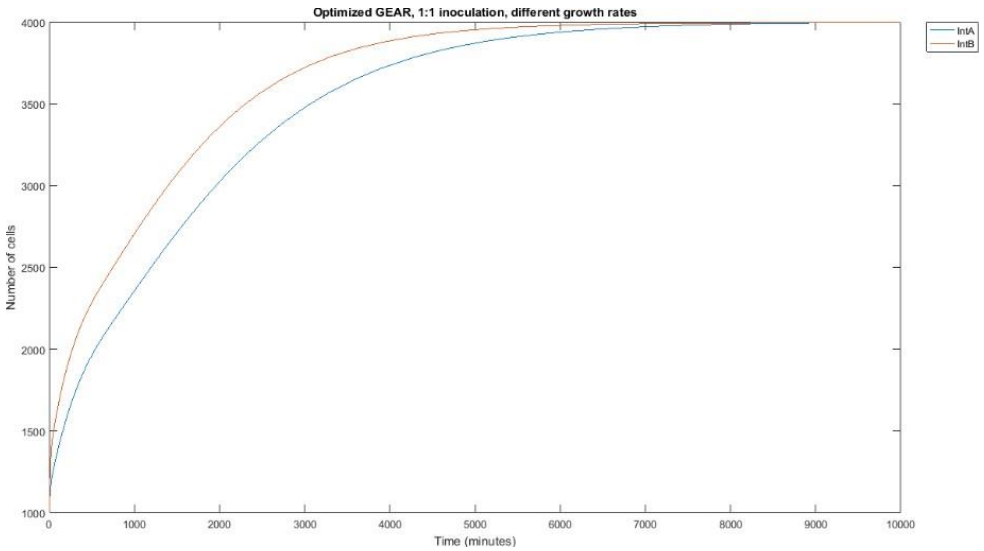


*Figure 24: Population dynamics with an optimized GEAR system [6]*

### 2.4.3. Ratiometric control for differentiation of cell populations endowed with synthetic toggle switches

Another important research, which addresses the problem of regulating the ratio of two cell populations, was published on 2019 by the researcher's team of the University of Naples Federico II. In this case, the two cell populations belong to the same strain and each cell has a bistable memory mechanism, such as a genetic toggle switch, embedded inside that allows them to switch role from one population to another in response to some external control inputs. In order to regulate the populations ratio, they present three different feedback control strategies; Bang-Bang controller, PI controller and model predictive controller (MPC) [5].

First of all, being LacI and TetR the two repressor proteins of the genetic regulatory network of the toggle switch and aTc and IPTG the two inducer molecules whose concentration induces the flip of the toggle switch, they defined the dynamic model of the *i*-th cell in the consortium as follows:

$$\frac{dmRNA_{LacI}^{i}}{dt} = k_L^{m0} + k_L^m \cdot \Phi_T(t) - \gamma_L^m \cdot mRNA_{LacI} \tag{41}$$

$$\frac{dmRNA_{TetR}^{i}}{dt} = k_T^{m0} + k_T^m \cdot \Phi_L(t) - \gamma_T^m \cdot mRNA_{TetR} \tag{42}$$

$$\frac{dLacI^i}{dt} = k_L^p \cdot mRNA_{LacI}^i - \gamma_L^p \cdot LacI^i \tag{43}$$

$$\frac{dTetR^i}{dt} = k_T^p \cdot mRNA_{TetR}^i - \gamma_T^p \cdot TetR^i \tag{44}$$

$$\frac{daTc^i}{dt} = k_{aTc} \cdot (u_a - aTc^i) \tag{45}$$

$$\frac{dIPTG^i}{dt} = k_{IPTG} \cdot (u_p - IPTG^i) \tag{46}$$

Where the state variables denote concentrations of molecules inside the cell and the parameters $k_{L/T}^{m0}, k_{L/T}^m, k_{L/T}^p, \gamma_{L/T}^m, \gamma_{L/T}^p, k_{aTc/IPTG}$ are the leakage transcription, transcription, translation, mRNA degradation and protein degradation rates and diffusion rates of the inducers across the cell membrane, respectively. Variables $u_a$ and $u_p$ denote concentrations of the inducer molecules in the growth medium and also represent the control inputs common to every cell in the populations [5]. $\Phi_T$ and $\Phi_L$ represent the input effects and are modelled as follows:

$$\Phi_T(t) := \frac{1}{1+\left(\frac{TetR^i}{\theta_{TetR}} \cdot \frac{1}{1+\left(\frac{aTc^i}{\theta_{aTc}}\right)^{\eta_{TetR}}}\right)} \tag{47}$$

$$\Phi_T(t) \coloneqq \frac{1}{1+\left(\frac{TetR^i}{\theta_{TetR}} \cdot \frac{1}{1+\left(\frac{aTc^i}{\theta_{aTc}}\right)^{\eta TetR}}\right)} \tag{48}$$

Once the model is defined, considering $N_t$ the finite set of all cells in the consortium at time t and **A** and **B** the two possible stable equilibria of the genetic toggle switch (the two populations in the strain), each associated with the full expression of TetR and LacI, respectively, they identified the three sets the cells can belong to:

$$A_t \coloneqq \{i \in N_t : TetR^i(t) > 2LacI^i(t)\} \tag{49}$$

$$B_t \coloneqq \{i \in N_t : LacI^i(t) > 2TetR^i(t)\} \tag{50}$$

$$C_t \coloneqq \{i \notin A_t \notin B_t\} \tag{51}$$

This way, at time t, a cell will belong to population A if $i \in A_t$ or to population **B** if $i \in B_t$ [5]. Finally, being $r_A(t) = \frac{n_A(t)}{N(t)}$ and $r_B(t) = \frac{n_B(t)}{N(t)}$ the ratio off cells that belong to population **A** and **B**, respectively, they state that given a consortium whose dynamics are described by the model above and a desired ratio $r \in [0,1]$ the control law $u(t) = [u_a(t), u_p(t)]^T$ solves the ratiometric control problem if, for some small positive constant $\epsilon$:

$$\lim_{t\to\infty} |e_A(t)| < \epsilon \quad and \quad \lim_{t\to\infty} |e_B(t)| < \epsilon \tag{52}$$

$$where\ e_B(t) = r - r_B(t)\ and\ e_A(t) = (1 - r) - r_A(t)$$

In order to solve the ratiometric problem, they analyze the realistic and technological constrains that occur in experimental microfluidic platforms, and conclude that there are two possible implementations for the input signals $u(t) = \left[u_a(t), u_p(t)\right]^T$ that can be generated by the actuators: The T-Junction implementation and the Dial-A-Wave (DAW) system. Using these implementations, they finally define the three controllers. First, in the Bang-Bang controller implemented via a T-junction, secondly, the PI controller implemented via DAW and lastly the Model Predictive Controller also implemented via DAW [5].

With all the controllers already defined, they begin with the in-silico testing. After fixing some control inputs, they started with a batch of numerical simulations in Matlab, obtaining the evolution of the errors, $e_A(t)$ and $e_B(t)$, and of the inputs $u_a(t)$, $u_p(t)$ (Figure 25). The Bang-Bang controller shows good performance, Figure 25 (a) and (b). However, the PI controller shows half the settling time and lower error values at the steady state, compared with the Bang-Bang controller, Figure 25 (c) and (d). Finally the MPC controller seems to be the best strategy with a settling time almost 40% shorter than the one of PI, Figure 25 (e) and (f) [5]

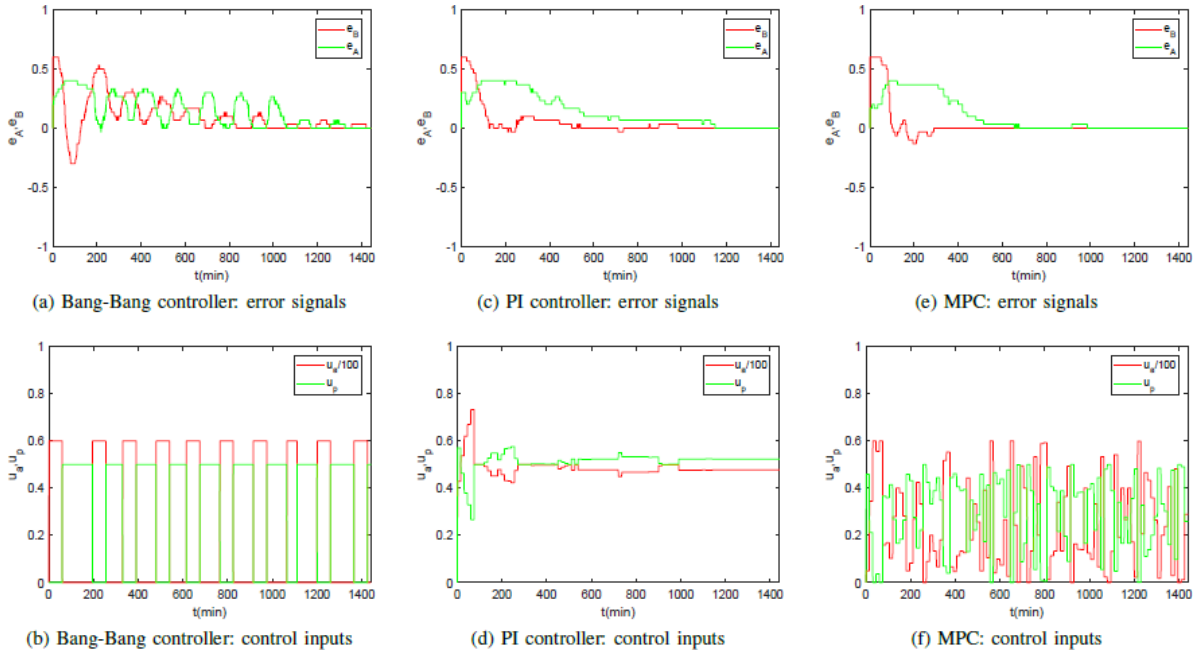


*Figure 25: Evolution in time of the error signals and the control inputs [5].*

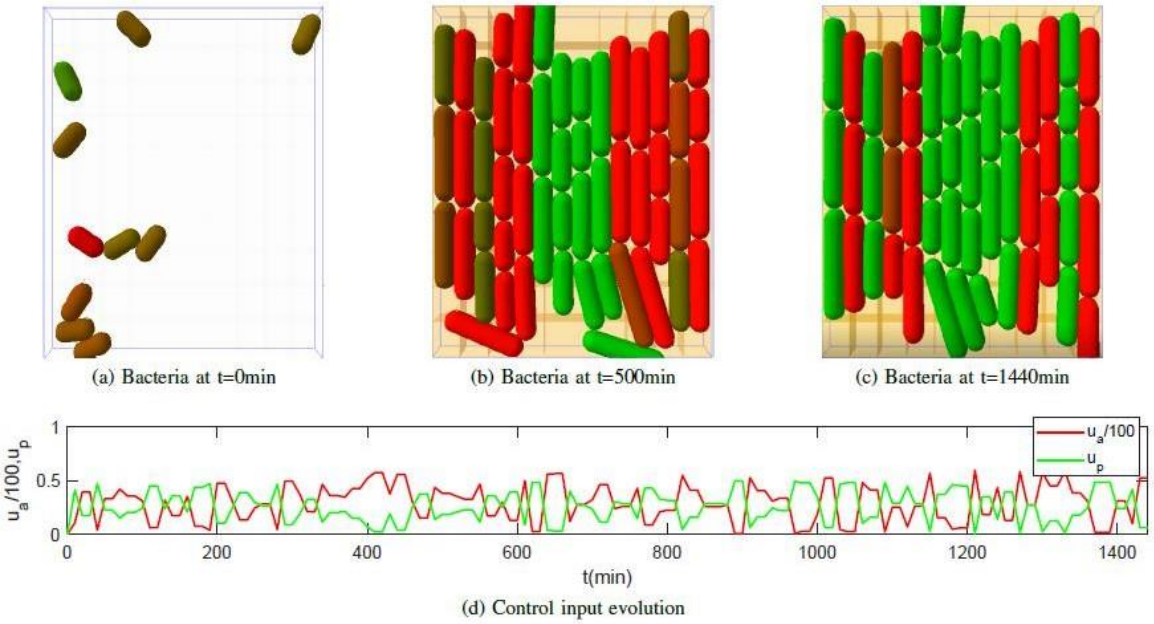


*Figure 26: Snapshots of a BSim simulation performed using the MPC algorithm. Red cells belong to $A_t$, green cells to $B_t$ and intermediate colored cells to $C_t$. The bottom panel shows the corresponding control input [5].*

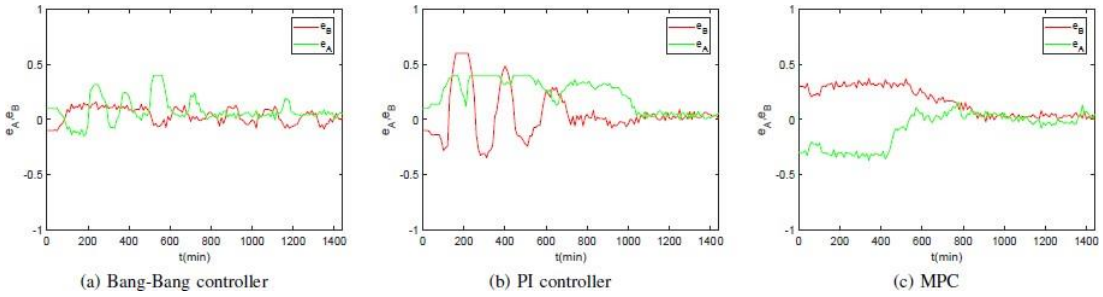


*Figure 27: BSim simulations: evolution in time of the error signals [5].*

To obtain more accurate simulations they used BSim extended with an Euler-Maruyama solver. In Figure 26 (a), (b) and (c) we can see the snapshots of the simulation where the red cells represent the bacteria belonging to $A_t$, the green cells the bacteria belonging to $B_t$ and the ones with intermediate coloration the bacteria belonging to $C_t$ .Then, in Figure 27 it is represented the evolution of the error signals $e_A(t)$ and $e_B(t)$. Comparing with the error signals evolution of the Matlab simulations, the BSim simulations show bigger fluctuations, but the average error evolution is qualitative the same, what proves the good performance of the controllers [5].

The model proposed sorts out the problem of dynamic equilibrium as is intrinsically robust to extinction events. Furthermore, this research provides a very interesting approach if you are interested in controlling the populations ratio externally or in having the possibility of changing the ratios online in real time. Unfortunately, this approach is not suitable in situation in which the final goal is to have a self-regulated system in which the cells in the populations have the capacity to autoregulate their ratios, as you must provide the system with this external control inputs.

# 3. Self-regulation of cell population model

## 3.1. Mathematical model

In this project we propose a novel approach to the problem of stable co-existence in microbial consortia by means of self-regulation of the cell population. The aim is to develop an alternative strategy to those presented in section *Stable co-existence in microbial consortia*, for the stabilization of the relative population numbers (i.e. ratio) in microbial consortium. With this objective in mind, we consider the case in which there exists a bistable memory mechanism inside each cell, such as the genetic toggle switch, whose current internal state defines which of the two possible populations the cell belongs to. Then, by communicating via quorum sensing molecules the cells are capable of self-regulating their ratios. This way, if one of the two populations increases its number of cells with respect to the other, the system can stabilize the situation, equalizing both populations. This desired behavior is illustrated in Figure 28, where it can be seen how an initially unbalanced consortium, that is with different number of cells belonging to each population, can reach the stabilization of the ratio after some time t.

The genetic toggle switch consists, as usual, of two repressor proteins, let's say $P_1$ and $P_2$, that repress each other's expression, so that only one of them is fully expressed at any time. This way, if a cell is expressing P1 due to the current state of the toggle switch, the cell belongs to Population 1 and vice versa. However, in order to accomplish the objectives explained above, the system must include not only the genetic toggle switch but also some communication mechanism to ensure the communication between all the cells in the consortium and a biological comparator in charge of sensing the unbalance in the consortium and then acting accordingly on the toggle switch.

