
Supplementary materials for
**Biological oscillations without genetic
oscillator or external forcing**

Vincent Vandenbroucke^{a,1}, Lucas Henrion^{a,1}, and Frank Delvigne^{a,2}

¹ V.V. and L.H. contributed equally to this work.

² To whom correspondence should be addressed. E-mail: F.Delvigne@uliege.be

^a Terra Research and Teaching Centre, Microbial Processes and Interactions (MiPI), Gembloux
Agro-Bio Tech, University of Liège, Gembloux, Belgium

Contents

1	Stress model in continuous culture	2
1.1	When $S > K_I$	4
1.2	When $S \leq K_I$	7
1.3	Solution of the steady-state	10
2	Limitations of the model and additional notes	11
3	Growth and switching parameter determination	12
4	Code and Data Analysis	13
4.1	Matlab code	13
4.2	Python script for data representation	14
5	Dataset legends	23
	References	24

1 Stress model in continuous culture

A chemostat is usually modelled with the following equations:

$$\frac{dS}{dt} = (S_{in} - S)D - \frac{\mu X}{Y} \quad (\text{S.1})$$

$$\frac{dX}{dt} = (\mu - D)X \quad (\text{S.2})$$

However, often, cells have a second state that can be induced when they are stressed (in our case, the switching of *B. subtilis* to spore formation), in which case their growth rate decreases. We can represent those two states in a chemostat with the following equations, if we consider that the stressed state is irreversible and cannot go back to non-stressed. As the time required to dilute the stress proteins is quite long, and made longer by the lower growth rate caused by stress, switching to a stress state is much faster and this is a reasonable assumption for a simplified approach. That is without considering the fact that, in a chemostat with non-stressed cells, the stressed-cells would be overtaken slowly and, if not renewed, are likely to simply be washed out before they can go back to their non-stressed state. This general stress model can be reduced further in the case of *B. subtilis*, as spores do not grow at all and the growth rate of the stressed phenotype is zero ($k = 0$ below), but the general case is solved here.

$$\frac{dS}{dt} = (S_{in} - S)D - \frac{\mu X_1}{Y_{X1}} - \frac{k\mu X_2}{Y_{X2}} \quad (\text{S.3})$$

$$\frac{dX_1}{dt} = (\mu - D)X_1 - H_1(S)X_1 \quad (\text{S.4})$$

$$\frac{dX_2}{dt} = (k\mu - D)X_2 + H_1(S)X_1 \quad (\text{S.5})$$

$$\mu = \mu_{max} \frac{S}{K_S + S} \quad (\text{S.6})$$

$$\mu_{max,1} = \mu_{max} \quad (\text{S.7})$$

$$\mu_{max,2} = k\mu_{max} \quad (\text{S.8})$$

$$H_1(S) = \begin{cases} H & \text{if } S \leq K_I \\ 0 & \text{if } S > K_I \end{cases} \quad (\text{S.9})$$

Where the variables are defined as:

- X_1 the concentration of the fast growing phenotype in the reactor.
(in the case of *B. subtilis*, vegetative, non-stressed cells)
- X_2 the concentration of the slow growing phenotype in the reactor.
(in the case of *B. subtilis*, dormant spores)
- Y_{X1} the yield of the fast growing phenotype.
- Y_{X2} the yield of the slow growing phenotype.
- D the dilution rate.
- k the ratio of the growth rate of the slow growing phenotype to the
growth rate of the fast growing phenotype.
- K_I the threshold substrate concentration for the switch.
- K_S the half-saturation constant of the Monod equation.
- H the switching rate of the fast growing to the slow growing phenotype.
- $H_1(S)$ the switch function from the fast growing to the slow growing phenotype.
- μ the instantaneous growth rate of the fast growing phenotype.
- μ_{max} the maximum growth rate of the fast growing phenotype.
- S the substrate (usually glucose) concentration in the reactor.
- S_{in} the substrate concentration in the feed.
- t the time.

We can note the following conditions on the parameters based on the parameters:

- All variables are positive due to the constraints of reality and considering growing phenotypes.
- D , Y_{Xi} , μ_{max} , K_S , K_I and H are strictly positive.

Considering those equations for a chemostat, the question that now occurs is:
What happens at steady state?

Due to the switch to a stress conditions, we have two kinds of steady-state to consider: ones where the substrate concentration is above the threshold K_I , and ones where the substrate concentration is below the threshold K_I . Separating those cases let us replace $H_1(S)$ by H or 0 in the equations, greatly simplifying the system. Note that by definition, the steady-state requires the residual substrate concentration S to be constant, therefore all the possible steady-state can be found by splitting the equations this way.

To jump to the solution of the steady-state, go to section 1.3.

1.1 When $S > K_I$

In this case, at steady-state (derivatives are null), there is no switching and we can replace equations S.3 to S.5 by the following:

$$0 = (S_{in} - S)D - \frac{\mu X_1}{Y_{X1}} - \frac{k\mu X_2}{Y_{X2}} \quad (\text{S.10})$$

$$0 = (\mu - D)X_1 \quad (\text{S.11})$$

$$0 = (k\mu - D)X_2 \quad (\text{S.12})$$

According to the latter 2 equations, there are 4 possible cases:

1. $X_1 = 0$ and $X_2 = 0$.
2. $X_1 = 0$ and $D = k\mu$.
3. $X_2 = 0$ and $D = \mu$.
4. If $k = 1$, $D = \mu$

These are the expected steady-states and show already known results: in the absence of switching, at steady-state, either the two populations are washed out or not initially present (case 1); and there is a marginally stable case where if both phenotypes have exactly equal growth rates, they can both survive, in any proportion as they behave as a single population and this whole model becomes irrelevant (case 4).

For the sake of this argument, we can consider $k \neq 1$ as that is a trivial and uninteresting case.

Both phenotypes cannot coexist in this case if $k \neq 1$, as one existing, in the absence of switching, imposes a specific dilution rate that implies, at steady-state, the other no being there.