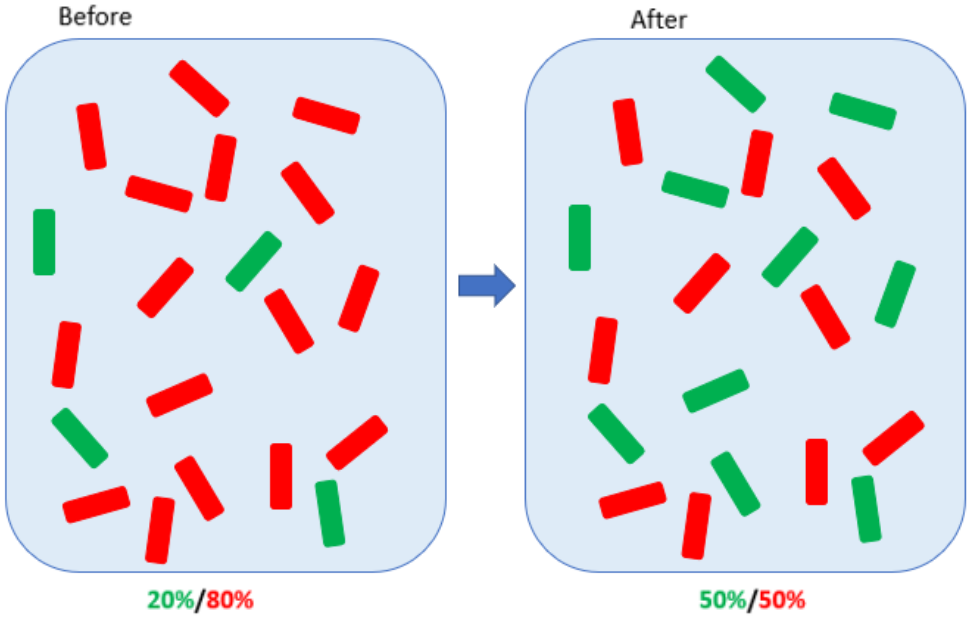


*Figure 28: Abstract representation of the control scenario. The red and green rectangles represent the cells belonging to population i and j respectively. Number of cells N=20. On the left it is represented the initial state of the unbalanced consortium whose population ratio is different from 0.5 (16 red cells and 4 green cells). On the right it is represented the state of the consortium after some time t. The system has self-regulated the cell populations reaching the desired ratio of 0.5 (10 red and green cells).*

As explained above, we need to extra ingredients:

- Communication mechanism: The communication between the cells is realized by means of a quorum sensing mechanism [16]. Each cell secretes a small signaling molecule which diffuses in and out the cell, this way these molecules have different concentrations inside each cell and outside of them. Thus, the concentration of signaling molecules in the environment is proportional to the densities of each population.
- Comparator and actuator: To compare the sensed populations densities, we need two molecules, where one molecule sequesters the other inhibiting its function. Finally, the remainder of subtracting the expression of the two molecules is expressed.

The representation of the biological implementation is illustrated in Figure 29:.This scheme represents the auto-regulation system inside each cell and the communication between all the cells through the environment in a two populations consortium. In the consortium, each cell will express either $P_1$ (i.e. $P_1 >> P_2$) or $P_2$ (i.e. $P_2 >> P_1$), which defines the current population they belong to, Population 1 or Population 2, respectively. We denote with $N_1$ and $N_2$ the number of cells belonging to Population 1 and Population 2, respectively (being the total population of the consortium $N= N_1 + N_2$). The quorum sensing molecules $S_1$ and $S_2$ are produced proportionally to $P_1$ and $P_2$. The concentration of molecules $S_1$ and $S_2$ inside a cell are $S_1^{in}$ and $S_2^{in}$ and the global concentration of $S_1$ and $S_2$ in the environment are $S_1^e$ and $S_2^e$. At steady state, the concentrations $S_1^e$ and $S_2^e$, will reach values that are proportional to the densities (and hence the number) of cells in each population. Such quantities are used by the control system as "measures" of the current populations ratio. These molecules activate the production of molecules $AP_1$ (Anti- $P_1$) and $AP_2$ (Anti- $P_2$). These two species combine together in an inert molecule annihilating each other. If the two populations are not balanced, one of these two molecules, either $AP_1$ or $AP_2$, is produced in excess and it will influence the expression of the toggle switch. Specifically, if $AP_1$ is produced in excess, it will promote $P_2$ and thus making the cell pass from Population 1 to Population 2. Vice versa for $AP_2$.

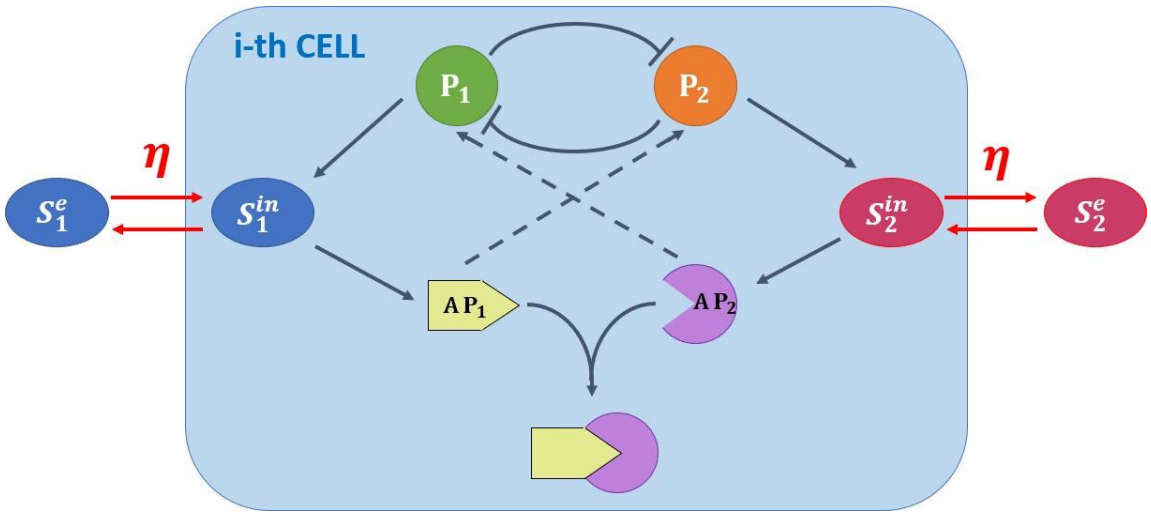


*Figure 29:Abstract biological implementation of the self-regulation of cell population model. Each cell expresses protein $P_1$ or $P_2$ (circles) depending on the current internal state of the toggle switch. Proportional to these proteins they produce the quorum sensing molecules (ovals) $S_1$ and $S_2$ whose internal and external concentrations are $S_i^{in}$ and $S_i^e$ ,rrespectively. These molecules activate the production of the comparator molecules $AP_1$ and $AP_2$ (Yellow and violet) and the remainder of the subtraction of both quantities promotes the expression of the protein $P_i$ related with the population with a smaller number of cells.*

For example, if a cell belongs to Population 1 then $P_1 > P_2$ inside the cell and as the quorum sensing signaling molecules, $S_1^{in}$ and $S_2^{in}$, are produced proportional to $P_1$ and $P_2$, then $S_1^{in} > S_2^{in}$ inside the cell increasing the proportion of $S_1^e$ in the environment. If in the consortium $N_1 > N_2$, then $S_1^e > S_2^e$ in the environment. In this situation, inside the cells $AP_1 > AP_2$, meaning that $AP_2$ is practically absent and the remaining portion of $AP_1$ (that is $AP_1$ - $AP_2$) will promote the expression of $P_2$. When this happens inside a cell belonging to Population 1 and if this action is strong enough, $P_2$ will increase and the state of the toggle switch will change, turning the cell into a cell belonging to Population 2. This way, more cells will change their internal state from $P_1$ to $P_2$ balancing the cell population in the consortium.

By $P_1$ and $P_2$ the concentrations of molecules $P_1$ and $P_2$ in each cell, the dynamical model of the toggle switch is given by:

$$\dot{P}_1 = \alpha_p^o + \frac{\alpha_p}{1+\left(\frac{P_2}{\theta_p}\right)^{n_p}} - d_p \cdot P_1 + U_1 \tag{53}$$

$$\dot{P}_2 = \alpha_p^o + \frac{\alpha_p}{1+\left(\frac{P_1}{\theta_p}\right)^{n_p}} - d_p \cdot P_2 + U_2 \tag{54}$$

All the parameters are described in Table 1. This model was created taking as reference the toggle switch model described in [11], which is based on the Hill function for a repressor. This model also includes the degradation of the proteins inside the cell, $-d_p \cdot P_i$, and $U_1$ and $U_2$ which represent the effect of two inputs that can be used to toggle the switch between one state and the other. These inputs are defined as follows:

$$U_1 = \beta_u \cdot \frac{A_{p2}^{n_u}}{\theta_u^{n_u} + A_{p2}^{n_u}} \tag{55}$$

$$U_2 = \beta_u \cdot \frac{A_{p1}^{n_u}}{\theta_u^{n_u} + A_{p1}^{n_u}} \tag{56}$$

This equations describe how the inputs of the toggle-switch $U_1$ and $U_2$ are associated with the concentrations of the molecules $AP_2$ and $AP_1$, that is $A_{p2}$ and $A_{p1}$, respectively, as shown in Figure 29.

Then, in order to define the dynamical equations for the comparator molecules $AP_1$ and $AP_2$ we took as reference the model of the co-regulation loop of [4]:

$$\dot{A}_{p1} = \alpha_{Ap}^o + \alpha_{Ap} \cdot \frac{S_1^{n_{Ap}}}{\theta_s + S_1^{n_{Ap}}} - K \cdot A_{p1} \cdot A_{p2} - d_{Ap} \cdot A_{p1} \tag{57}$$

$$\dot{A}_{p2} = \alpha_{Ap}^o + \alpha_{Ap} \cdot \frac{S_2^{n_{Ap}}}{\theta_s{}^{n_{Ap}} + S_2^{n_{Ap}}} - K \cdot A_{p1} \cdot A_{p2} - d_{Ap} \cdot A_{p2} \tag{58}$$

These equations describe how the production of the molecule $AP_i$ is promoted by the quorum sensing molecule $S_i$. At the same time, the concentration of $AP_i$ decreases as the compound $AP_2$:$AP_1$ is produced and because of the degradation inside the cell.

Next, we defined the equations to describe the evolution of the concentrations of the quorum sensing molecules inside the cells, $S_1^{in}$ and $S_2^{in}$, using as reference both [11] and [4]:

$$\dot{S}_1^{in} = c_s \cdot P_1 - d_s \cdot S_1^{in} + \eta \cdot (S_1^e - S_1^{in}) \quad (59)$$

$$\dot{S}_2^{in} = c_s \cdot P_2 - d_s \cdot S_2^{in} + \eta \cdot (S_2^e - S_2^{in}) \quad (60)$$

As described in Figure 29, the production of $S_1^{in}$ and $S_2^{in}$ is proportional to the expression of $P_1$ and $P_2$, respectively. Simultaneously, they are degraded due to cell division. The last term in the equations represents the diffusion of the quorum sensing molecules across the cell membrane.

On the other hand, taking again the research [11] as reference, the concentrations of the quorum sensing molecules secreted by the cells in the environment, $S_1^e$ and $S_2^e$, evolve according to:

$$\dot{S}_1^e = \eta \cdot \sum_{k=1}^{N} \left(S_1^{in,k} - S_1^e\right) - \gamma_e \cdot S_1^e \quad (61)$$

$$\dot{S}_2^e = \eta \cdot \sum_{k=1}^{N} \left(S_2^{in,k} - S_2^e\right) - \gamma_e \cdot S_2^e \quad (62)$$

Where instead of having degradation due to cell division, $S_1^e$ and $S_2^e$ are diluted due to wash out from the chemostat.

*Table 1: Parameters of the self-regulation control system*

| | |
|---|---|
| $\alpha_p^o$ | Basal expression rate of species P |
| $\alpha_p$ | Maximal expression rate of species P |
| $\theta_p$ | Activation coefficient of species P |
| $n_p$, $n_u$, $n_{Ap}$ | Hill coefficients |
| $d_p$, $d_{Ap}$, $d_s$ | Degradation rates inside the cell |
| $\beta_u$ | Maximal promoter activity coefficient |
| $\theta_u$ | Activation coefficient of species U |
| $\alpha_{Ap}^o$ | Basal expression rate of species Ap |
| $\alpha_{Ap}$ | Maximal expression rate of species Ap |
| $\theta_s$ | Dissociation constant of species S |
| $K$ | Binding rate of effective annihilation |
| $c_s$ | Synthesis rate of $S_{in}$ |
| $\eta$ | Diffusion rate across cell membrane |
| $\gamma_e$ | Dilution rate in the environment |

## 3.2. Non dimensional model

The complexity of the model, described in the previous equations, makes difficult to select the good values for the parameters. For this reason, we decided to simplify the model assuming several hypotheses. First of all, we assumed that the species P and Ap don't have a basal expression level (Equation 63). Then, we considered that all the species had the same Hill coefficients (Equation 64) and degradation rates (Equation 65).

$$\alpha_p^o = \alpha_{Ap}^o = 0 \tag{63}$$
$$n_p = n_u = n_{Ap} = n \tag{64}$$
$$d = d_p = d_{Ap} = d_s \tag{65}$$

Introducing all these considerations in equations 53 - 62 we obtain the following system of equations, being (i, j) = (1, 2):

$$\frac{dP_i}{dt} = \frac{\alpha_p}{1+\left(\frac{P_j}{\theta_p}\right)^n} - d \cdot P_i + \beta_u \cdot \frac{A_{pj}^n}{\theta_u^n + A_{pj}^n} \tag{66}$$

$$\frac{dA_{pi}}{dt} = \alpha_{Ap} \cdot \frac{S_i^n}{\theta_s + S_i^n} - K \cdot A_{pi} \cdot A_{pj} - d \cdot A_{pi} \tag{67}$$

$$\frac{dS_i^{in}}{dt} = c_s \cdot P_i - d \cdot S_i^{in} + \eta \cdot (S_i^e - S_i^{in}) \tag{68}$$

$$\frac{dS_i^e}{dt} = \eta \cdot \sum_{k=1}^{N} \left(S_i^{in,k} - S_i^e\right) - \gamma_e \cdot S_i^e \tag{69}$$

Where the equations 66 – 68 hold for every k-th cell while the Equation 69 represents the concentrations of the quorum sensing molecules secreted by the cells in the environment.

Then using the dimensionless state variables $\tilde{P}_i = \frac{P_i}{\theta_p}$, $\tilde{A}_{pi} = \frac{A_{pi}}{\theta_u}$, $\tilde{S}_i^{in} = \frac{S_i^{in}}{\theta_s}$, $\tilde{S}_i^e = \frac{S_i^e}{\theta_s}$ and time $t' = d \cdot t$, we obtain the final simplified model as follows:

1)

$$\frac{d\tilde{P}_i}{dt'} = \frac{dP_i}{dt} \cdot \frac{1}{\theta_p \cdot d} = \frac{1}{\theta_p \cdot d} \cdot \frac{\alpha_p}{1+\left(\tilde{P}_j\right)^n} - \tilde{P}_i + \frac{1}{\theta_u \cdot d} \cdot \beta_u \cdot \frac{\left(\tilde{A}_{pj}\right)^n}{1+\left(\tilde{A}_{pj}\right)^n} \tag{70}$$

Considering $\tilde{\alpha}_p = \frac{\alpha_p}{\theta_p \cdot d}$ and $\tilde{\beta}_u = \frac{\beta_u}{\theta_u \cdot d}$ the final simplification of the Equation 67 is:

$$\frac{d\tilde{P}_i}{dt'} = \frac{\tilde{\alpha}_p}{1+\left(\tilde{P}_j\right)^n} - \tilde{P}_i + \tilde{\beta}_u \cdot \frac{\left(\tilde{A}_{pj}\right)^n}{1+\left(\tilde{A}_{pj}\right)^n} \tag{71}$$

2)

$$\frac{d\tilde{A}_{pi}}{dt\prime} = \frac{dA_{pi}}{dt} \cdot \frac{1}{\theta_u \cdot d} = \frac{1}{\theta_u \cdot d} \cdot \alpha_{Ap} \cdot \frac{\left(\tilde{S}_i^{in}\right)^n}{1+\left(\tilde{S}_i^{in}\right)^n} - \frac{1}{\theta_u \cdot d} \cdot K \cdot {\theta_u}^2 \cdot \tilde{A}_{pi} \cdot \tilde{A}_{pj} - \tilde{A}_{pi} \qquad (72)$$

Considering $\tilde{\alpha}_{\mathrm{Ap}} = \frac{\alpha_{Ap}}{\theta_u \cdot d}$ and $\tilde{\mathrm{K}} = \frac{\mathrm{K} \cdot \theta_u}{d}$ the final simplification of the Equation 68 is:

$$\frac{d\tilde{A}_{pi}}{dt\prime} = \tilde{\alpha}_{Ap} \cdot \frac{\left(\tilde{S}_i^{in}\right)^n}{1+\left(\tilde{S}_i^{in}\right)^n} - \tilde{K} \cdot \tilde{A}_{pi} \cdot \tilde{A}_{pj} - \tilde{A}_{pi} \qquad (73)$$

3)

$$\frac{d\tilde{S}_i^{in}}{dt\prime} = \frac{dS_i^{in}}{dt} \cdot \frac{1}{\theta_s \cdot d} = \frac{1}{\theta_s \cdot d} \cdot c_s \cdot \theta_p \cdot \tilde{P}_i - \tilde{S}_i^{in} + \frac{1}{d} \cdot \eta \cdot (\tilde{S}_i^{e} - \tilde{S}_i^{in}) \qquad (74)$$

Considering $\tilde{c}_s = \frac{c_s \cdot \theta_p}{\theta_s \cdot d}$ and $\tilde{\eta} = \frac{\eta}{d}$ the final simplification of the Equation 69 is:

$$\frac{d\tilde{S}_i^{in}}{dt\prime} = \tilde{c}_s \cdot \tilde{P}_i - \tilde{S}_i^{in} + \tilde{\eta} \cdot (\tilde{S}_i^{e} - \tilde{S}_i^{in}) \qquad (75)$$

4)

$$\frac{d\tilde{S}_i^{e}}{dt\prime} = \frac{dS_i^{e}}{dt} \cdot \frac{1}{\theta_s \cdot d} = \frac{1}{d} \cdot \eta \cdot \sum_{k=1}^{N}\left(\tilde{S}_i^{in,k} - \tilde{S}_i^{e}\right) - \frac{1}{d} \cdot \gamma_e \cdot \tilde{S}_i^{e} \qquad (76)$$

Considering $\tilde{\gamma}_e = \frac{\gamma_e}{d}$ and $\tilde{\eta} = \frac{\eta}{d}$ the final simplification of the Equation 70 is:

$$\frac{d\tilde{S}_i^{e}}{dt\prime} = \tilde{\eta} \cdot \sum_{k=1}^{N}\left(\tilde{S}_i^{in,k} - \tilde{S}_i^{e}\right) - \tilde{\gamma}_e \cdot \tilde{S}_i^{e} \qquad (77)$$

The final simplified dynamical model for every k-th cell, being (i, j) = (1, 2), is then:

$$\begin{aligned}
\frac{d\tilde{P}_i}{dt'} &= \frac{\tilde{\alpha}_p}{1+\left(\tilde{P}_j\right)^n} - \tilde{P}_i + \tilde{\beta}_u \cdot \frac{\left(\tilde{A}_{pj}\right)^n}{1+\left(\tilde{A}_{pj}\right)^n} \\
\frac{d\tilde{A}_{pi}}{dt'} &= \tilde{\alpha}_{Ap} \cdot \frac{\left(\tilde{S}_i^{in}\right)^n}{1+\left(\tilde{S}_i^{in}\right)^n} - \tilde{K} \cdot \tilde{A}_{pi} \cdot \tilde{A}_{pj} - \tilde{A}_{pi} \\
\frac{d\tilde{S}_i^{in}}{dt'} &= \tilde{c}_s \cdot \tilde{P}_i - \tilde{S}_i^{in} + \tilde{\eta} \cdot (\tilde{S}_i^{e} - \tilde{S}_i^{in})
\end{aligned} \qquad (78)$$

And the simplified equation for the concentrations of the quorum sensing molecules secreted by the cells in the environment, being (i, j) = (1, 2), is:

$$\frac{d\tilde{S}_i^{e}}{dt\prime} = \tilde{\eta} \cdot \sum_{k=1}^{N}\left(\tilde{S}_i^{in,k} - \tilde{S}_i^{e}\right) - \tilde{\gamma}_e \cdot \tilde{S}_i^{e} \qquad (79)$$

## 3.3. Parameters definition

The first step we made in the selection of the parameter's value, was to find the relation between the parameters $\tilde{\alpha}_p$ and $\tilde{\beta}_u$ of the toggle switch equations (Equation 78.1), which ensures the desired behavior of the toggle switch. For this aim, we analyze the nullclines with the MATLAB function Pplane selecting high values of $\tilde{A}_{pj}$ (so that, $\frac{(\tilde{A}_{pj})^n}{1+(\tilde{A}_{pj})^n} \approx 1$) and look for the critical value (i.e., the bifurcation point) that led to only one equilibrium point. After this analysis we came up with the following condition:

$$\tilde{\beta}_u > \frac{1}{\tilde{\alpha}_p} \tag{80}$$

Next, we took into account that the communication between the cells needs to be faster than the comparator mechanism and the toggle switch, because in this way we make sure that the cell will not change its internal state (that is the population it belongs to) unless it is necessary due to a detected unbalance of the quorum sensing molecules in the environment. Otherwise, if the communication were slower, the comparison process would start inside the cell prematurely and it could promote the change in its state erroneously because the information received from the quorum sensing molecules would not be updated. For this reason, we needed to ensure a very high diffusion rate:

$$\eta \gg d_s = d \rightarrow High\ \tilde{\eta} \tag{81}$$

Finally, following the same reasoning than above we considered that not only the communication block should be the fastest one but also that the comparison block must be faster than the toggle switch to ensure a correct performance of the model. With this rate relation in mind (Communication > Comparison > Toggle Switch) we defined the rest of the parameters (Table 2).

*Table 2: Nominal parameters of the simplified system*

| | |
|---|---|
| $n$ | 2 |
| $\tilde{\alpha}_p$ | 4 |
| $\tilde{\beta}_u$ | 3 |
| $\tilde{\alpha}_{Ap}$ | 1 |
| $\tilde{K}$ | 3.5 |
| $\tilde{c}_s$ | 1 |
| $\tilde{\eta}$ | 10 |
| $\tilde{\gamma}_e$ | 1.5 |

## 3.4. Numerical validation

First of all, taking as reference the problem statement defined in [5], we denoted $\mathbb{N}$ as the finite set of cells in the consortium with $N(t) = |\mathbb{N}|$ its cardinality and we defined the three different sets the i-th cell can belong to at time t:

$$A_t := \left\{ i \in \mathbb{N} : P_1^i(t) > 2 \cdot P_2^i(t) \right\} \quad (82)$$

$$B_t := \left\{ i \in \mathbb{N} : P_2^i(t) > 2 \cdot P_1^i(t) \right\} \quad (83)$$

$$C_t := \{ i \in \mathbb{N} : i \notin A_t \notin B_t \} \quad (84)$$

This way, if a cell meets the conditions of set $A_t$ $(i \in A_t)$, $B_t$ $(i \in B_t)$, or $C_t$ $(i \in C_t)$ , we consider that this cell belongs to Population 1, to Population 2 or to no population at time t, respectively. Moreover, we denoted $n_A(t)$, $n_B(t)$ and $n_C(t)$ the cardinalities of $A_t$, $B_t$and $C_t$, respectively. And finally, in order to check the efficacy of the model we defined the ratio of cells belonging to Population 1 (Equation 85), Population 2 (Equation 86) and no Population (Equation 87), at time t as:

$$r_1(t) = \frac{n_A(t)}{N} \quad (85)$$

$$r_2(t) = \frac{n_B(t)}{N} \quad (86)$$

$$r_{12}(t) = \frac{n_C(t)}{N} \quad (87)$$

Then with the simplified model defined and the nominal parameters selected (Table 2) we used MATLAB to test the system performance. For this aim, we considered a consortium composed by 100 identical cells. We performed 50 simulations where the initial state of the $P_1$ and $P_2$ variables in each cell are randomly defined with a uniform distribution in the interval [0 – 5], . The rest of the variables ($Ap_1, Ap_2, S_1^{in}, S_2^{in}, S_1^e, S_2^e$) are initially equal to zero. From these simulations we computed the mean relative error and the mean settling time (see Appendix 1).

The results obtained from these simulations are a mean relative error of 2.44% and a mean settling time of 50.2906. This demonstrates that, even if we obtain a small error, the system is always capable to self-regulate the population ratio almost perfectly. The results of one of these simulations is represented in Figure 30. In this figure we can see the evolution in time of the ratios $r_1$, $r_2$ and $r_{12}$ of the cells belonging to sets $A_t$, $B_t$ and $C_t$, respectively. This illustration shows how the consortium is capable of stabilize its own population numbers after some time.

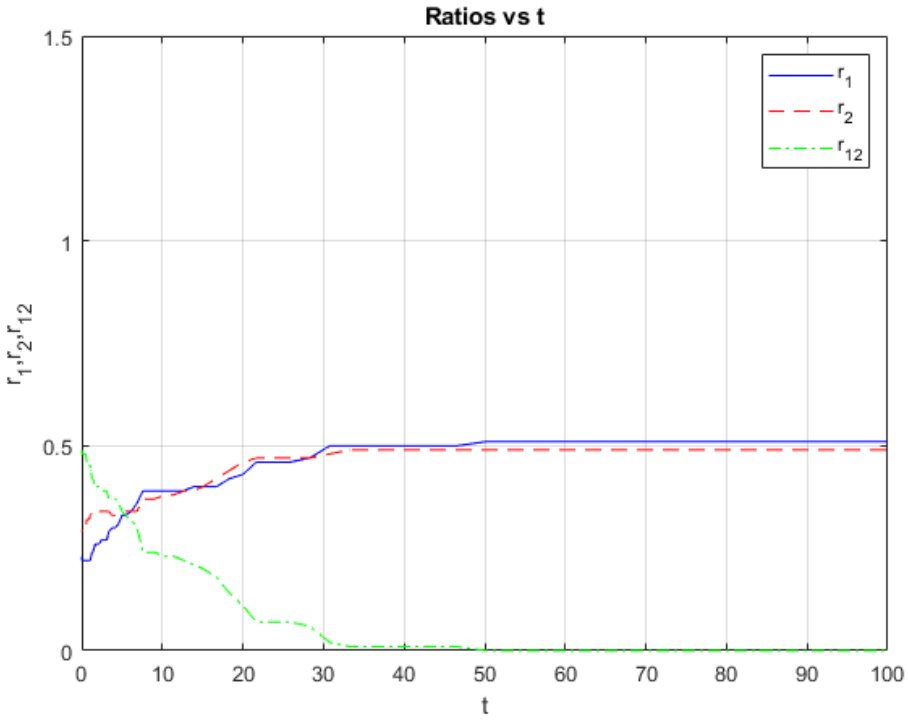


*Figure 30: Ratios evolution in time of a simulation of the simplified model. N=100 cells and 50 simulations. The evolution of the ratios $r_1$, $r_2$ and $r_{12}$ are represented in blue, red and green respectively. It can be seen how the system starts with an unbalance in the ratios ($r_1 = 0.23, r_2 = 0.29$ and $r_{12} = 0.48$) and progresively is capable of self-regulate them with a settling time of 49 t.u. and a relative error of 2% ($r_1 = 0.49, r_2 = 0.51$ and $r_{12} = 0$).*

Once the effectiveness of the model was demonstrated, we wanted to analyze its robustness against different perturbations of the nominal case. For this aim, we performed several tests that are explained in the following subsections. The simulation plan is composed by three different tests:

1. Robustness to parameters perturbation with identical cells: This test aims analyzing the role of each parameter in the model and its robustness to changes in the parameter's value. We simulated the whole system changing each time the value of a single parameter (between 20 different values) in the nominal simplified model. With each different value of the parameters we performed 50 simulations with different initial conditions and obtain the mean and standard deviation of the relative error and settling time of the evolution of the ratios.

2. Ratio tuning: The objective of this test was to find how the ratio of the two populations can be tuned to a desired value different to 0.5. Introducing an unbalance in the comparator block we created an unbalance in the final ratio. This unbalance was created by introducing an offset in the nominal value of the parameter $\tilde{\alpha}_{Ap}$. We varied this offset between 20 different values and performed 50 simulations with different initial conditions and obtained the average value of the ratios of the two populations with the different offset values.

3. Robustness to cells heterogeneity: The aim of this test was to analyze the robustness of the design to cell-to-cell variability. We changed the parameters of each cell in the consortium by drawing them from a normal distribution centered on the nominal values and varying the value of the coefficient of variation. We simulated the model 50 times with each value of coefficient of variation (20 different values) and obtained the mean and standard deviation of the relative error and settling time of the ratios at steady state.

### 3.4.1. Robustness to parameters perturbation with identical cells

First of all, we wanted to analyze the role of each parameter in the model and its robustness to changes in the parameter's value. For this aim, we simulated the whole system changing each time the value of a single parameter in the nominal simplified model. We considered a consortium of 100 identical cells. Being Pnom the nominal value of each parameter (Table 2), we varied each parameter in the interval described in the Equation 88 and sampled uniformly 20 values in such interval.

$$Parameter = \ (Pnom - 60\% \cdot Pnom, \ \ Pnom + 60\% \cdot Pnom) \qquad (88)$$

Then for each parameter variation we computed 50 simulations with random initial conditions and obtained the mean and standard deviation of the relative error and settling time of the time evolution of the ratios $r_1$ and $r_2$. In the following figures (31 – 42) the evolution of the mean relative error and mean settling time with the variation of the parameters are illustrated. The sub index "s" in the figures indicates that the varied parameters are the ones of the simplified model (see Appendix 2).

In Figure 31 and Figure 32 is represented the evolution of the mean relative error and mean settling time of the time evolution of the ratios, r1 and r2, with the evolution the parameter $\tilde{\alpha}_{Ap}$. We can see how the relative error remains practically constant, around 4%-6%, until it reaches the limiting value of 1.158 after which the error increases rapidly until it reaches the 100% error with $\tilde{\alpha}_{Ap} \geq 1.221$. Meanwhile, the settling time increases (from 18.65 t.u.) as $\tilde{\alpha}_{Ap}$ increases until it hits the limiting value 1.158, where it achieves its maximum with 92.42 t.u. We can conclude then that, by decreasing the value of $\tilde{\alpha}_{Ap,}$ we could decrease the settling time (up to 73 t.u.) of the system without a significant increase in the relative error.

Figure 33 and Figure 34 show the evolution of the mean relative error and mean settling time with the evolution of $\tilde{K}$. This time, the limiting value (2.505) is reached by decreasing the parameter. Again, until $\tilde{K}$ reaches this value, the mean relative error seems to be constant around 4%, and then increases to 100% with $\tilde{K} \leq 2.284$. The settling time increases, from 30.73 t.u., as the parameter decreases until it achieves the limiting value 2.505 where it hits the maximum settling time of 88.33 t.u. From this analysis we can depict that we could decrease the final settling time (up to 57 t.u.) by increasing the value of $\tilde{K}$ without an increase in the relative error.

In the case of the parameter $\tilde{c}_s$, whose results are illustrated in Figure 35 and Figure 36, we can see that it does not reach a limiting value after which the system stops working. The mean relative error does not experiment important variations and fluctuates between 4.8% and 2.5%. However, the mean settling time increases as the value of $\tilde{c}_s$ increases, going from a minimum of 17.96 t.u. to a maximum of 79.38 t.u. Then we can assume that by decreasing the value of $\tilde{c}_s$ we can decrease the final settling time up to 61 t.u.

In Figure 37 and Figure 38 is illustrated the evolution of the mean relative error and settling time for different values of $\tilde{\eta}$. We can see a similar behavior to that of $\tilde{K}$. The mean relative error remains quite constant (2% - 3%) until the parameter decrease enough to reach its limiting value (4.632). The settling time increases also as $\tilde{\eta}$ decreases until it hits the limiting value. Nevertheless, this time

the variation of the settling time is lower than with $\widetilde{K}$. It goes from a maximum of 85.56 t.u. to a minimum of 44.73 t.u. For this reason, we could also increase the value of $\tilde{\eta}$ in order to decrease the final settling time, but this action would not have an impact as significant as the other strategies mentioned before.

In the same way, the behavior of the mean relative error and mean settling time versus the evolution of $\tilde{\beta}_u$, represented in Figure 39 and Figure 40, is similar to the one obtained with $\tilde{\alpha}_{Ap}$. This time the limiting value is 3.474. Until $\tilde{\beta}_u$ reaches it, the mean relative error is practically constant (2.5% - 5%) and as happened with $\tilde{\alpha}_{Ap}$, the settling time increases (from 19.79 t.u. to 86.94 t.u.) as the value of $\tilde{\beta}_u$ increases until it hits the limiting value, after which the system fails. Again, we can conclude that we can decrease the settling time significantly (up to 67 t.u.) by decreasing the value of $\tilde{\beta}_u$.

Finally, Figure 41 and Figure 42 represent the evolution of the mean relative error and mean settling time with the evolution of $\tilde{\alpha}_p$. As with $\widetilde{K}$ and $\tilde{\eta}$ the limiting value (3.621) is obtained decreasing the value of the parameter. The mean relative error is also practically constant (around 3%) until it reaches the limiting value. As before the settling time increases as the value of $\tilde{\alpha}_p$decreases (from 18.3 t.u. to 82.57). With the same criteria, we can confirm that decreasing the value of $\tilde{\alpha}_p$ we can decrease the final settling time (up to 64 t.u.) without increasing the relative error.