The last case, a wash-out of both/either phenotype, will always happen for a population if $(\mu - D) = 0$ is impossible. To show when it is forced, finding out when it is possible is the easiest way.

Then, re-writing equations S.10 to S.12 for a steady-state with a single phenotype:

$$(S_{in} - S)D = \frac{X_i}{Y_{Xi}} \mu_{max,i} \frac{S}{K_S + S} \quad (\text{S.13})$$

$$\mu_{max,i} \frac{S}{K_S + S} = D \quad (\text{S.14})$$

where i indicates either phenotype, whichever is present since we established that in this case only one can be present.

Since Y_{Xi} , D and K_S are strictly positive, we can deduce:

$$\mu_{max,i} \frac{S}{K_S + S} = D \quad \Rightarrow \quad \frac{D}{\mu_{max,i}} = \frac{S}{K_S + S} \quad (\text{S.15})$$

$$S_{in} - S = \frac{X_i}{Y_{Xi}} \quad \Rightarrow \quad S = S_{in} - \frac{X_i}{Y_{Xi}} \quad (\text{S.16})$$

Therefore:

$$\begin{aligned} \frac{D}{\mu_{max,i}} &= \frac{S}{K_S + S} \\ &= 1 - \frac{K_S}{K_S + S} \\ \Leftrightarrow \frac{K_S}{K_S + S} &= 1 - \frac{D}{\mu_{max,i}} \\ &= \frac{\mu_{max,i} - D}{\mu_{max,i}} \\ \Leftrightarrow K_S + S &= \frac{\mu_{max,i} K_S}{\mu_{max,i} - D} \\ \Leftrightarrow S &= \frac{\mu_{max,i} K_S}{\mu_{max,i} - D} - K_S \\ &= \frac{DK_S}{\mu_{max,i} - D} \end{aligned} \quad (\text{S.17})$$

Plugging equation S.17 into equation S.16 and solving for X_i gives:

$$\begin{aligned} S_{in} - \frac{DK_S}{\mu_{max,i} - D} &= \frac{X_i}{Y_{Xi}} \\ X_i &= Y_{Xi} \left(S_{in} - \frac{DK_S}{\mu_{max,i} - D} \right) \end{aligned} \quad (\text{S.18})$$

Thus, eventually, from equations S.15, S.16, S.17, and S.18 we get:

$$\begin{cases} D &= \mu_{max,i} \frac{S}{K_S + S} \\ S &= \frac{DK_S}{\mu_{max,i} - D} = S_{in} - \frac{X_i}{Y_{Xi}} \\ X_i &= Y_{Xi} \left(S_{in} - \frac{DK_S}{\mu_{max,i} - D} \right) = Y_{Xi}(S_{in} - S) \end{cases} \quad (\text{S.19})$$

at steady states, for cases 2 and 3. And since $S \geq 0$ and $X_i \geq 0$, for this to be possible, we need:

- $\mu_{max,i} > D$ for $S \geq 0$. When a given phenotype is initially present, a wash-out and no biomass is leftover at steady-state (it becomes case 1) if the maximum growth rate of the phenotype is lower than the dilution rate (as expected).

- $S_{in} > DK_S/(\mu_{max,i} - D)$ for $X_i > 0$. This can be rearranged (as per standard chemostat equations) as $D < \mu_{max,i} S_{in}/(K_S + S_{in}) < \mu_{max,i}$, dilution rate above which there is no glucose consumption at equilibrium as there is not enough substrate to sustain the population.

Those equations and deductions are only valid when $S > K_I$, or:

$$\frac{DK_S}{\mu_{max,i} - D} > K_I \quad (\text{S.20})$$

Firstly, we established that, when there is no wash-out, and there is biomass at steady-state, $\mu_{max,i} > D$ is required. Therefore, the left hand-side of the inequality is always positive.

We can examine when transform the inequality to examine when it is possible or:

$$\begin{aligned} & \frac{DK_S}{\mu_{max,i} - D} > K_I \\ \Leftrightarrow & DK_S > K_I(\mu_{max,i} - D) && \text{since: } \mu_{max,i} > D \\ \Leftrightarrow & DK_S + DK_I > K_I\mu_{max,i} \\ \Leftrightarrow & D(K_S + K_I) > K_I\mu_{max,i} \\ \Leftrightarrow & D > \frac{K_I\mu_{max,i}}{K_S + K_I} && \text{since: } K_S, K_I > 0 \end{aligned} \quad (\text{S.21})$$

Therefore, steady-state without switching can be reached when $\mu_{max,i} S_{in}/(S_{in} + K_S) > D > \mu_{max,i} K_I/(K_S + K_I)$ for each phenotype if the three parameters allow (although, for the stressed phenotype, we are assuming that there is no switch back to the first one, which is probably wrong if waiting long enough in this high substrate concentration case).

As a summary of this section, we have for each case described at the start:

1. Wash out of all the cells initially present, when $D > \mu_{max,i} \frac{S_{in}}{(S_{in} + K_S)}$ for all i phenotypes non-zero in the initial conditions.
2. a steady state with only the second phenotype present, which could happen if all cells switched to the second phenotype, and never switched back, maybe in the case of differentiation:

$$\begin{cases} X_1 = 0 \\ D = k\mu_{max} S/(K_S + S) \\ S = DK_S/k\mu_{max} - D = S_{in} - X_2/Y_{X2} \\ X_2 = Y_{X2}(S_{in} - S) \\ 0 \leq S \leq S_{in} \\ k\mu_{max} S_{in}/(S_{in} + K_S) > D > k\mu_{max} K_I/(K_S + K_I) \end{cases} \quad (\text{S.22})$$

3. A steady state with only the first phenotype present:

$$\begin{cases} X_2 = 0 \\ D = \mu_{max} S / (K_S + S) \\ S = DK_S / (\mu_{max} - D) = S_{in} - X_1 / Y_{X1} \\ X_1 = Y_{X1} (S_{in} - S) \\ 0 \leq S \leq S_{in} \\ \mu_{max} S_{in} / (S_{in} + K_S) > D > \mu_{max} K_I / (K_S + K_I) \end{cases} \quad (\text{S.23})$$

1.2 When $S \leq K_I$

A similar reasoning can be applied to the alternative case, where the switching always occurs.