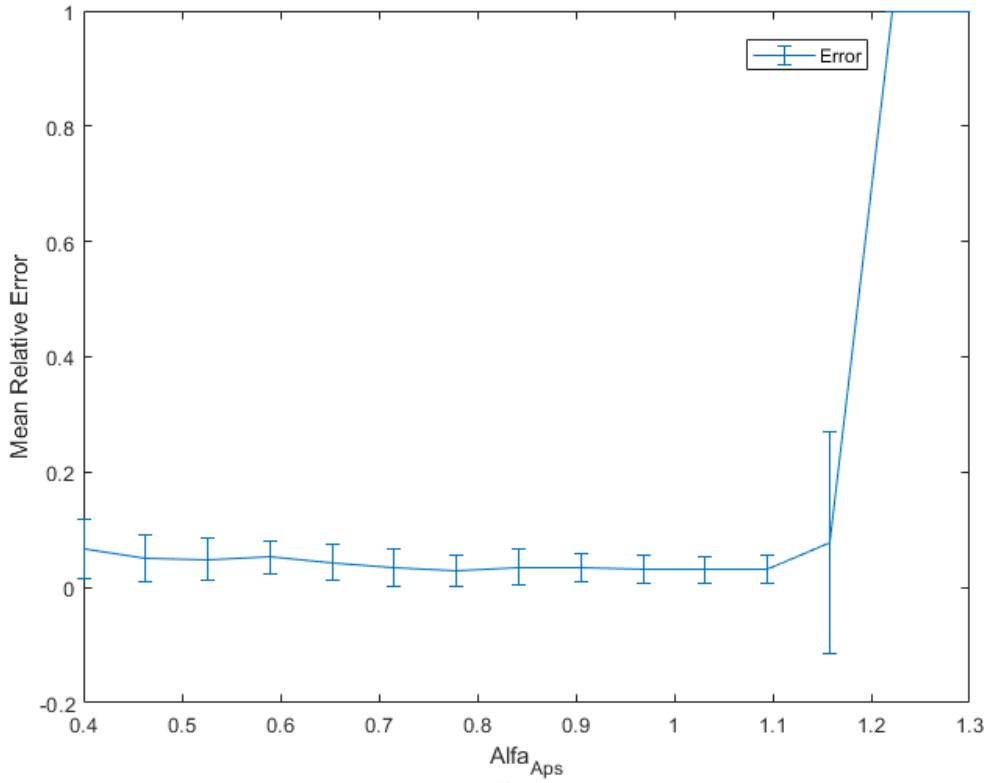


*Figure 31:Evolution of the mean relative error of the time evolution of the ratios, r1 and r2, with the evolution the parameter $\tilde{\alpha}_{Ap}$. With values of $\tilde{\alpha}_{Ap}$ higher than 1.158 the system starts to fail, reaching the 100% relative error with $\tilde{\alpha}_{Ap} \geq 1.221$.*

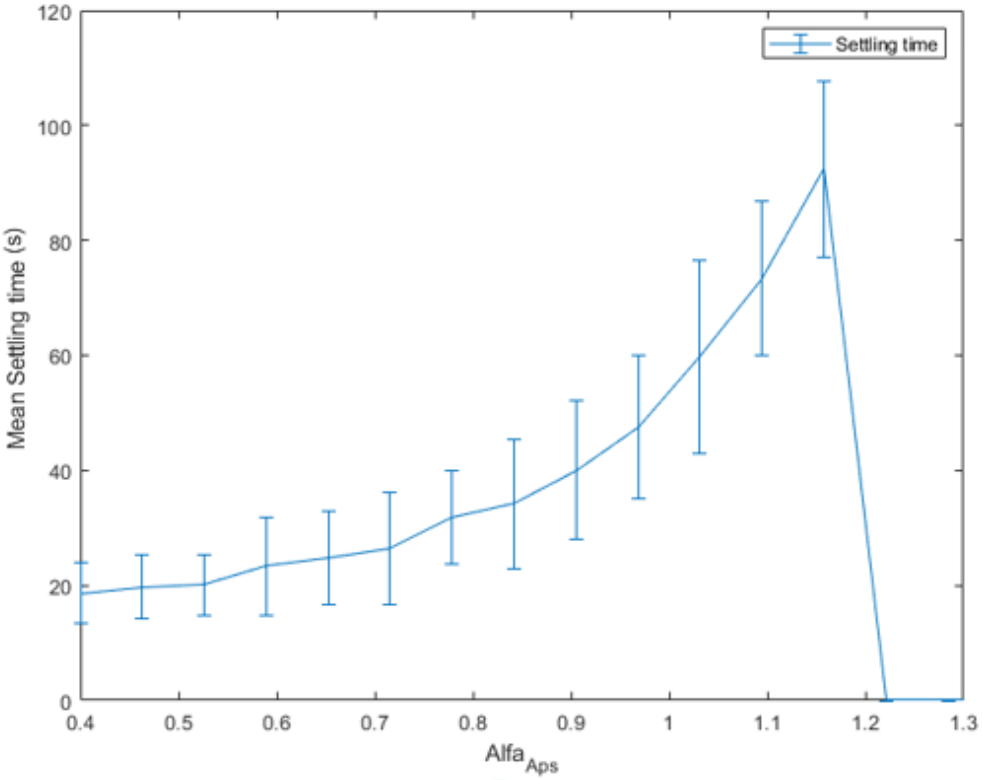


*Figure 32: Evolution of the mean settling time of the time evolution of the ratios, r1 and r2, with the evolution the parameter $\tilde{\alpha}_{Ap}$. The increase of $\tilde{\alpha}_{Ap}$ produces an increase in the settling time. The maximum settling time (92.42 t.u.) is obtained with $\tilde{\alpha}_{Ap} = 1.158$. With higher values the system fails.*

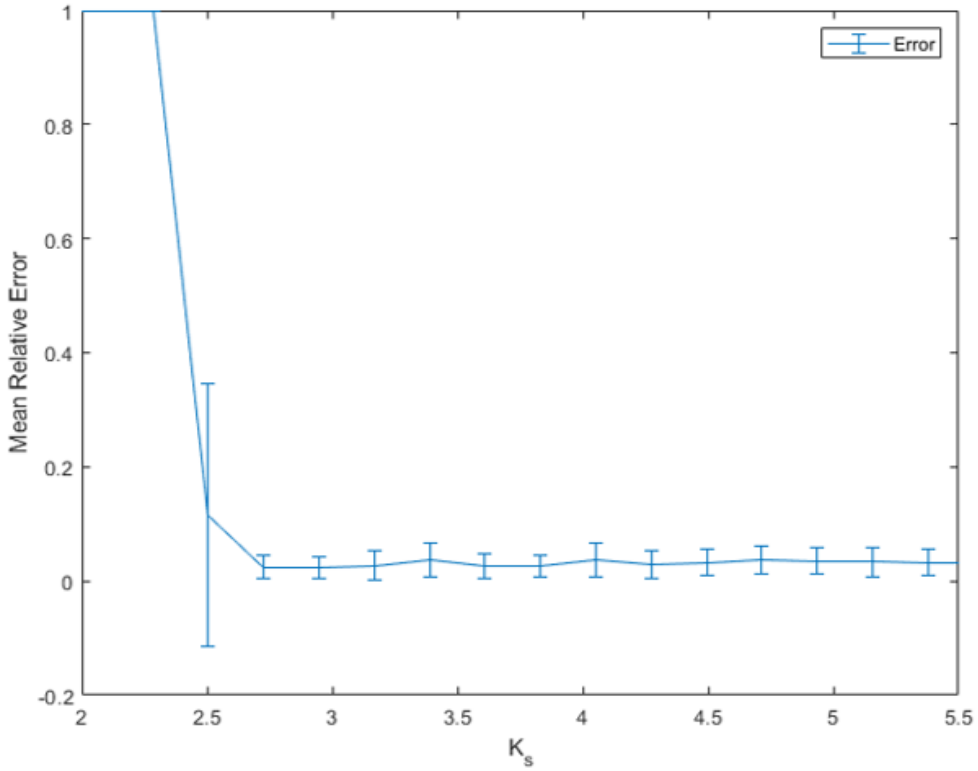


*Figure 33: Evolution of the mean relative error of the time evolution of the ratios, r1 and r2, with the evolution the parameter $\widetilde{K}$. With values of $\widetilde{K}$ lower than 2.505 the system starts to fail, reaching the 100% relative error with $\widetilde{K} \leq$ 2.284.*

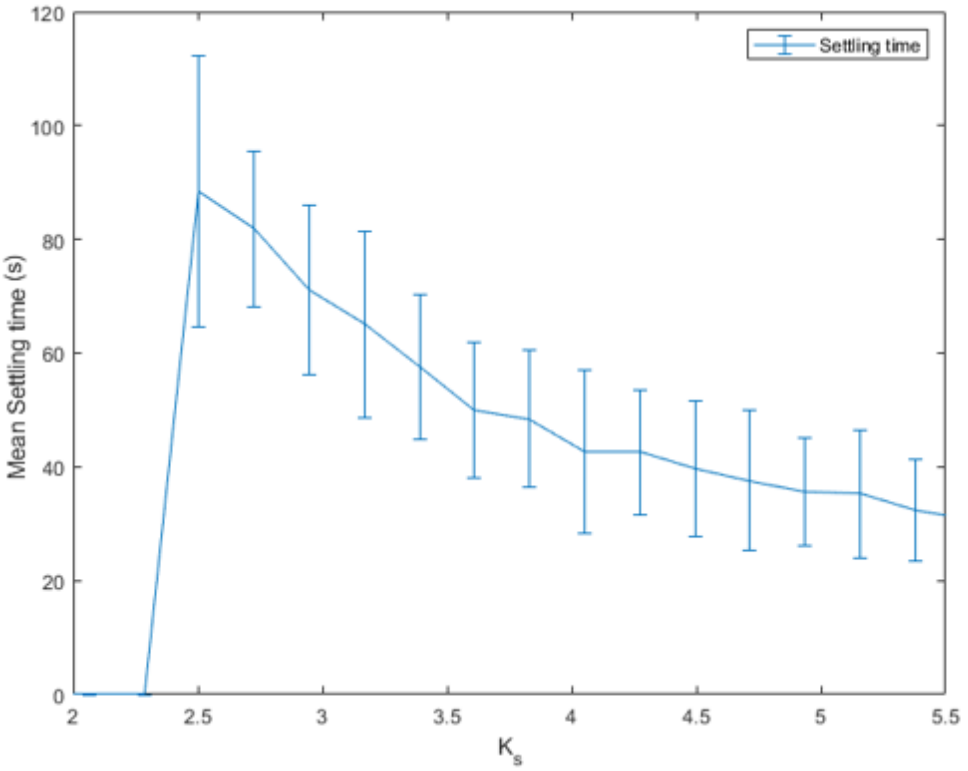


*Figure 34: Evolution of the mean settling time of the time evolution of the ratios, r1 and r2, with the evolution the parameter $\widetilde{K}$. The decrease of $\widetilde{K}$ produces an increase in the settling time. The maximum settling time (88.33 t.u.) is obtained with $\widetilde{K} = 2.505$. With lower values the system fails.*

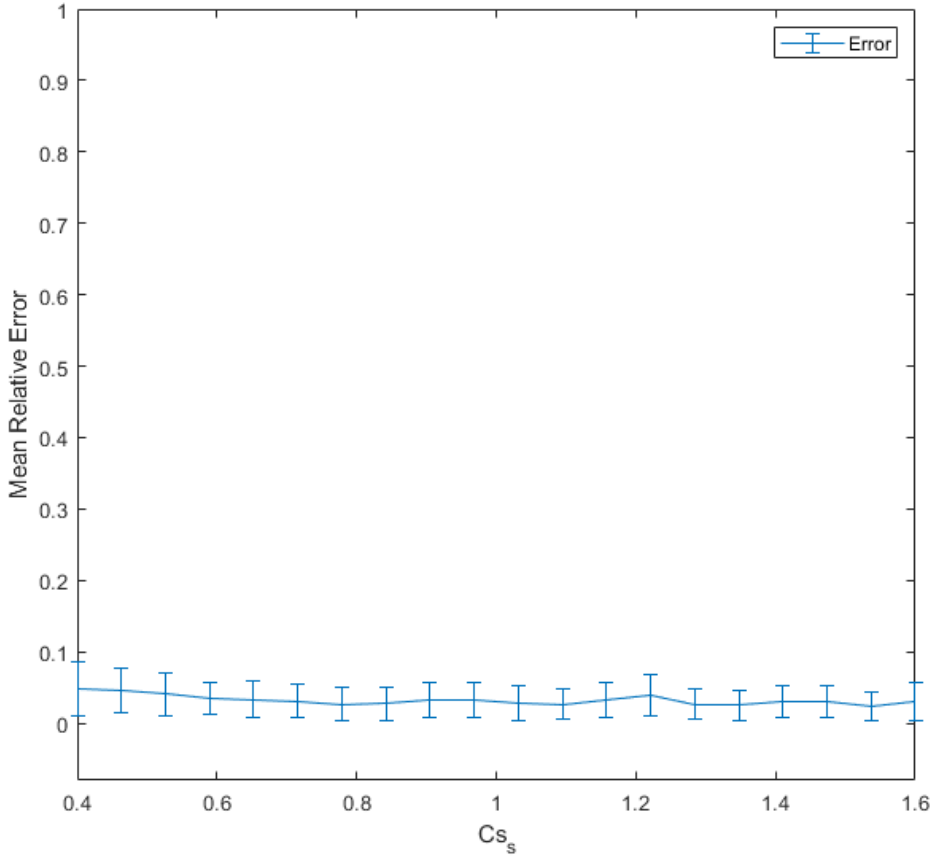


*Figure 35: Evolution of the mean relative error of the time evolution of the ratios, r1 and r2, with the evolution the parameter $\tilde{c}_s$. The changes in $\tilde{c}_s$ doesn't lead to big changes in the error. We can see an almost insignificant increase of the mean relative error with the decrease of $\tilde{c}_s$.*

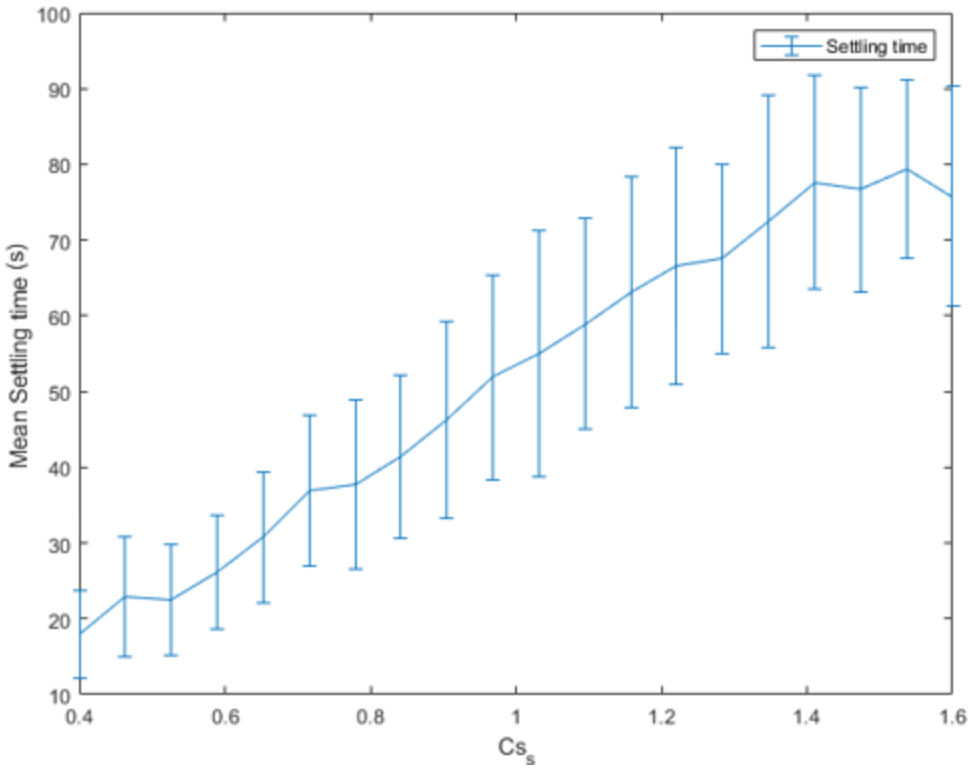


*Figure 36: Evolution of the mean settling time of the time evolution of the ratios, r1 and r2, with the evolution the parameter $\tilde{c}_s$. The increase of $\tilde{c}_s$ produces an increase in the settling time. The system doesn't reach a limiting value of $\tilde{c}_s$ which leads to failure.*

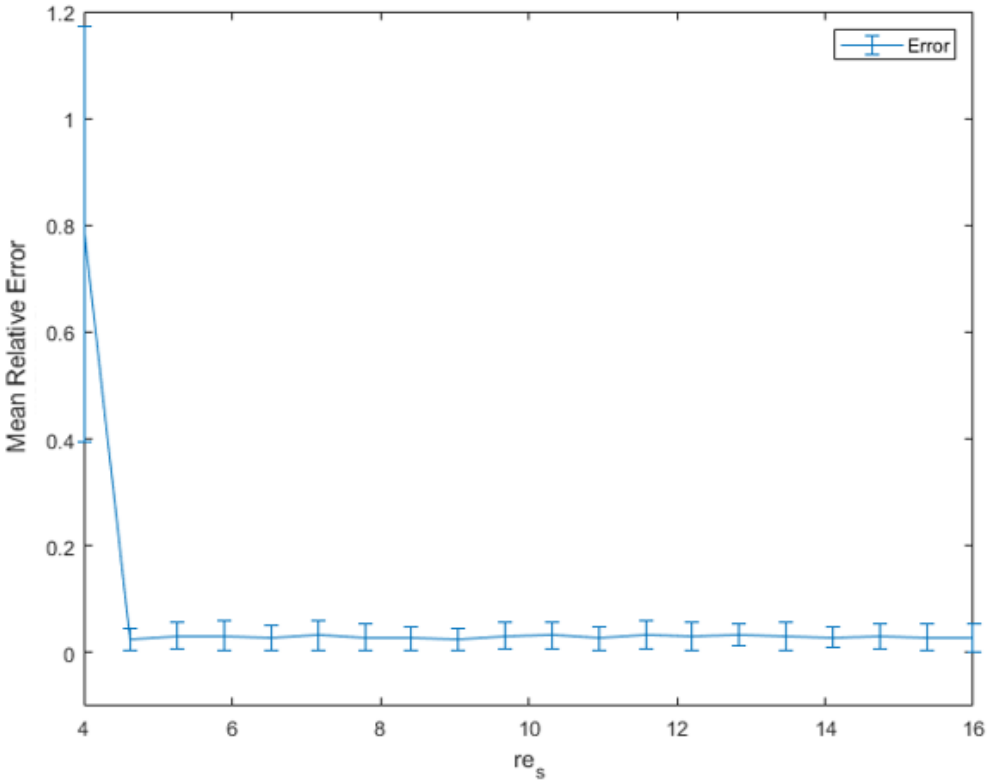


*Figure 37: Evolution of the mean relative error of the time evolution of the ratios, r1 and r2, with the evolution the parameter $\tilde{\eta}$. With values $\tilde{\eta}$ of lower than 4.632 the system starts to fail.*