At steady-state, we get:

$$0 = (S_{in} - S)D - \frac{\mu X_1}{Y_{X1}} - \frac{k\mu X_2}{Y_{X2}} \quad (\text{S.24})$$

$$0 = (\mu - D - H)X_1 \quad (\text{S.25})$$

$$0 = (k\mu - D)X_2 + HX_1 \quad (\text{S.26})$$

The equations can be solved easily when $X_1 = 0$ and $X_2 = 0$, as this means there is no biomass. Notably, this case is incompatible with the assumption that there is switching regardless as $S_{in} = S$ in that case, and in a bioreactor $S_{in} > K_I$, so we come back to the previous section.

In case $X_1 = 0$, the equations become identical to the previous section. The only difference is the condition of existence derived at the end, which becomes $D < \mu_{max,i} K_I / (K_S + K_I)$ instead. This completes all cases where only the second phenotype is present with a region where it is actually likely that all cells stay in the second phenotype, as the substrate concentration would be too low to permit switching back to the first phenotype.

Thus the only case of interest to discuss here is when both X_1 and X_2 are non-zero.

Re-arranging equations S.24 to S.26 with those assumptions:

$$(S_{in} - S)D = \frac{\mu X_1}{Y_{X1}} - \frac{k\mu X_2}{Y_{X2}} \quad (\text{S.27})$$

$$\mu = \mu_{max} \frac{S}{K_S + S} = D + H \quad (\text{S.28})$$

$$(D - k\mu)X_2 = HX_1 \quad (\text{S.29})$$

Interestingly, equation S.28 implies that the first phenotype, through D , H , μ_{max} and K_S , fully determines the substrate concentration, with only H differing from a normal chemostat and no direct influence of the phenotype ratio in the culture.

Based on equation S.28, we can deduce:

$$\begin{aligned}
 S &= \frac{(K_S + S)(D + H)}{\mu_{max}} \\
 \Leftrightarrow S \left(1 - \frac{D + H}{\mu_{max}}\right) &= K_S \frac{D + H}{\mu_{max}} \\
 S \frac{\mu_{max} - D - H}{\mu_{max}} &= \\
 \Leftrightarrow S &= \frac{K_S(D + H)}{\mu_{max} - D - H}
 \end{aligned} \tag{S.30}$$

Equation S.29 can easily be re-arranged into:

$$X_1 = \frac{D - k\mu}{H} X_2 \Leftrightarrow X_2 = \frac{H}{D - k\mu} X_1 \tag{S.31}$$

From there, we can plug into the remaining equation S.27:

$$\begin{aligned}
 (S_{in} - S) D &= \frac{\mu X_1}{Y_{X1}} + \frac{k\mu}{Y_{X2}} \frac{H}{D - k\mu} X_2 \\
 \Leftrightarrow X_1 &= \frac{(S_{in} - S) D}{\frac{\mu}{Y_{X1}} + \frac{k\mu H}{Y_{X2} (D - k\mu)}}
 \end{aligned} \tag{S.32}$$

$$\begin{aligned}
 \text{and} \quad (S_{in} - S) D &= \frac{(D - k\mu)\mu}{HY_{X1}} X_2 + \frac{k\mu X_2}{Y_{X2}} \\
 \Leftrightarrow X_2 &= \frac{(S_{in} - S) D}{\mu \left(\frac{(D - k\mu)}{HY_{X1}} + \frac{k}{Y_{X2}} \right)}
 \end{aligned} \tag{S.33}$$

We can thus write the solution for this case of the steady-state as:

$$\begin{cases} S = \frac{K_S(D+H)}{\mu_{max}-D-H} \\ \mu = \mu_{max} \frac{S}{K_S+S} = D + H \\ X_1 = \frac{(S_{in}-S) D}{\frac{\mu}{Y_{X1}} + \frac{k\mu H}{Y_{X2} (D - k\mu)}} \\ X_2 = \frac{(S_{in}-S) D}{\mu \left(\frac{(D - k\mu)}{HY_{X1}} + \frac{k}{Y_{X2}} \right)} \\ \frac{X_1}{X_2} = \frac{D - k\mu}{H} \end{cases} \tag{S.34}$$

For this solution to be possible, we need:

- $S > 0$
- $X_1 > 0$
- $X_2 > 0$

Since $X_1 = X_2^{(D-k\mu)/H}$, we need for both to be positive:

$$D > k\mu \quad (\text{S.35})$$

Since $D + H = \mu$, we actually need:

$$\frac{D}{D+H} > k \quad (\text{S.36})$$

Note that this condition can only be true if $k < 1$, as the left hand side is always smaller than 1. Thus, if the induced phenotype were to grow faster than the initial one, it would be impossible for the two phenotypes to coexist in the reactor. In this case, we are considering a stress phenotype, so $k < 1$ is a reasonable assumption.

In other musings, this condition means at its origin that the dilution rate will be faster than the instantaneous growth rate of the second phenotype.

The condition can be easily re-arranged in terms of D as well:

$$\frac{D}{D+H} > k \quad \Leftrightarrow \quad D > \frac{kH}{1-k} \quad (\text{S.37})$$

This ensures that both X_1 and X_2 have the same sign. To ensure they are positive, we need:

$$S_{in} > S \quad (\text{S.38})$$

which is always true if cells grow.

Finally, we need for $S \geq 0$:

$$D < \mu_{max} - H \quad (\text{S.39})$$

The latter also implies that, for both phenotypes to coexist,

$$\mu_{max} > H \quad (\text{S.40})$$

is required.