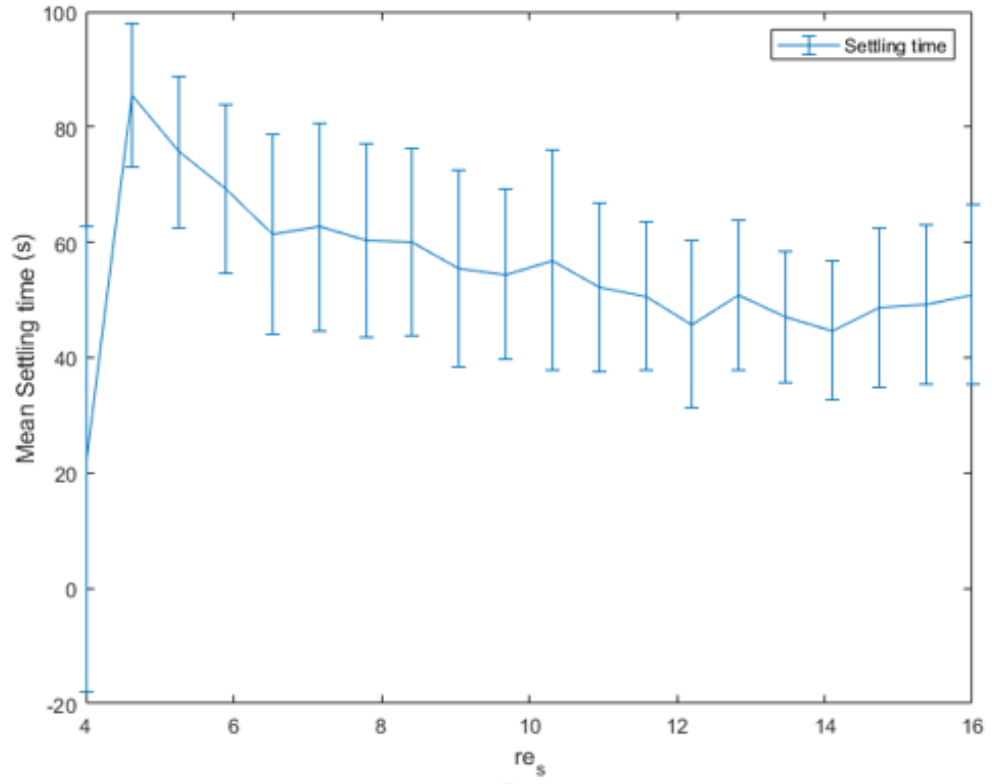


*Figure 38 Evolution of the mean settling time of the time evolution of the ratios, r1 and r2, with the evolution the parameter $\tilde{\eta}$. The decrease of $\tilde{\eta}$ produces an increase in the settling time. The maximum settling time (85.56 t.u.) is obtained with $\tilde{\eta} = 4.632$. With lower values the system fails.*

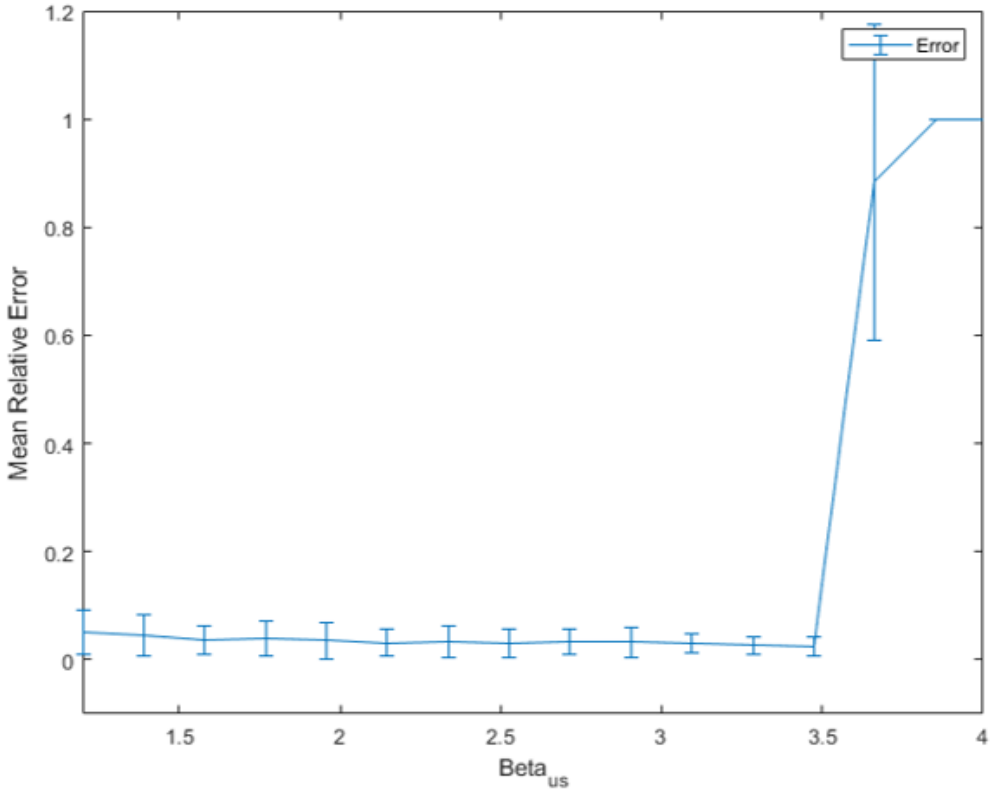


*Figure 39: Evolution of the mean relative error of the time evolution of the ratios, r1 and r2, with the evolution the parameter $\tilde{\beta}_u$. With values of $\tilde{\beta}_u$ higher than 3.474 the system starts to fail, reaching the 100% relative error with $\tilde{\beta}_u \geq 3.853$.*

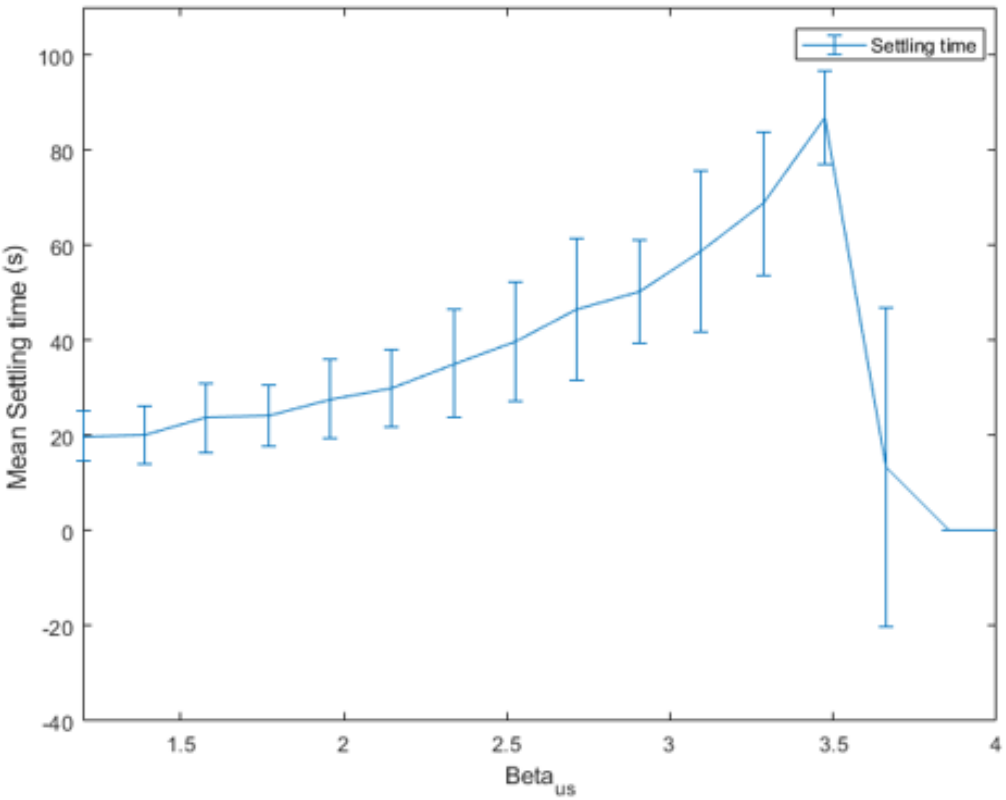


*Figure 40: Evolution of the mean settling time of the time evolution of the ratios, r1 and r2, with the evolution the parameter $\tilde{\beta}_u$. The increase of $\tilde{\beta}_u$ produces an increase in the settling time. The maximum settling time (86.94 t.u.) is obtained with $\tilde{\beta}_u = 3.474$. With higher values the system fails.*

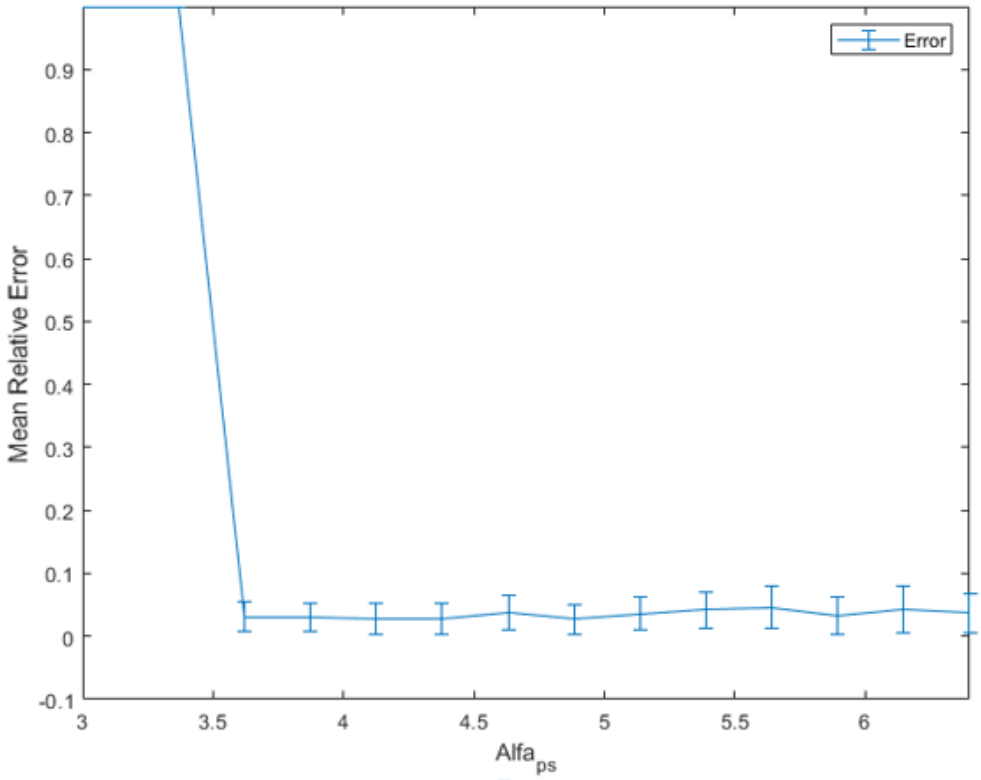


*Figure 41: Evolution of the mean relative error of the time evolution of the ratios, r1 and r2, with the evolution the parameter $\tilde{\alpha}_p$. With values of $\tilde{\alpha}_p$ lower than 3.621 the system starts to fail, reaching the 100% relative error with $\tilde{\alpha}_p \geq 3.368$.*

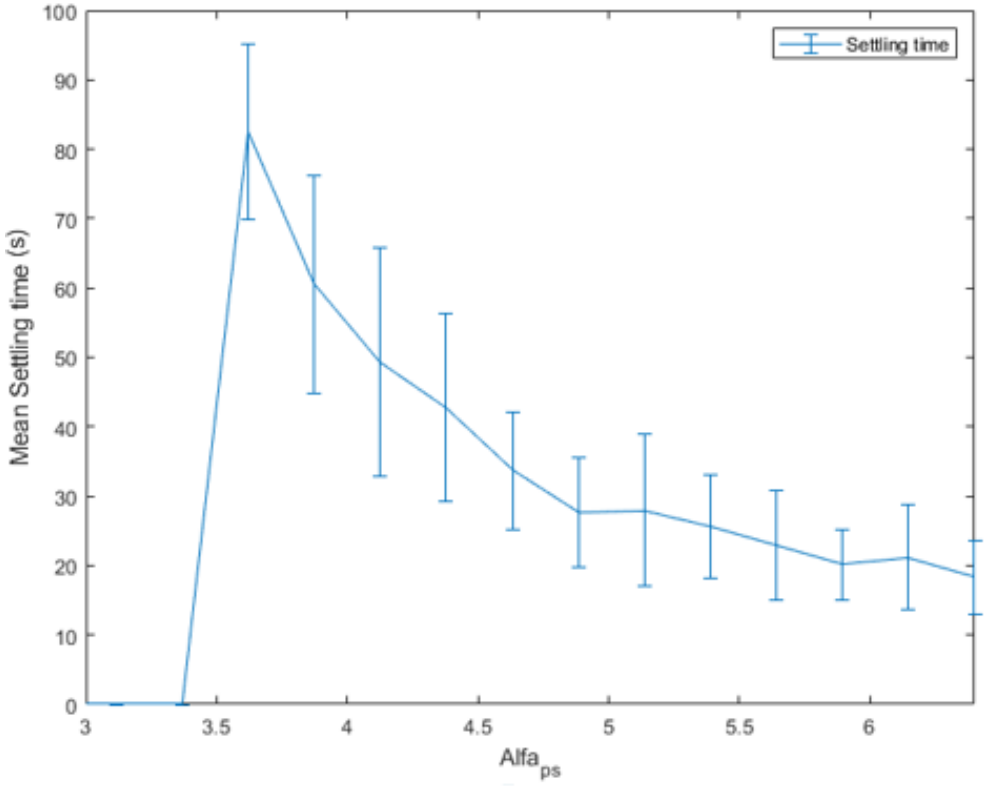


*Figure 42: Evolution of the mean settling time of the time evolution of the ratios, r1 and r2, with the evolution the parameter $\tilde{\alpha}_p$. The decrease of $\tilde{\alpha}_p$ produces an increase in the settling time. The maximum settling time (82.57 t.u.) is obtained with $\tilde{\alpha}_p = 3.621$. With higher values the system fails*

### 3.4.2. Ratio tuning

The purpose of this analysis was to find how the ratio of the two populations can be tuned to a desired value different to 0.5. For this aim, we wanted to demonstrate that introducing an unbalance in the comparator block, that is in the $Ap_i$ equations, we could obtain an unbalance in the final ratio.

This unbalance in the $Ap_i$ equations was created by giving a different value of the parameter $\tilde{\alpha}_{Ap}$ for the $Ap_1$ and $Ap_2$ equations. For this purpose, we introduced an offset in the nominal value of the parameter $\tilde{\alpha}_{Ap}$, creating this way two new parameters:

$$\tilde{\alpha}_{Ap1} = \tilde{\alpha}_{Ap} + \frac{Offset}{2} \tag{89}$$

$$\tilde{\alpha}_{Ap2} = \tilde{\alpha}_{Ap} - \frac{Offset}{2} \tag{90}$$

These two parameters were then introduced in the model having the following $Ap_i$ equations:

$$\frac{d\tilde{A}_{p1}}{dt'} = \tilde{\alpha}_{Ap1} \cdot \frac{\left(\tilde{S}_1^{in}\right)^n}{1+\left(\tilde{S}_1^{in}\right)^n} - \tilde{K} \cdot \tilde{A}_{p1} \cdot \tilde{A}_{p2} - \tilde{A}_{p2} \tag{91}$$

$$\frac{d\tilde{A}_{p2}}{dt'} = \tilde{\alpha}_{Ap2} \cdot \frac{\left(\tilde{S}_2^{in}\right)^n}{1+\left(\tilde{S}_2^{in}\right)^n} - \tilde{K} \cdot \tilde{A}_{p1} \cdot \tilde{A}_{p2} - \tilde{A}_{p2} \tag{92}$$

To analyze the impact of this offset in the final ratio, the value of the offset changed between 20 different values from -0.8 to 0.8. This way the parameters $\tilde{\alpha}_{Ap1}$ and $\tilde{\alpha}_{Ap2}$ varied from 0.6 to 1.4.

As before, we performed 50 simulations with different initial conditions for each value of the offset and computed the mean and standard deviation of the final ratio obtained from these simulations. This time, we decided to illustrate the evolution of the final ratios and the standard deviation as result of the different offset values. The results are illustrated in Figure 43. We can clearly see that there exists a correlation between the value of the offset and the final population ratios. With a 0 offset the system equalize both populations (0.5 ratios), and as the offset increases (or decreases) the difference between the two ratios also increases (decreases) linearly. Finally, with an offset of 0.8 (-0.8) the ratio relation ends up being 0:1 (1:0) (see Appendix 3).

The results obtained in this graph are especially interesting because they can definitely be used as a useful and easy tool to modulate the ratios of the populations. You just need to decide the desired final ratio of the populations and look at the graph to find the corresponding offset.

On the other hand, the standard deviation is around ±0.02. We can consider it a low standard deviation, that, for our particular case, is good as it indicates that the ratios tend to be very close to the final mean. This implies that using this graph as reference is a reliable method to define future consortia with specific population ratios.

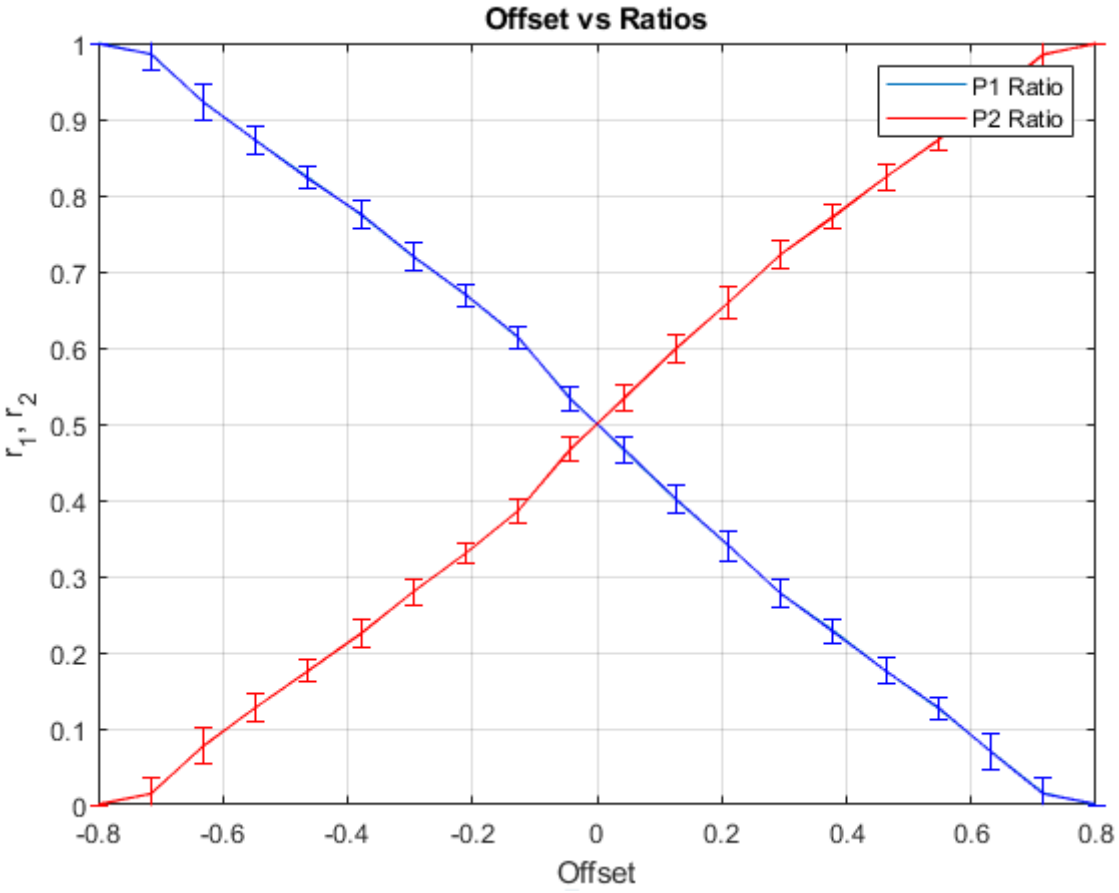


*Figure 43: Analysis of the influence of the offset in the final ratios. The red and blue lines represent the value of the ratios of Population 1 and Population 2 at steady state, respectively. The vertical lines represent the standard deviation of the final ratios. The initial ratio with 0 offset is 0.5:0.5. As the offset increases (decreases) the final value of the ratios also increases (decreases) reaching the 0:1 (1:0) ratio with an offset of +0.8 (-0.8).*

### 3.4.3. Robustness to cells heterogeneity

Finally, we wanted to analyze the robustness of the design to cell-to-cell variability, that is, to perturbations in the parameters of each cell. This is very important test because, in a real cell consortium all the cells are going to have different parameters, so this analysis shows the actual viability of the model to be implemented in the future.

For this aim, we changed the parameters of each cell in the consortium (100 cells) by drawing them from a normal distribution centered on the nominal values (Table 2), and with a standard deviation, $STD = CV * Pnom$ , being $Pnom$ and $CV$ the nominal values of the parameters and the coefficient of variation, respectively. To analyze effect of these perturbations, we simulated the model for 20 different values of CV in the interval [0 – 50%]. As before, we performed 50 simulations for each coefficient of variation and obtained the mean and standard deviation of the relative error and settling time of the ratios at steady state (see Appendix 4).

The results obtained from these simulations are illustrated in Figure 44 and Figure 45. First, we can see how the mean relative error clearly increases as the coefficient of variation increases. With a 0% coefficient of variation, that is with all the cells having the same parameters, we obtain a 3.3% relative error, this means that the final ratios accomplish almost the 0.5:0.5 ratio. However, with a coefficient of variation of 50% the final mean relative error obtained is 54%.

In the case of the settling time, we observe the opposite behavior, as the coefficient of variation increases the settling time decreases. With a 0% coefficient of variation the system shows a settling time of 52.2 t.u., while with 50% coefficient of variation it takes only 1.809 t.u. We can deduce that, even if the final relative error is increased because of the cell heterogeneity, this effect is countered by the increase in the system speed to reach the final ratios. These results make sense because the increase in the heterogeneity of the cells accelerates the toggle switch convergence. This way, allowing an extra error in the regulation we can have a faster convergence.

These results demonstrate the robustness of the model to cells heterogeneity. We performed our simulations up to 50% coefficient of variation because we wanted to see how the system would behave with high levels of heterogeneity. However, we can consider that a realistic system would have around 20% coefficient of variation. In this condition, the relative error increase of +30% while the settling time is about -50%, which is good.

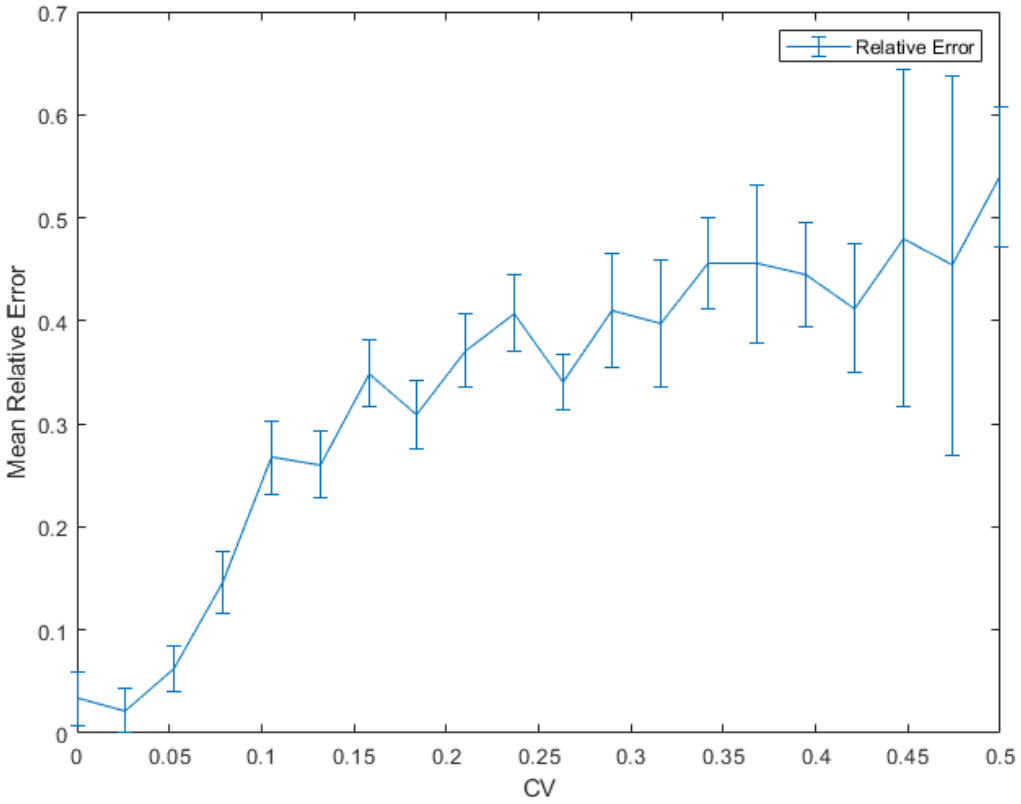


*Figure 44: Relative error of the steady state values of the ratios, r1 and r2, with respect to change of coefficient of variation of the parameters. As the coefficient of variation increases the relative error also increases, reaching a maximum of 54% with a 50% of coefficient of variation.*

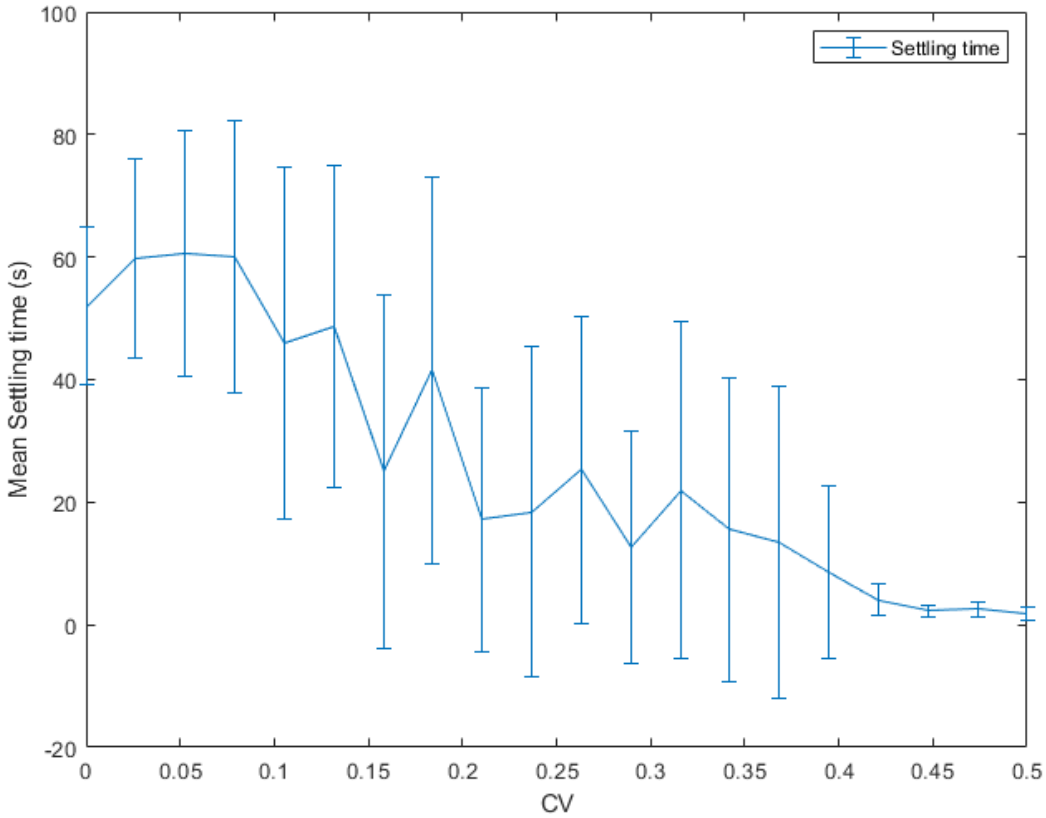


*Figure 45: Settling time of the steady-state values of the ratios, r1 and r2, with respect to change of coefficient of variation of the parameters. As the coefficient of variation increases the settling time decreases, reaching a minimum of 1.809 t.u. with a 50% of coefficient of variation.*

# 4. Conclusions

This master thesis was focused on developing new multicellular strategies for the problem of stable co-existence in microbial consortia. More precisely, we aimed to create a mathematical model for the regulation of the ratio in a two populations consortium. The main novelty of the idea was to use a bistable memory mechanism inside each cell, the toggle switch, to introduce a self-regulation capability in the cells. This makes them able to auto regulate the consortium ratios without the need of external inputs or auxiliar cells, as presented in previous works.

These goals had been successfully achieved, with the design of a mathematical model composed by three different modules; the toggle switch whose internal state defines which population each cell belongs to, the communication module consisting of two quorum sensing molecules that act as a proxy of the current population ratios of the cells in the consortium, and the comparator module in charge of identifying if there exists an unbalance in the consortium populations.

We have demonstrated the effectiveness of our model, showing how an initially unbalanced consortium, is capable of self-regulating its own relative numbers after some time. Then, we have also performed a parameter analysis, looking for the actual role of each parameter in the model. We have been able to characterize each parameter, and according to the obtained results, we have determined possible future strategies to improve the performance of the system. We have also studied the possibility of tuning the final ratio of the consortium to a desired value different to 0.5. With this analysis we found that, introducing an offset in the parameters $\tilde{\alpha}_{\mathrm{Api}}$ of the comparator module we could modify the final ratio. Finally, we have analyzed the robustness of our system to cell variability by introducing some heterogeneity in the cells of the consortium. This final test has proved the effectiveness of our system in a more realistic environment, making it viable for a future implementation.

Future research lines of this project might be, first of all, more advanced in-silico experiments to confirm the effectiveness of the design by using for example BSim, a realistic agent-based simulator of bacterial populations. Once these experiments have been performed, the implementation of the model could be considered.

# References

[1] O. Wolkenhauer and A. Muir, "The Complexity of Cell-Biological Systems," in *Philosophy of Complexity, Chaos and Non-Linearity*, Handbook of the Philosophy of Science, 17 April 2009.

[2] V. Hsiao, A. Swaminathan and R. M. Murray, "Control Theory for Synthetic Biology," *IEEE Control Systems Magazine*, vol. 18, pp. 32-62, June 2018.

[3] E. Cameron, C. J. Bashor and J. J. Collins, "A brief history of synthetic biology," *Nature Reviews Microbiology*, vol. 12, pp. 381-390, 2014.

[4] X. Ren, A.-A. Baetica, A. Swaminathan and R. M. Murray, "Population regulation in microbial consortia using dual feedback control," in *IEEE 56th Annual Conference on Decision and Control*, Melbourne, Australia, December 2017.

[5] D. Salzano, D. Fiore and M. di Bernardo, "Ratiometric control for differenciation of cell populations endowes with synthetic toggle switches," in *IEEE 58th Conference on Decision and Control* , Nice, France, December 2019.

[6] Imperial College London. 2016 iGEM - Ecolibrium project, [Online]. Available: http://2016.igem.org/Team:Imperial_College.

[7] F. K. Balagaddé, H. Song, J. Ozaki, C. H. Collins, M. Barnet, F. H. Arnold, S. R. Quake and L. You, "A synthetic Escherichia coli predator-prey ecosystem," *Molecular Systems Biology*, vol. 4, no. 1, p. 187, 2008.

[8] G. Fiore, A. Matyjaszkiewicz, F. Annunziata, C. Grierson, N. J. Savery, L. Marucci and M. di Bernardo, "In-silico analysis and implementation of multicellular feedback control strategy in a synthetic bacterial consortium," *ACS Synthetic Biology*, vol. 6, no. 3, pp. 507-517, 2016.

[9] Y. Chen, J. k. Kim, A. J. Hirning, K. Josic and M. R. Bennett, "Emergent genetic oscillations in a synthetic microbial consortium," *Science*, vol. 349, no. 6251, pp. 986-989, 2015.

[10] M. Sadeghpour, A. Veliz-Cuba, G. Orosz, K. Josic and M. R. Bennett, "Bistability and oscillations in co-repressive synthetic microbial consortia," *Quantitative biology*, vol. 5, no. 1, pp. 55-66, 2017.

[11] D. Fiore, D. Salzano, E. Cristobal-Coppulo, J. M. Olm and M. di Bernardo, "Multicellular feedback control of a genetic toggle-switch in microbial consortia," *bioRxiv*, 2020.

[12] Cosy-Bio, 2017. [Online]. Available: https://www.cosy-bio.eu/.

[13] U. Alon, An introduction to systems biology, CRC Press, 2020.

[14] T. S. Gardner, C. R. Cantor and J. J. Collins, "Construction of genetic toggle switch in Escherichia coli," *NATURE*, vol. 403, pp. 339-342, 20 January 2000.

[15] A. Guarino, D. Fiore, D. Salzano and M. di Bernardo, "Balancing cell populations endowed with a synthetic toggle switch via adaptative pulsatile feedback control," *ACS Syntethic Biology*, vol. 9, no. 4, pp. 793-803, March 2020.

[16] M. B. Miller and B. L. Bassler, "Quorum sensing in bacteria," *Annual Reviews in Microbiology*, vol. 55, no. 1 , pp. 165-199, 2001.