Those conditions, combined with the one required by this section, $S \leq K_I$, are the only conditions required for the existence of a steady-state with both phenotypes present.

The latter condition can be developed:

$$\begin{aligned} & \frac{K_S(D+H)}{\mu_{max} - (D+H)} < K_I \\ \Leftrightarrow & K_S(D+H) < K_I (\mu_{max} - (D+H)) \quad \text{"<" sign unchanged since (S.39)} \\ \Leftrightarrow & DK_S + HK_S < \mu_{max}K_I - DK_I - HK_I \\ \Leftrightarrow & D(K_S + K_I) < \mu_{max}K_I - H(K_S + K_I) \\ \Leftrightarrow & D < \frac{\mu_{max}K_I}{K_S + K_I} - H \end{aligned} \quad (\text{S.41})$$

Similarly to condition S.40, we can deduce that

$$\mu_{max} \frac{K_I}{K_S + K_I} > H \quad (\text{S.42})$$

from the previous condition.

Note that condition S.41 is stronger than condition S.39, as:

$$\begin{aligned} \mu_{max} - H &> \frac{\mu_{max} K_I}{K_S + K_I} - H \\ \Leftrightarrow \mu_{max} &> \mu_{max} \frac{K_I}{K_S + K_I} \end{aligned} \quad (\text{S.43})$$

Inequation S.43 is always true, as $\frac{K_I}{(K_S + K_I)}$ is smaller than one, so condition S.39 is irrelevant. The deduced condition on the relationship is similarly stronger in (S.42) than in (S.40).

In summary, the steady-state with both phenotypes present is possible with the non-trivial conditions:

- $(\mu_{max} K_I)/(K_S + K_I) - H > D > kH/(1 - k)$
- $\mu_{max} K_I/(K_S + K_I) > H$

1.3 Solution of the steady-state

Overall, the steady-state of the cells can be described as a function of an increasing dilution rate:

X_2 only :	$\begin{cases} D < k\mu_{max} K_I/(K_S + K_I) \\ k\mu_{max} K_I/(K_S + K_I) < D < k\mu_{max} S_{in}/(K_S + S_{in}) \\ k\mu_{max} S_{in}/(K_S + S_{in}) < D \end{cases}$	Only X_2 is present, no induction Only X_2 is present, but $S > K_I$ Wash-out of X_2 only
X_1 and X_2 :	$\begin{cases} kH/(1 - k) < D < \mu_{max} K_I/(K_S + K_I) - H \end{cases}$	$X_1/X_2 = (D - k\mu)/H$
X_1 only :	$\begin{cases} \mu_{max} K_I/(K_S + K_I) < D < \mu_{max} S_{in}/(K_S + S_{in}) \\ \mu_{max} S_{in}/(K_S + S_{in}) < D \end{cases}$	Only X_1, no induction Wash-out

For X_2 only, while there are multiple cases, when those cases are overlapping with another region, then that other region is most likely to be reached instead: other regions include X_1 , and while our assumptions include that there is no switch from X_2 to X_1 , that assumption would likely prove wrong in the long term. In particular, the region where X_2 is the only phenotype present and $S > K_I$ (inhibiting the switch from X_2 to X_1) is probably not stable and thus we will not consider it further.

Interestingly, since no assumption can be made on the relative values of the different parameters, the different regions do not always define the entire range of dilution rates.

Between the regions where only X_2 and both phenotypes are present, there may be a region without steady-state if $kH/(1-k) > k\mu_{max}^{K_I/(K_S+K_I)}$. Between those two thresholds of dilution rates, the substrate concentration would be too high to keep inducing the second phenotype.

Between the regions where only X_1 and both phenotypes are present, there may be a region without steady-state if $\mu_{max}^{K_I/(K_S+K_I)} > \max\{\mu_{max}^{K_I/(K_S+K_I)} - H, k \cdot (\mu_{max}^{K_I/(K_S+K_I)})\}$, where the second term of the maximum is in case the X_2 only region reaches beyond the cohabitation region. Whichever is the maximum, this inequality is always true (because of the $-H$ or because $k < 1$). Therefore, there is always a region without steady-state when decreasing the dilution rate and the stressed phenotype starts to be induced. This interval is the one discussed in the main paper.

In those regions without steady-state, any attempt to reach steady-state when $S > K_I$ causes the biomass to increase and S to decrease. When $S < K_I$, the cells start switching and substrate consumption decreases, causing S to increase above K_I again. Since cells do not respond instantaneously to change, and in this region where S is close to K_I small changes in concentration can cause large changes in responses, the cells may oscillate between the two states.

In the specific case of *B. subtilis* sporulation, spores do not grow. Therefore, the region with no fast growing/unstressed cells is not possible. In addition, the switching rate H that we measured (see section 3) is high enough that the threshold for a steady-state with X_1 and X_2 is negative, $\mu_{max}^{K_I/(K_S+K_I)} - H < 0$. In other words, for our system, there are two possibilities: either only the fast growing phenotype X_1 is present, or there are oscillations, the threshold between these states being $\mu_{max}^{K_I/(K_S+K_I)}$.

2 Limitations of the model and additional notes

There are a few additional things that can be added on this model, in no particular order:

- This model predicts that there is a region where there is no steady-state when there are sharp transitions in cell metabolism, such as with stress induction, or induction of burdensome production with lactose. Oscillations may appear thanks to the delay in response time of the cells, and very small changes in substrate concentration causing large changes in cell response. However, the switch is not a perfect step function, and the delay will be shorter or longer depending on the cells. Further, a constant switching speed may not be a good approximation, at least when close to the switching threshold. All in all, oscillations may appear, but they may be dampened, or the system may stabilize quickly in an intermediate state that this model is incapable of representing.
- In the demonstration, we considered what happens when $K_I < S$ and $K_I > S$. When $K_I = S$, the details of what happen in the model depend on how $H_1(S)$ was defined (here, $H_1(K_I) = H$, therefore the $K_I = S$ would be the limit of the $K_I < S$ case). In practice, however, reality is not completely discontinuous, and this is the edge of an undefined region regardless, so the mathematics of this particular fictional point do not really matter, only the limits of the region between cases does.
- We consider here that both phenotypes are growing, no matter how slowly. If the slow growing phenotype were to have a death rate instead ($k < 0$), several things would happen:

- No region would exist where only the second phenotype can survive, as even a batch culture would eventually have no cells left. This is expressed by condition S.37 being always true.
- Nothing else would change, as in no place in section 1.2 is there any division by k for any condition. We can therefore expect exactly the same dynamics and region of uncertainty as with $k \geq 0$.