# Appendix 1

```
%MAIN FILE OF THE MODEL

%Tin this code we compute the nominal case of the model.
%The objective of this code is to verify the efficacy of the model.
%We define 3 different sets each cell can belong to at a time t, being N
%the finite set of cells in the consortium:

    %A_t:={ieN: P_1^i(t)>2*P_2^i(t)}     Cell belongs to Population 1
    %B_t:={ieN: P_2^i(t)>2*P_1^i(t)}     Cell belongs to Population 2
    %C_t:={ieN: ie/A_te/B_t}             Cell belons to No Population

%Then the ratio of cells belonging to Population 1, Population 2 and No
%Population are defined, being n_A (t), n_B (t) and n_C (t) the
cardinalities of A_t, B_t and C_t:

    %r_1 (t)=  (n_A (t))/N
    %r_2 (t)=  (n_B (t))/N
    %r_3 (t)=  (n_C (t))/N

%Then we illustrate the ratios evolution and verify if they tend to 0.5:0.5
%We also compute the mean relative error and mean settling time of 50
simulations with different initial conditions

clc
close all
clear

% Order of the variables inside the cell:
% x(1)=P1, % x(2)=P2, x(3)=Ap1, x(4)=Ap2, x(5)=s_in1, x(6)=s_in2,

%Definition of the number of cells
N = 100;

%Definition of the simulation time
tf = 100;

%Definition of the Initial conditions inside the cell
x0 = [10,0,0,0,0,0]';   %p10=10, p20=0, Ap10=0, Ap20=0, s1_in0=0, s2_in0=0
n = length(x0);         %Number of variables inside the cell (6)
qn = 50; %Number of simulations with different initial conditions

for q = 1 : qn
disp("Evaluating cycle q = " + num2str(q))

%Creation of a vector of length N*n with all the initial conditions for the
internal variables for the N number of cells
    X0 = [];
    for k = 1 : N
        X0( (k-1)*n+1 : k*n ) = [ rand(2,1) * 5 ; x0(3:n)]; % [p1,p2] are
randomly drawn with uniform distribution in [0;5]x[0;5]
    end
```

```
%Definition of the initial conditions of the external variables (quorum
sensing molecules)
    y0 = [0, 0];   %s1_e0=0, s2_e0=0

%Creation of a vector of length N*n+2 with all the initial conditions for the
internal variables for the N number of cells and the two external variables
    Z0 = [X0, y0];

%Resolution of the whole system equations:
    [t,Z] = ode15s(@whole_sys_Main,[0,tf],Z0,[],[n,N]); % output: stack
matrix of all the systems

% X is a matrix having n*N columns and length(t)rows
% Y is a matrix having 2 columns and length(t)rows
% Z is a matrix having (n*N + 4) columns and length(t)rows

%Ratios computation
    P1 = Z(:, 1:n:N*n);  %Creating matrix with all the P1 of each cell (txN)
    P2 = Z(:, 2:n:N*n);  %Creating matrix with all the P2 of each cell (txN)

%Creation of two sets A (Population 1) and B (Population 2)
    A = P1 > (2*P2);
    B = P2 > (2*P1);
    C = (P1 < 2*P2) & (P2 < 2*P1);
    r1 = sum(A,2)/N; %Vector of ratios of population 1 through time
    r2 = sum(B,2)/N; %Vector of ratios of population 2 through time
    r12 = sum(C,2)/N; %Vector of ratios of not_pop1 or not_pop2, so that
r1+r2+r12=1

% r is a matrix having 3 columns and length(t)rows
    r=[r1, r2, r12];

%Relative error
    error = norm( r1(end) - 0.5 )/0.5;
    Error(q) =  error;% Vector with the relative errors

%Settling time
    Y = r1(end);
    e = Y * 0.01; %Error
    st = 0;   %Initializing settling time
    for T = 1 : length(t)
          if (r1(T) < (Y + e)) && (r1(T) > (Y - e))
               st = t(T);
               break;
          end
    end
    SettleT(q) =  st;

figure(q)
plot(t,r1,'b-',t,r2,'r--',t,r12,'g-.')
grid on
axis([0 inf 0 1.5])
xlabel('t')
ylabel('r_1,r_2,r_1_2');
title('Ratios vs t')
legend('r_1','r_2','r_1_2')
end
MeanError = mean(Error);
MeanSettleT= mean(SettleT);
```

```
%FUNCTION OF THE MAIN CODE

%This code is computed for each simulation and contains the simplified
equations of the model for the nominal case

function [dZdt] = whole_sys_Main(t,Z,p)

%Definition of the number of variables inside the cells
n = p(1);

%Definition of the number of cells
N = p(2);

%Creation of a vector with s1_e, s2_e from Z
y = Z(end-1:end);     %y=[s1_e;s2_e]

%Creation of a vector N*n with all the internal variables of all cells at
%time t
FZ = [];

for k = 1:N
    x = Z( (k - 1)*n+1 : k*n );  % x: sub vector of X with length n
(variables of each cell)
    FZ = [FZ; single_cell(x,y)];
end

v = Z(5:n:end); %Creating vector with all the s_in1 of each cell (length=N)
u = Z(6:n:end); %Creating vector with all the s_in2 of each cell (length=N)

% FY: Dynamics of global (external) concentration on quorum sensing molecules
that link together all the cell.
FY =  environ(v,u,y);

%%Creation of a vector of length N*n+2+2 with all the conditions of the
internal variables for the N number of cells and the two external variables
at t

dZdt = [FZ; FY];

end

function [dydt] = environ(v,u,y)

%%% s1_e and s2_e parameters
re_s = 10;                  % re/d
Gamma_e_s = 1.5;            %Gamma_e/d

%Variables definition
Se1_s = y(1);   % Se1_s = s1_e/Theta_s
Se2_s = y(2);   % Se2_s = s2_e/Theta_s

dydt =  [re_s * sum(v - Se1_s) - Gamma_e_s * Se1_s;
        re_s * sum(u - Se2_s) - Gamma_e_s * Se2_s];

end
```

```
%Creation of a vector with the computed internal variables at t for each
%cell
function [dxdt] = single_cell(x,y)

%SIMPLIFICATION PARAMETERS
n = 2;              %Hill coef
Alfa_Ap_s = 1;      %Alfa_Ap/(Theta_u*d)
K_s = 3.5;          %(k*Theta_u)/d
cs_s = 1;           %(cs*Theta_p)/(Theta_s*d)
re_s = 10;          %re/d
Beta_u_s = 3;       %Beta_u/(Theta_p*d)
Alfa_p_s = 4;       %Alfa_p/(Theta_p*d)

%Variables definition
P1_s = x(1);     % P1_s = p1/Theta_p
P2_s = x(2);     % P2_s = p2/Theta_p
Ap1_s = x(3);    % Ap1_s = Ap1/Theta_u
Ap2_s = x(4);    % Ap2_s = Ap2/Theta_u
Sin1_s = x(5);   % Sin1_s = s_in1/Theta_s
Sin2_s = x(6);   % Sin2_s = s_in2/Theta_s
Se1_s = y(1);    % Se1_s = s1_e/Theta_s
Se2_s = y(2);    % Se2_s = s2_e/Theta_s

%Dynamics of internal variables inside each cell.
%SIMPLIFIED MODEL:
dxdt = [
    ( Alfa_p_s/(1 + P2_s^n) ) - P1_s + Beta_u_s * Ap2_s^n/( 1 + (Ap2_s^n) );
    ( Alfa_p_s/(1 + P1_s^n) ) - P2_s + Beta_u_s * Ap1_s^n/( 1 + (Ap1_s^n) );
    Alfa_Ap_s * Sin1_s^n/(1 + Sin1_s^n) - K_s * Ap1_s * Ap2_s - Ap1_s;
    Alfa_Ap_s * Sin2_s^n/(1 + Sin2_s^n) - K_s * Ap1_s * Ap2_s - Ap2_s;
    cs_s * P1_s - Sin1_s + re_s * (Se1_s - Sin1_s);
    cs_s * P2_s - Sin2_s + re_s * (Se2_s - Sin2_s)
    ];

end
```

# Appendix 2

```
%FIRST TEST: ROBUSTNESS TO PARAMETERS HETEROGENEITY WITH IDENTICAL CELLS

%The objective is to analyze the roll of each parameter in the system and
%the robustness of the system to parameters variations in the nominal
%parameters.

%For each parameter we varied their values between 20 different amounts in
%the interval described as:
    %Parameter= (Pnom-0.6*Pnom,   Pnom+0.6*Pnom)

%Then we illustrate the mean relative error and settling time vs the
%evolution of each parameter.

clc
close all
clear

% Order of the variables inside the cell:
% x(1)=P1, % x(2)=P2, x(3)=Ap1, x(4)=Ap2, x(5)=s_in1, x(6)=s_in2,

%Definition of the number of cells
N = 100;

%Definition of the Initial conditions inside the cell
x0 = [10,0,0,0,0,0]';   %p10=10, p20=0, Ap10=0, Ap20=0, s1_in0=0, s2_in0=0
n = length(x0);       %Number of variables inside the cell (6)

%Definition of the initial conditions of the external variables (quorum
sensing molecules)
y0 = [0, 0];   %s1_e0=0, s2_e0=0

%Definition of the simulation time
tf = 100;

kn = 20; %Number of different values for each parameter

qn = 50; %Number of trials with different values for a parameter

%Initialization of vectors
Error = zeros(1, qn);
SettleT = zeros(1, qn);

MeanError = zeros(1, kn);
MeanSettleT = zeros(1, kn);

StdError = zeros(1, kn);
StdSettleT = zeros(1, kn);

%Vector with the nominal parameters
P_nom = [1, 3.5, 1, 10, 3, 4] ; %[Alfa_Ap_s, K_s, cs_s, re_s, Beta_u_s,
Alfa_p_s]
```

```
%Matrix with the parameters of each cell wit variations form the nominal
%case

percent_var = 60/100;
Param = [];

P_names = ["Alfa_A_p_s", "K_s", "Cs_s", "re_s", "Beta_u_s", "Alfa_p_s"];

for k = 1 : n
    Param = [Param; linspace( (1 - percent_var) * P_nom(k), (1 + percent_var)
* P_nom(k), kn)];
end

for i = 1 : n
    disp("Evaluating cycle i = " + num2str(i))
    p_i = Param(i, :);

    for j = 1 : kn
        var = P_nom;
        var(i) = p_i(j);

        tic
        for q = 1 : qn

            %Creation of a vector of length N*n with all the initial
            conditions for the internal
            %variables for the N number of cells
            X0 = [];

            for m = 1 : N
                      X0( (m-1)*n+1 : m*n ) = [ rand(2,1) * 5 ; x0(3:n)];
                  % [p1,p2] are randomly drawn with uniform distribution in
                  [0;5]x[0;5]
            end

            %Creation of a vector of length N*n+2 with all the initial
            conditions for the internal variables for the N number of cells
            and the two external variables
            Z0 = [X0, y0];

            %Resolution of the whole system equations:
            [t,Z] = ode15s(@whole_sys_NumVal1,[0,tf],Z0,[],[n,N,var]);

            % output: stack matrix of all the systems

            %For the different parameters instead of Alfa_Ap_s: K_s,cs_s...

            % X is a matrix having n*N columns and length(t)rows
            % Y is a matrix having 2 columns and length(t)rows
            % Z is a matrix having (n*N + 4) columns and length(t)rows

            %Ratios computation
            P1 = Z(:, 1:n:N*n);  %Creating matrix with all the P1 of each
            cell (txN)
            P2 = Z(:, 2:n:N*n);  %Creating matrix with all the P2 of each
            cell (txN)
```

```
            %Creation of two sets A (Population 1) and B (Population 2)
            A = P1 > (2*P2);
            B = P2 > (2*P1);
            C = (P1 < 2*P2) & (P2 < 2*P1);

            r1 = sum(A,2)/N; %Vector of ratios of population 1 through time
            r2 = sum(B,2)/N; %Vector of ratios of population 2 through time
            r12 = sum(C,2)/N; %Vector of ratios of not_pop1 or not_pop2, so
            that r1+r2+r12=1

            % r is a matrix having 3 columns and length(t)rows
            r=[r1, r2, r12];

            %Relative error
            error = norm( r1(end) - 0.5 )/0.5;
            Error(q) =  error;% Vector with the relative errors

            %Settling time
            Y = r1(end);
            e = Y * 0.01; %Error
            st = 0;   %Initializing settling time

            for T = 1 : length(t)
                if (r1(T) < (Y + e)) && (r1(T) > (Y - e))
                    st = t(T);
                    break;
                end
            end
            SettleT(q) =  st;
        end
        toc

        MeanError(i,j) = mean(Error);
        MeanSettleT(i,j) = mean(SettleT);

        StdError(i,j) = std(Error);
        StdSettleT(i,j) = std(SettleT);
    end

    figure(i)
    plot(Param(i,:),MeanError(i,:))
    grid on
    axis([0 inf 0 1.1])
    errorbar(Param(i,:),MeanError(i,:),StdError(i,:))
    xlabel(P_names(i))
    ylabel('Mean Relative Error');
    legend('Error')


    figure(i+6)
    plot(Param(i,:),MeanSettleT(i,:))
    grid on
    axis([0 inf 0 inf])
    errorbar(Param(i,:),MeanSettleT(i,:),StdSettleT(i,:))
    xlabel(P_names(i))
    ylabel('Mean Settling time (s) ');
    legend('Settling time')
end
```

```
%FUNCTION FOR THE FIRST TEST

%The value of the parameters is introduced in each simulation

function [dZdt] = whole_sys_NumVal1(t,Z,p)    %Z=Z0'

%Definition of the number of variables inside the cells
n = p(1);

%Definition of the number of cells
N = p(2);

%Order of variables = [Alfa_Ap_s, K_s, cs_s, re_s, Beta_u_s, Alfa_p_s]

%Creation of a vector with s1_e, s2_e from Z
y = Z(end-1:end);      %y=[s1_e;s2_e]

%Creation of a vector N*n with all the internal variables of all cells at
%time t
FZ = [];

for k = 1:N
    x = Z( (k - 1)*n+1 : k*n );  % x: sub vector of X with length n
    (variables of each cell)
    FZ = [FZ; single_cell(x,y,p)];
end

v = Z(5:n:end); %Creating vector with all the s_in1 of each cell (length=N)
u = Z(6:n:end); %Creating vector with all the s_in2 of each cell (length=N)

% FY: Dynamics of global (external) concentration on quorum sensing molecules
that link together all the cell.
FY =  environ(v,u,y);

%%Creation of a vector of length N*n+2+2 with all the conditions of the
internal variables for the N number of cells and the two external variables
at t

dZdt = [FZ; FY];

end

function [dydt] = environ(v,u,y)

%%% s1_e and s2_e parameters
re_s = 10;                  % re/d
Gamma_e_s = 1.5;            %Gamma_e/d

%Variables definition
Se1_s = y(1);    % Se1_s = s1_e/Theta_s
Se2_s = y(2);    % Se2_s = s2_e/Theta_s

dydt =  [re_s * sum(v - Se1_s) - Gamma_e_s * Se1_s;
        re_s * sum(u - Se2_s) - Gamma_e_s * Se2_s];

end
```

```
%Creation of a vector with the computed internal variables at t for each
%cell
function [dxdt] = single_cell(x,y,p)

%SIMPLIFICATION PARAMETERS
n = 2;                  %Hill coef
Alfa_Ap_s = p(3);       %Alfa_Ap/(Theta_u*d)
K_s = p(4);             %(k*Theta_u)/d
cs_s = p(5);            %(cs*Theta_p)/(Theta_s*d)
re_s = p(6);            %re/d
Beta_u_s = p(7);        %Beta_u/(Theta_p*d)
Alfa_p_s = p(8);        %Alfa_p/(Theta_p*d)

%Variables definition
P1_s = x(1);     % P1_s = p1/Theta_p
P2_s = x(2);     % P2_s = p2/Theta_p
Ap1_s = x(3);    % Ap1_s = Ap1/Theta_u
Ap2_s = x(4);    % Ap2_s = Ap2/Theta_u
Sin1_s = x(5);   % Sin1_s = s_in1/Theta_s
Sin2_s = x(6);   % Sin2_s = s_in2/Theta_s
Se1_s = y(1);    % Se1_s = s1_e/Theta_s
Se2_s = y(2);    % Se2_s = s2_e/Theta_s

%Dynamics of internal variables inside each cell.
%SIMPLIFIED MODEL:
dxdt = [
    ( Alfa_p_s/(1 + P2_s^n) ) - P1_s + Beta_u_s * Ap2_s^n/( 1 + (Ap2_s^n) );
    ( Alfa_p_s/(1 + P1_s^n) ) - P2_s + Beta_u_s * Ap1_s^n/( 1 + (Ap1_s^n) );
    Alfa_Ap_s * Sin1_s^n/(1 + Sin1_s^n) - K_s * Ap1_s * Ap2_s - Ap1_s;
    Alfa_Ap_s * Sin2_s^n/(1 + Sin2_s^n) - K_s * Ap1_s * Ap2_s - Ap2_s;
    cs_s * P1_s - Sin1_s + re_s * (Se1_s - Sin1_s);
    cs_s * P2_s - Sin2_s + re_s * (Se2_s - Sin2_s)
    ];

end
```

# Appendix 3

```
%SECOND TEST: RATIO TUNING

%The objective is to find the way to tune the ratio of the two population
%to a desired value different to 0.5
%Introducing an unbalance in the comparator block, that is in the Api
%equations, we obtain an unbalance in the final ratio
%This unbalance is created by giving a different value of the parameter
alfa_ap_s for equations Ap1 and Ap2
%We introduce an offset in the nominal value of the parameter alfa_ap_s
%Then we illustrate the evolution of the ratios with the different values of
offset

clc
close all
clear

% Order of the variables inside the cell:
% x(1)=P1, % x(2)=P2, x(3)=Ap1, x(4)=Ap2, x(5)=s_in1, x(6)=s_in2,

%Definition of the number of cells
N = 100;

%Definition of the Initial conditions inside the cell
x0 = [10,0,0,0,0,0]';   %p10=10, p20=0, Ap10=0, Ap20=0, s1_in0=0, s2_in0=0
n = length(x0);        %Number of variables inside the cell (6)

%Definition of the initial conditions of the external variables (quorum
sensing molecules)
y0 = [0, 0];   %s1_e0=0, s2_e0=0

%Definition of the simulation time
tf = 100;

kn = 20; %Number of different Offset values

qn = 50; %Number of trials computed (with different initial conditions) for
each value of the offset

%Initialization of vectors
FinalR1 = zeros(1, qn);
FinalR2 = zeros(1, qn);
FinalR12 = zeros(1, qn);

MeanFinalR1 = zeros(1, kn);
MeanFinalR2 = zeros(1, kn);
MeanFinalR12 = zeros(1, kn);
Offset_Val = linspace(-0.8,0.8,kn);

for k = 1 : kn
    disp("Evaluating cycle k = " + num2str(k))

    offset = Offset_Val(k);

    for q = 1 : qn
```

```
    %Creation of a vector of length N*n with all the initial conditions
    for the internal variables for the N number of cells
    X0 = [];

    for m = 1 : N

            X0( (m-1)*n+1 : m*n ) = [ rand(2,1) * 5 ; x0(3:n)];
            % [p1,p2] are randomly drawn with uniform distribution in
            [0;5]x[0;5]
    end

    %Creation of a vector of length N*n+2 with all the initial conditions
    for the internal variables for the N number of cells and the two
    external variables
    Z0 = [X0, y0];

    %Resolution of the whole system equations:
    [t,Z] = ode15s(@whole_sys_NumVal2,[0,tf],Z0,[],[n,N,offset]);

    % output: stack matrix of all the systems
    %For the different parameters instead of Alfa_Ap_s: K_s,cs_s...
    % X is a matrix having n*N columns and length(t)rows
    % Y is a matrix having 2 columns and length(t)rows
    % Z is a matrix having (n*N + 4) columns and length(t)rows

    %Ratios computation
    P1 = Z(:, 1:n:N*n);
    %Creating matrix with all the P1 of each cell (txN)
    P2 = Z(:, 2:n:N*n);
    %Creating matrix with all the P2 of each cell (txN)

    %Creation of two sets A (Population 1) and B (Population 2)
    A = P1 > (2*P2);
    B = P2 > (2*P1);
    C = (P1 < 2*P2) & (P2 < 2*P1);

    r1 = sum(A,2)/N; %Vector of ratios of population 1 through time
    r2 = sum(B,2)/N; %Vector of ratios of population 2 through time
    r12 = sum(C,2)/N; %Vector of ratios of not_pop1 or not_pop2, so that
    r1+r2+r12=1

    % r is a matrix having 3 columns and length(t)rows
    r=[r1, r2, r12];

    %Final ratios
    FinalR1(q) = r1(end);
    FinalR2(q) = r2(end);
    FinalR12(q) = r12(end);

end

MeanFinalR1(k) = mean(FinalR1);
MeanFinalR2(k) = mean(FinalR2);

StdFinalR1(k) = std(FinalR1);
StdFinalR2(k) = std(FinalR2);

end
```

```
figure(1)
plot(Offset_Val, MeanFinalR1)
hold on
plot(Offset_Val, MeanFinalR2,'r')
grid on
axis([-inf inf 0 1])
xlabel('Offset')
ylabel('r_1, r_2');
title('Offset vs Ratios')
errorbar(Offset_Val, MeanFinalR1,StdFinalR1,'b')
errorbar(Offset_Val, MeanFinalR2,StdFinalR2,'r')
legend('P1 Ratio','P2 Ratio')
```

```
%FUNCTION FOR THE SECOND TEST

%All the values of the parameters are the nominal ones except for alfa_ap_s

%Two new parameters are created alfa_ap_s_1 and alfa_ap_s_2:
    %Alfa_Ap_s_1 = Alfa_Ap_s + offset/2;
    %Alfa_Ap_s_2 = Alfa_Ap_s - offset/2;

%In each simulation a new value of the offset is introduced

function [dZdt] = whole_sys_NumVal2(t,Z,p)    %Z=Z0'

n = p(1);
N = p(2);

%Creation of a vector with s1_e, s2_e from Z
y = Z(end-1:end);      %y=[s1_e;s2_e]

%Creation of a vector N*n with all the internal variables of all cells at
%time t
FZ = [];

for k = 1:N

    x = Z( (k - 1)*n+1 : k*n );  % x: sub vector of X with length n
    (variables of each cell)
    FZ = [FZ; single_cell(x,y,p)];
end

v = Z(5:n:end); %Creating vector with all the s_in1 of each cell (length=N)
u = Z(6:n:end); %Creating vector with all the s_in2 of each cell (length=N)

% FY: Dynamics of global (external) concentration on quorum sensing molecules
that link together all the cell.
FY =  environ(v,u,y);

%%Creation of a vector of length N*n+2+2 with all the conditions of the
internal variables for the N number of cells and the two external variables
at t

dZdt = [FZ; FY];

end

function [dydt] = environ(v,u,y)

%%% s1_e and s2_e parameters
re_s = 10;                  % re/d
Gamma_e_s = 1.5;            %Gamma_e/d

%Variables definition
Se1_s = y(1);    % Se1_s = s1_e/Theta_s
Se2_s = y(2);    % Se2_s = s2_e/Theta_s

dydt =  [re_s * sum(v - Se1_s) - Gamma_e_s * Se1_s;
        re_s * sum(u - Se2_s) - Gamma_e_s * Se2_s];
end
```

```
%Creation of a vector with the computed internal variables at t for each
%cell
function [dxdt] = single_cell(x,y,p) %p añadida

%SIMPLIFICATION PARAMETERS
n = 2;              %Hill coef

%Creation of the new parameters with the offset
Alfa_Ap_s = 1;
offset = p(3);
Alfa_Ap_s_1 = Alfa_Ap_s + offset/2;
Alfa_Ap_s_2 = Alfa_Ap_s - offset/2;

K_s = 3.5;          %(k*Theta_u)/d
cs_s = 1;           %(cs*Theta_p)/(Theta_s*d)
re_s = 10;          %re/d
Beta_u_s = 3;       %Beta_u/(Theta_p*d)
Alfa_p_s = 4;       %Alfa_p/(Theta_p*d)


%Variables definition
P1_s = x(1);    % P1_s = p1/Theta_p
P2_s = x(2);    % P2_s = p2/Theta_p
Ap1_s = x(3);   % Ap1_s = Ap1/Theta_u
Ap2_s = x(4);   % Ap2_s = Ap2/Theta_u
Sin1_s = x(5);  % Sin1_s = s_in1/Theta_s
Sin2_s = x(6);  % Sin2_s = s_in2/Theta_s
Se1_s = y(1);   % Se1_s = s1_e/Theta_s
Se2_s = y(2);   % Se2_s = s2_e/Theta_s

%Dynamics of internal variables inside each cell.
%SIMPLIFIED MODEL:
dxdt = [
    ( Alfa_p_s/(1 + P2_s^n) ) - P1_s + Beta_u_s * Ap2_s^n/( 1 + (Ap2_s^n) );
    ( Alfa_p_s/(1 + P1_s^n) ) - P2_s + Beta_u_s * Ap1_s^n/( 1 + (Ap1_s^n) );
    Alfa_Ap_s_1 * Sin1_s^n/(1 + Sin1_s^n) - K_s * Ap1_s * Ap2_s - Ap1_s;
    Alfa_Ap_s_2 * Sin2_s^n/(1 + Sin2_s^n) - K_s * Ap1_s * Ap2_s - Ap2_s;
    cs_s * P1_s - Sin1_s + re_s * (Se1_s - Sin1_s);
    cs_s * P2_s - Sin2_s + re_s * (Se2_s - Sin2_s)
    ];

end
```

# Appendix 4

```
%THIRD TEST: ROBUSTNESS TO CELLS HETEROGENEITY

%The objective is to analyze the robustness of the design to cell-to-cell
variability

%The parameters of each cell in the consortium are changed by drawing them
from a normal distribution centered in the nominal values.
%The covariance of this normal distribution is varied from 0-50%

%Then we illustrate the evolution of the mean relative error and mean
%settling time with the different values of covariance

clc
close all
clear

% Order of the variables inside the cell:
% x(1)=P1, % x(2)=P2, x(3)=Ap1, x(4)=Ap2, x(5)=s_in1, x(6)=s_in2,

%Definition of the number of cells
N = 100;

%Definition of the Initial conditions inside the cell
x0 = [10,0,0,0,0,0]';   %p10=10, p20=0, Ap10=0, Ap20=0, s1_in0=0, s2_in0=0
n = length(x0);       %Number of variables inside the cell (6)

%Definition of the initial conditions of the external variables (quorum
sensing molecules)
y0 = [0, 0];   %s1_e0=0, s2_e0=0

%Definition of the simulation time
tf = 100;

kn = 20; %Number of different values of covariance

qn = 50; %Number of trials with each covariance

%Initialization of vectors
Error = zeros(1, qn);
SettleT = zeros(1, qn);

MeanError = zeros(1, kn);
MeanSettleT = zeros(1, kn);

%Vector with the nominal parameters
P_nom = [1, 3.5, 1, 10, 3, 4] ; %[Alfa_Ap_s, K_s, cs_s, re_s, Beta_u_s,
Alfa_p_s]

%Creation of kn covariances
CV = linspace(0,0.5,kn);
Param = [];
```

```
for k = 1 : kn

        disp("Evaluating cycle k = " + num2str(k))
        %Creation of vectors with N random numbers for each variable with
      each value of CV

        Alfa_Ap_s = (P_nom(1) + (CV(k) * P_nom(1)) * randn(N,1));
        K_s = (P_nom(2) + (CV(k) * P_nom(2)) * randn(N,1));
        Cs_s = (P_nom(3) + (CV(k) * P_nom(3)) * randn(N,1));
        re_s = (P_nom(4) + (CV(k) * P_nom(4)) * randn(N,1));
        Beta_u_s = (P_nom(5) + (CV(k) * P_nom(5)) * randn(N,1));
        Alfa_p_s = (P_nom(6) + (CV(k) * P_nom(6)) * randn(N,1));

        for q = 1 : qn

        %Computation of qn simulation for each set of variables
        %Creation of a vector of length N*n with all the initial conditions
      for the internal variables for the N number of cells
            X0 = [];

            for m = 1 : N

                X0( (m-1)*n+1 : m*n ) = [ rand(2,1) * 5 ; x0(3:n)];
                % [p1,p2] are randomly drawn with uniform distribution in
            [0;5]x[0;5]
            end

            %Creation of a vector of length N*n+2 with all the initial
            conditions for the internal variables for the N number of cells
            and the two external variables
            Z0 = [X0, y0];

            %Resolution of the whole system equations:
            [t,Z] =ode15s(@whole_sys_NumVal3,[0,tf],Z0,[],[n,N,Alfa_Ap_s',
            K_s',Cs_s',re_s',Beta_u_s',Alfa_p_s']);

            % output: stack matrix of all the systems

            %For the different parameters instead of Alfa_Ap_s: K_s,cs_s...
            % X is a matrix having n*N columns and length(t)rows
            % Y is a matrix having 2 columns and length(t)rows
            % Z is a matrix having (n*N + 4) columns and length(t)rows

            %Ratios computation
            P1 = Z(:, 1:n:N*n);
            %Creating matrix with all the P1 of each cell (txN)
            P2 = Z(:, 2:n:N*n);
            %Creating matrix with all the P2 of each cell (txN)

            %Creation of two sets A (Population 1) and B (Population 2)
            A = P1 > (2*P2);
            B = P2 > (2*P1);
            C = (P1 < 2*P2) & (P2 < 2*P1);

            r1 = sum(A,2)/N; %Vector of ratios of population 1 through time
            r2 = sum(B,2)/N; %Vector of ratios of population 2 through time
            r12 = sum(C,2)/N; %Vector of ratios of not_pop1 or not_pop2, so
            that r1+r2+r12=1
```

```
            % r is a matrix having 3 columns and length(t)rows
            r=[r1, r2, r12];

            %Relative error
            error = norm( r1(end) - 0.5 )/0.5;
            Error(q) =  error; % Vector with the relative errors

            %Settling time
            Y = r1(end);
            e = Y * 0.01; %Error
            st = 0;   %Initializing settling time

            for T = 1 : length(t)

                if (r1(T) < (Y + e)) && (r1(T) > (Y - e))
                    st = t(T);
                    break;
                end
            end

            SettleT(q) =  st;

        End

        MeanError(k) = mean(Error);
        MeanSettleT(k) = mean(SettleT);
        StdError(k) = std(Error);
        StdSettleT(k) = std(SettleT);
end

    figure(1)
    plot(CV,MeanError)
    grid on
    errorbar(CV,MeanError,StdError)
    xlabel('CV')
    ylabel('Mean Relative Error')
    legend('Relative Error')

    figure(2)
    plot(CV,MeanSettleT)
    grid on
    errorbar(CV,MeanSettleT,StdSettleT)
    xlabel('CV')
    ylabel('Mean Settling time (s)')
    legend('Settling time')
```

```
%FUNCTION FOR THE THIRD TEST

%In each simulation different values of the parameters for each cell are
introduced

function [dZdt] = whole_sys_NumVal3(t,Z,p)    %Z=Z0'

n = p(1);
N = p(2);

Alfa_Ap_s_vec = p(3 : 2 + N);
K_s_vec = p(3 + N : 2 + N*2);
cs_s_vec = p( 3 + N*2 : 2 + N*3);
re_s_vec = p(3 + N*3 : 2 + N*4);
Beta_u_s_vec = p(3 + N*4 :  2 + N*5);
Alfa_p_s_vec = p(3 + N*5 : 2 + N*6);

%Creation of a vector with s1_e, s2_e from Z
y = Z(end-1:end);      %y=[s1_e;s2_e]

%Creation of a vector N*n with all the internal variables of all cells at
%time t and creation of a vector with the parameters of each cell.
FZ = [];

for k = 1:N

    x = Z( (k - 1)*n+1 : k*n );  % x: sub vector of X with length n
    (variables of each cell)
    Param = [Alfa_Ap_s_vec(k), K_s_vec(k), cs_s_vec(k), re_s_vec(k),
    Beta_u_s_vec(k), Alfa_p_s_vec(k)];
    FZ = [FZ; single_cell(x,y,Param)];
end

v = Z(5:n:end); %Creating vector with all the s_in1 of each cell (length=N)
u = Z(6:n:end); %Creating vector with all the s_in2 of each cell (length=N)

% FY: Dynamics of global (external) concentration on quorum sensing molecules
that link together all the cell.
FY =  environ(v,u,y);

%%Creation of a vector of length N*n+2+2 with all the conditions of the
internal variables for the N number of cells and the two external variables
at t
dZdt = [FZ; FY];

end

function [dydt] = environ(v,u,y)

%%% s1_e and s2_e parameters
re_s = 10;                  % re/d
Gamma_e_s = 1.5;            %Gamma_e/d

%Variables definition
Se1_s = y(1);    % Se1_s = s1_e/Theta_s
Se2_s = y(2);    % Se2_s = s2_e/Theta_s
```

```
dydt =  [re_s * sum(v - Se1_s) - Gamma_e_s * Se1_s;
        re_s * sum(u - Se2_s) - Gamma_e_s * Se2_s];

end

%Creation of a vector with the computed internal variables at t for each
%cell
function [dxdt] = single_cell(x,y,Param)

%SIMPLIFICATION PARAMETERS
n = 2;                          %Hill coef
Alfa_Ap_s = Param(1);            %Alfa_Ap/(Theta_u*d)
K_s = Param(2);                  %(k*Theta_u)/d
cs_s = Param(3);                 %(cs*Theta_p)/(Theta_s*d)
re_s = Param(4);                 %re/d
Beta_u_s = Param(5);             %Beta_u/(Theta_p*d)
Alfa_p_s = Param(6);             %Alfa_p/(Theta_p*d)


%Variables definition
P1_s = x(1);     % P1_s = p1/Theta_p
P2_s = x(2);     % P2_s = p2/Theta_p
Ap1_s = x(3);    % Ap1_s = Ap1/Theta_u
Ap2_s = x(4);    % Ap2_s = Ap2/Theta_u
Sin1_s = x(5);   % Sin1_s = s_in1/Theta_s
Sin2_s = x(6);   % Sin2_s = s_in2/Theta_s
Se1_s = y(1);    % Se1_s = s1_e/Theta_s
Se2_s = y(2);    % Se2_s = s2_e/Theta_s

%Dynamics of internal variables inside each cell.
%SIMPLIFIED MODEL:
dxdt = [
    ( Alfa_p_s/(1 + P2_s^n) ) - P1_s + Beta_u_s * Ap2_s^n/( 1 + (Ap2_s^n) );
    ( Alfa_p_s/(1 + P1_s^n) ) - P2_s + Beta_u_s * Ap1_s^n/( 1 + (Ap1_s^n) );
    Alfa_Ap_s * Sin1_s^n/(1 + Sin1_s^n) - K_s * Ap1_s * Ap2_s - Ap1_s;
    Alfa_Ap_s * Sin2_s^n/(1 + Sin2_s^n) - K_s * Ap1_s * Ap2_s - Ap2_s;
    cs_s * P1_s - Sin1_s + re_s * (Se1_s - Sin1_s);
    cs_s * P2_s - Sin2_s + re_s * (Se2_s - Sin2_s)
    ];

end
```