3 Growth and switching parameter determination

A triplicate batch cultivation in the DASBox was performed to estimate μ_{max} and switching (K_I , H) parameters of our strain. To this end, each bioreactor was inoculated with a OD of roughly 0.08 and samples were collected throughout the batch to assess the OD and the residual glucose concentration. The residual glucose concentration was determined using a D-glucose enzymatic assay kit from Megazyme in 98-well plates with a plate reader. During the batch, FC analysis was performed with our custom platform, the Segregostat, to observe when and at what speed cells were sporulating. The K_S was defined based on the analysis the residual glucose concentration at steady state at a high dilution rates (where cells are not sporulating) during the chemostats with varying dilution rates.

The following parameters were measured or estimated:

- $H = 1.1 \text{ h}^{-1} \rightarrow$ Average from the three batches where it was measured, which presented 0.83, 1.2, and 1.3 with $r_2 = 0.98, 0.99$ and 0.99 respectively (computed by linear regression over the time when there is switching)
- $K_S = 0.14 \text{ g/L} \rightarrow$ Based on the concentration at $D=0.32 \text{ h}^{-1}$, measured as 0.2761 and 0.2432 g/L. From $D = \mu_{max} S / (K_S + S)$, we get $K_S = \mu_{max} S / D - S$
- $K_I \in [0.01, 0.1] \text{ g/L}$, and we used 0.075 g/L $\rightarrow$ Explained below
- $\mu_{max} = 0.49 \text{ h}^{-1} \rightarrow$ Average from the three maximum growth rates measured in batch (0.52, 0.47 and 0.48 h^{-1} , measured by linear regression over the time of exponential growth, see code with $r^2 > 0.99$ for all three)
- $k = 0 \rightarrow$ Sporulating cells are considered non-growing
- For the model, Y_{X1} was set at $0.5 \text{ g}_{biomass} / \text{g}_{substrate}$ as a standard value. The previous analysis shows that this parameter does not affect the equilibrium directly: the dilution rate values where the instability appear are independent from the biomass yield, and, given that $k = 0$, Y_{X2} becomes irrelevant and Y_{X1} only appears as a scaling factor for the amount of biomass present in the bioreactor, which does not impact the population dynamics.
- A delay of 0.5 h was used for the model, which corresponds to half the shortest bursts of expression we observed (about 1 h). Any value of delay, however, caused sustained oscillations, from 0.01 h to 10 h.

To determine the start of sporulation, we considered the time at which the measured optical density dropped in the batches. That drop likely corresponds to a change in morphology of the cells. For that stretch of time, samples were taken more often than the automated flow cytometry, providing a better time resolution, in addition to having no absolutely no cross-contamination (using the segregostat with our DasGip in triplicate setup, a few percent leftover from each sample may be measured with the next sample), which is relevant when looking at very low percentages of induction. Based on these OD measurements, we considered the induction threshold was passed during the preceding measured time interval. This corresponds to 0.013 to 0.097 g/L, 0.009 to 0.068 g/L and 0.010 to 0.282 g/L for reactors 1, 2 and 3 respectively. Thus we estimated K_I to be in the range of 0.01 to 0.1 g/L, although considering the cells need time to switch, K_I 's order of magnitude is likely the higher end of the range.

Based on these measurements, we could compute the unstable range of dilution rates for *B. subtilis*. The lower end of the range is at $D = 0^{-1}$, as $k = 0$ and $(\mu_{max}K_I)/(K_S + K_I) - H < 0$ ($\mu_{max} < H$). The upper end depends on K_I . Based on the estimated range of K_I , it should be between $(\mu_{max}K_I)/(K_S + K_I) \in [0.03, 0.2] \text{ h}^{-1}$. The first replicate of the experiment showcasing a range of dilution rates showed the start of the instability at $D = 0.17 \text{ h}^{-1}$, which would correspond to a K_I of 0.075 g/L.

4 Code and Data Analysis

Three different codes were used for this paper, all are available in the supplementary dataset. First, a short Matlab code was used to simulate the system with a delay.

Second, a data analysis Python script was used to estimate the parameters from the experimental data. Due to its length, only the essential part of the code is presented in this document (the complete code is available in the supplementary dataset).

Third, a small data analysis Python script was used to plot the experimental data. The Python data analysis relies on the previously published toolbox [1] at <https://gitlab.uliege.be/mipi/published-software/mbiomass-core>. All the datasets used in this paper are available (<https://doi.org/10.5281/zenodo.13803913>).

4.1 Matlab code

```

1 % system:
2 % dS= (Sin - S)D - muX1/YX1 - kmuX2/YX2
3 % dX1 = (mu - D)X1 - H1 (S)X1
4 % dX2 = (kmu - D)X2 + H1 (S)X1
5 % mu = mumax*S/(KS + S)
6 % mumax,1 = mumax
7 % mumax,2 = kmumax
8 % if S <= KI
9 % H1 (S) = H
10 % if S > KI
11 % H1 (S) = 0

```

```

12
13 sol = dde23(@diff_test1,[0.5],[0.02,2,0],[0,80])
14
15 % plot with X1 and X2 on left axis and S on right axis:
16 plot(sol.x,sol.y(2:3,:))
17 yyaxis right
18 plot(sol.x,sol.y(1,:), 'k')
19 % add horizontal line at KI
20 hold on
21 plot([0,60],[0.075,0.075], 'k--')
22 legend('X1','X2','S')
23
24 % save sol.x and sol.y to csv
25 writematrix([sol.x',sol.y'], 'test1.csv')
26
27 function diff=diff_test1(~, y, Z)
28     % time unit: hour
29     H=1.1;
30     % k=0 so things easier
31     KS=0.14;
32     KI=0.075; % with D threshold=0.17 h-1 as experimented
33
34     Sin=5;
35     YX1=0.5; % standard value
36
37     D=0.1; % dilution rate
38     mu_max=0.49;
39
40     diff=zeros(3,1);
41     diff(1)= (Sin - y(1))*D-mumax*y(1)/(KS+y(1))*y(2)/YX1;
42     diff(2)= (mumax*y(1)/(KS+y(1))-D)*y(2)-H*(Z(1)<KI)*y(2);
43     diff(3)= H*(Z(1)<KI) * y(2) - D*y(3);
44 end

```

4.2 Python script for data representation

```

1 import os
2 import datetime
3 import glob
4 import numpy as np
5 import pandas as pd
6 import mbiomas.pipes
7 import mbiomas.read
8 import mbiomas.fca
9 import matplotlib.pyplot as plt

```

```

10 from scipy.stats import linregress
11
12
13 mbiomas.pipes.write_version()
14
15 # standard reading data from dasgip:
16
17 def read_old_dasgip(folder, zero_time, time_zone, reactor_equivalence,
18                     fields='all',
19                     meta_encoding='utf-8',
20                     time_offset_das=0):
21     """
22     Read data from the old DASGIP bioreactors (300 mL vessels from before 2020)
23     This may include:
24     * Online Flow cytometry data
25     * Oxygen data, either in the root folder, or in an 'Oxygen' subfolder
26       expected naming scheme:
27       '*-chx.txt' where x is the probe number
28     * Any number of offline data files, all the csv files.
29       To indicate reactor number, the file should be named as:
30       '*_x.csv' where n is the reactor number.
31       (offline data is expected in local time)
32       Metadata for those files should be named as the file, but with the
33       extension `*.meta.csv`.
34
35     Parameters
36     -----
37     folder : str
38         Path to the folder containing the data files
39     zero_time : datetime.Datetime or string
40         Time at which the zero should be set. It is expected in local time.
41     time_zone : str
42         A number between -12 and 12, which indicates the
43         difference between the zero time and the local time (UTC+2 → 2).
44     reactor_equivalence : np.array
45         Array with the equivalence between the different reactor numbers.
46         Each line is a vessel.
47         * The first column is the reactor number in the DASGIP software
48         * The second column is the reactor number in the Oxygen software
49         * The third column is the reactor number in the flow cytometry data
50         * The fourth column is the reactor number in the offline data
51         A '-1' indicates that the reactor is not present in the data.
52         The order used is the one of the vessels in the matrix,
53         top to bottom.
54     fields : list

```



```

55         List of fields to be read from the offline files. Will need to change so
56         the default is all of them.
57     meta_encoding : str
58         Encoding of the offline data metadata files. Default is 'utf-8'
59     time_offset_das : float
60         Time offset in hours in the DasGIP computer. Default is 0. For example:
61         Real time is 10:40, but the DasGIP computer is set to 10:50. Then the
62         offset is  $5/6 - 4/6 = 1/6 = 0.1667$ .
63     """
64     (... see supplementary dataset)
65
66     t=read_old_dasgip('batch',
67                     '2024-03-06 09:55',
68                     2,
69                     np.array(
70                         [[-1,2,1,1],
71                         [-1,3,2,2],
72                         [-1,4,3,3]]),
73                     time_offset_das=-52/60)#8 minutes offset + 1h to real time
74
75     # remove useless columns
76     fields=['FL2-A', 'FL3-A', 'FL4-A', 'FL2-H', 'FL3-H', 'FL4-H', 'Width', 'Time']
77     for set_data in t:
78         # for field in ['FL2-A', 'FL3-A', 'FL4-A',
79         #               'FL2-H', 'FL3-H', 'FL4-H', 'Width', 'Time']:
80         set_data[3].remove_fields(fields)
81
82
83     # H time threshold (see processing later):
84     H_time_threshold=np.array([[12.6, 17],
85                               [14.5, 18],
86                               [14.2, 18]])
87
88     # fluorescence threshold
89     threshold=400
90
91     # save the measured parameters to a csv file through a pandas dataframe
92     parameters=pd.DataFrame(index=[0,1,2])
93
94     for i,set_data in enumerate(t):
95
96         set_data[2], set_data[3],_=mbiomas.fca.data.clean(set_data[2], set_data[3])
97
98         fig, ax=plt.subplots()
99

```

```

100 plot=mbiomas.fca.plot.Plot(set_data[2])
101
102 plot.time_scatter(ax,set_data[2],set_data[3],y_label='FL1-A',
103                  y_log_lim=[0,5])
104
105 ## plot the percentage of cells above fluorescence threshold to determine
106 ## H the switching rate
107
108 fig, ax=plt.subplots()
109
110
111 # make an empty DF to store the results
112 percent_positive=pd.DataFrame(index=set_data[2].data["Time"])
113
114 # add absolute time too
115 # do it with a loop because it creates NaT otherwise
116 # probably due to differing index formats
117 # percent_positive.loc[:, 'time']=set_data[2].data["Time"]
118 # print(set_data[2].data["btimdilution"])
119 # print(percent_positive)
120 for j in range(len(percent_positive)):
121     percent_positive.loc[percent_positive.index[j], 'Absolute time']= \
122         set_data[2].data["btimdilution"][j]
123
124
125 for j in range(len(percent_positive)):
126     percent_positive.loc[percent_positive.index[j], 'percent']= \
127         (set_data[3].data[j]['FL1-A']>400).sum()/ \
128         len(set_data[3].data[j]['FL1-A'])*100
129
130 # save to csv
131 percent_positive.to_csv('percent_positive_reactor'+str(i+1)+'.csv')
132
133 plt.plot(percent_positive.index, percent_positive['percent'])
134 ax.set_yscale('log')
135 plt.title('Percentage of cells above threshold '+str(threshold)+
136          ' reactor '+str(i+1))
137
138 # approximate the switching rate as the slope of the linear part in the
139 # log plot
140 # reactor 1:
141 # from 12.6 h to 17 h
142 # reactor 2:
143 # from 14.5 h to 18 h
144 # reactor 3:

```

```

145     # from 14.2 h to 18 h
146
147     indexes=np.logical_and(percent_positive.index>=H_time_threshold[i,0],
148                             percent_positive.index<=H_time_threshold[i,1])
149
150     # add the relevant parts in bold red to the plot
151     plt.plot(percent_positive.index[indexes],
152              percent_positive['percent'][indexes],
153              'r', linewidth=2)
154
155     # fit a line to the data
156     # use a linear regression
157     slope, intercept, r_value, p_value, std_err = linregress(
158         percent_positive.index[indexes],
159         np.log(percent_positive['percent'][indexes]))
160
161     print('Reactor '+str(i+1)+':')
162     print('slope: '+str(slope))
163     print('intercept: '+str(intercept))
164     print('r_value: '+str(r_value))
165
166     # plot the linear regression
167     plt.plot(percent_positive.index[indexes],
168              np.exp(intercept)*np.exp(slope*percent_positive.index[indexes]),
169              'k--')
170
171
172     # save the parameters
173     parameters.loc[i, 'H']=slope
174     parameters.loc[i, 'H_r_value']=r_value
175     parameters.loc[i, 'H_r2']=r_value**2
176
177
178     ## measure the maximum growth rate
179
180     # set_data[0][0].data → OD
181     fig, ax=plt.subplots()
182
183     ax.plot(set_data[0][0].data['t'], set_data[0][0].data['m'],'r', label='OD')
184
185     ax.legend()
186
187     # x ax limits 0 to 25
188     plt.xlim([0,25])
189

```

```

190     # estimate growth rate between 6 and 12 hours, 12.5 h, and 12.2 h
191     # by fitting an exponential to the OD data
192
193     limits=[12,12.5,12.2]
194     indexes=np.logical_and(set_data[0][0].data['t']>=6,
195                           set_data[0][0].data['t']<=limits[i])
196
197     # fit an exponential to the data
198     # use a linear regression
199     slope, intercept, r_value, p_value, std_err = linregress(
200         set_data[0][0].data['t'][indexes],
201         np.log(set_data[0][0].data['m'][indexes]))
202
203     print("-----")
204     print("Growth rate reactor "+str(i+1)+":")
205     print('slope: '+str(slope))
206     print('intercept: '+str(intercept))
207     print('r_value: '+str(r_value))
208
209     # plot the linear regression
210     plt.plot(set_data[0][0].data['t'][indexes],
211             np.exp(intercept)*np.exp(slope*set_data[0][0].data['t'][indexes]),
212             'k--')
213
214     # save the parameters
215     parameters.loc[i, 'growth_rate']=slope
216     parameters.loc[i, 'growth_rate_r_value']=r_value
217     parameters.loc[i, 'growth_rate_r2']=r_value**2
218
219 parameters.to_csv('parameters.csv')
220
221
222 plt.show()

```

Python script for data representation:

```

1  from mbiomas import fca
2  from datetime import datetime
3  import matplotlib.pyplot as plt
4  import pandas as pd
5  import numpy as np
6
7  print('Reading Files in Folders')
8  Channel = "FITC-A"
9  # FCF,FCD,FCN=fca.read.multititl("batch/Reactor1",fields=['FSC-A','FSC-H',Channel],

```

```

10         clean=True,zeros=True,negs=True,doublets=True,delete=True)
11 FCF,FCD,FCN=fca.read.multititle("continuous/Reactor1",fields=['FSC-A','FSC-H',Channel],
12         clean=True,zeros=True,negs=True,doublets=True,delete=True)
13 Stats = fca.statistics.calculate(FCF,FCD,fields=[Channel])
14
15 #print(FCF.data["Time"])
16
17 #####
18 #Plotting
19
20 #Time Scatter
21 fig,ax=plt.subplots(figsize = (5, 2.5))
22 TS=fca.plot.Plot(FCF)
23 TS.time_scatter(ax, FCF, FCD, y_label=Channel,y_log_lim=[1,6], y_res=150)
24 plt.tick_params(axis='both', which='both', labels=8)
25 plt.plot(Stats.data[Channel]["Time"], 10**Stats.data[Channel]["Q75"],
26         color = "red",linewidth=1,
27         marker='o', markerfacecolor='red', markersize=3, alpha = 0.4)
28 plt.xlabel('Time [h]', fontsize=10) # Adjust the fontsize as needed
29 plt.ylabel('FL1-A [a.u.]', fontsize=10) # Adjust the fontsize as needed
30
31 # adjust depending on the experiment:
32 # plt.text(1,12000,s='D = 0.32',fontsize=14)
33 plt.vlines(x = 15.497,ymin=0,ymax=10**8,color="b")
34 # plt.text(16,12000,s='D = 0.13',fontsize=14)
35 plt.vlines(x = 43.73,ymin=0,ymax=10**8,color="b")
36 # plt.text(44,12000,s='D = 0.1',fontsize=14)
37 # plt.vlines(x = 91.25,ymin=0,ymax=10**8,color="b")
38 #plt.text(92,12000,s='D = 0.17',fontsize=14)
39
40 plt.xlim([0,90])
41 plt.ylim([10,20000])
42
43 fig.savefig('FL_Duplicate.png',
44         bbox_inches='tight',dpi=700)

```

Additionally, the function used to show the fluxes of cells was:

```

1 def flux_time_scatter(ax, FCFiles,FCData, y_label='FSC-A',
2         list=None,
3         y_log_lim=None, y_res=500,
4         y_scale='log',
5         y_lin_lim=None,
6         cmap='seismic',
7         **kwargs):

```

```

8      """
9
10     Plots a scatter-plot of the fluxes of cell values across time
11     for channel `index`.
12     This is actually a 3D histogram, thus some of the options
13         are related to bins of the histogram.
14
15     Parameters
16     -----
17     ax : matplotlib.axes.Axes
18         The matplotlib Axe object on which to plot the data.
19     FCFiles : mbiomas.core.mFCFiles
20         The mFCFiles object containing the data.
21     FCData : mbiomas.core.mFCData
22         The mFCData object containing the data.
23     y_label : str, optional
24         The label of the y-axis.
25     list : list, optional
26         The list of graph from which to take space to make the colorbar,
27         by default all of them.
28     y_log_lim=[2,8] : array-like, optional
29         A size 2 1-dimension array/array-like with the minimum and maximum
30         of the y axis, in log10. Default is based on data.
31         Can only be set if `y_lin_lim` is not set.
32     y_res : int, optional
33         The number of bins on the y-axis. Default is 500.
34     y_scale='log' : str, optional
35         The scale of the y-axis. Default is 'log'. Can be 'log' for a
36         log10 scale, 'symlog' for a symlog scale, or 'linear' for a linear
37         scale.
38     y_lin_lim : array-like, optional
39         A size 2 1-dimension array/array-like with the minimum and maximum
40         of the y axis, in linear space. Default is based on data.
41         Can only be set if `y_log_lim` is not set.
42     **kwargs : dict, optional
43         Additional arguments are passed to the colorbar method.
44     """
45
46     # kwargs are passed to the colorbar
47
48     if list is None:
49         list=[ax]
50
51     if y_log_lim is not None and y_lin_lim is not None:
52         raise ValueError("Both logspace_lim and normalspace_lim are set")

```

```

53 elif y_log_lim is not None:
54     y_lin_lim=[10**y_log_lim[0],10**y_log_lim[1]]
55
56
57 times=FCFiles.data["Time"].to_numpy()
58
59 # need to end up with a 1D array of all the data in every
60 # FCData.data[i][y_label]
61 # memory allocation takes time, first, iterate to get the total
62 # size of the array, then second iteration to allocate it
63 # and fill it
64
65 data_size=0
66 for dataset in FCData.data.values():
67     data_size+=len(dataset)
68
69 t=np.empty(data_size,dtype=np.float16)
70 d=np.empty(data_size,dtype=np.float32)
71 w=np.empty(data_size,dtype=np.float32)
72
73 i=0 # current index in full array
74 for j, dataset in FCData.data.items():
75     current_size=len(dataset)
76     t[i:i+current_size]=times[j]
77     d[i:i+current_size]=dataset[y_label]
78     w[i:i+current_size]=1.0/current_size
79     i+=current_size
80
81 if y_lin_lim is None:
82     y_lin_lim=[np.min(d),np.max(d)]
83
84 # logbins = np.logspace(y_log_lim[0],y_log_lim[1],y_res)
85 if y_scale=='log':
86
87     if y_lin_lim[0]<=0:
88         print("-----")
89         print("-----")
90         print("WARNING:")
91         print("You were attempting to represent data down to")
92         print(y_lin_lim[0])
93         print("in log scale. That is impossible, use symlog scale if")
94         print("you really want that. In the meantime, we set the lower")
95         print("values shown to 0.1")
96         print("-----")
97         print("-----")

```

```

98         y_lin_lim[0]=0.1
99         y_bins=np.logspace(np.log10(y_lin_lim[0]),
100                             np.log10(y_lin_lim[1]),y_res)
101     elif y_scale=='linear':
102         y_bins=np.linspace(y_lin_lim[0],y_lin_lim[1],y_res)
103     elif y_scale=='symlog':
104         y_bins=np.sinh(np.linspace(np.arcsinh(y_lin_lim[0]),
105                                     np.arcsinh(y_lin_lim[1]),y_res))
106     else:
107         raise ValueError("Value for scale not recognized")
108
109     timebins= (times[0:-1]+times[1:])/2
110     # add the two extreme bins for the time:
111     timebins=np.concatenate([[2*times[0]-times[1]], timebins,
112                             [1.5*times[-1]-0.5*times[-2]]])
113
114
115     hist2d=np.histogram2d(t, d,
116                           bins=[timebins.astype('float64'), y_bins.astype('float64')],
117                           weights=w)
118     diff= 5000*np.diff(hist2d[0], axis=0)
119     # diff=5000*np.gradient(hist2d[0])[0]
120     diff=gaussian_filter(diff, sigma=1.2)
121     # diff[diff<0]=0
122
123     c=cm.ScalarMappable(cmap=cmap)
124     vmax=np.max(np.abs(diff))
125     # hb=ax.pcolormesh(timebins.astype('float64'), y_bins.astype('float64'),
126     hb=ax.pcolormesh(times, y_bins.astype('float64'),
127                       diff.T,
128                       vmin=-vmax, vmax=vmax,
129                       # norm='asinh',
130                       # norm='linear',
131                       norm='symlog',
132                       cmap=c.get_cmap())
133
134     cb=plt.colorbar(cm.ScalarMappable(norm=hb.norm, cmap=hb.cmap),ax=list,
135                     **kwargs)

```

5 Dataset legends

The supplementary data includes:

- The triplicate batch fermentation data used to compute most parameters.

- A duplicate experiment with multiple dilution rates to assess the instability region.
- A confirmation experiment to show the instability at 0.1 h^{-1} . Note that samples between 34 and 39 h are unreliable due to a technical sampling error.

References

- [1] Juan Andres Martinez et al. *Automated Adjustment of Metabolic Niches Enables the Control of Natural and Engineered Microbial Co-Cultures*. May 2024. DOI: 10.1101/2024.05.14.594082. (Visited on 09/18/2024).