

# Dynamic modeling of enzyme controlled metabolic networks using a receding time horizon

Henning Lindhorst \* Alexandra-M. Reimers \*\*  
Steffen Waldherr \*\*\*

\* *Institute for Automation Engineering, Otto-von-Guericke-Universität  
Magdeburg; (e-mail: henning.lindhorst@ovgu.de).*

\*\* *Department of Mathematics and Computer Science, Freie  
Universität Berlin; (email: alexandra.reimers@fu-berlin.de)*

\*\*\* *KU Leuven, Department of Chemical Engineering; (e-mail:  
steffen.waldherr@kuleuven.be)*

---

**Abstract:** Microorganisms have developed complex regulatory features controlling their reaction and internal adaptation to changing environments. When modeling these organisms we usually do not have full understanding of the regulation and rely on substituting it with an optimization problem using a biologically reasonable objective function. The resulting constraint-based methods like the Flux Balance Analysis (FBA) and Resource Balance Analysis (RBA) have proven to be powerful tools to predict growth rates, by-products, and pathway usage for fixed environments. In this work, we focus on the dynamic enzyme-cost Flux Balance Analysis (deFBA), which models the environment, biomass products, and their composition dynamically and contains reaction rate constraints based on enzyme capacity. We extend the original deFBA formalism to include storage molecules and biomass-related maintenance costs. Furthermore, we present a novel usage of the receding prediction horizon as used in Model Predictive Control (MPC) in the deFBA framework, which we call the short-term deFBA (sdeFBA). This way we eliminate some mathematical artifacts arising from the formulation as an optimization problem and gain access to new applications in MPC schemes. A major contribution of this paper is also a systematic approach for choosing the prediction horizon and identifying conditions to ensure solutions grow exponentially. We showcase the effects of using the sdeFBA with different horizons through a numerical example. © International Federation of Automatic Control.

*Keywords:* model predictive control, metabolic engineering, gene expression, linear optimization

---

## 1. INTRODUCTION

Microorganisms encounter a vast array of environmental conditions and have developed complex regulatory mechanisms to cope with them. While a lot of research is done to investigate this, most regulatory features are still unknown. An effective alternative approach is the substitution of the regulation with an optimization problem as originally done with the Flux Balance Analysis (FBA) in (Varma and Palsson, 1994b). This method models the organism as a metabolic network in steady-state and maximizes a single biomass flux. This approach led to a family of methods focusing on different aspects.

Initial steps towards dynamic models with the ability to react to changing environments were made with the dynamic FBA (Mahadevan et al., 2002). But this method still lacks a connection between reaction rates and the enzyme levels necessary to realize them. The first optimization method to take this into account is the Resource Balance

Analysis (RBA) (Goelzer et al., 2011). In this method the growth rate of the cell is optimized to a fixed medium composition while enzymatic flux constraints limit uptake and metabolic reaction rates. The combination of these enzymatic constraints and a dynamic approach resulted in the dynamic enzyme-cost Flux Balance Analysis (deFBA) presented in (Waldherr et al., 2015). The deFBA predicts all reaction rates and enzymatic levels for given nutrient dynamics on a chosen time frame. An application of the deFBA to a genome scale model can be found in (Reimers et al., 2017a).

During a recent study (Waldherr and Lindhorst, 2017) we learned that the fixed end-time in the deFBA can lead to artificial solutions usually not observed in the modeled organisms. Furthermore, we plan to use deFBA inside a model predictive controller to maximize certain biomass products by manipulation of the medium composition. Thus, we present in this work the *short-term deFBA* (sdeFBA), which combines the deFBA with the idea of a receding prediction horizon. This also allows us to solve problems with large end-times piece-wise and in some cases reduces the computational cost for the simulation.

---

\* H.L. and A.-M.R. are funded by ERANET for Systems Biology ERASysApp, project ROBUSTYEAST, BMBF grant IDs 031L0017A and 031L0017B.

## 2. DYNAMIC ENZYME-COST FLUX BALANCE ANALYSIS

### 2.1 Constructing the optimization problem

In this section we present the basics of the deFBA and showcase the extensions of our current formulation in comparison to the original one (Waldherr et al., 2015). At the heart of deFBA models lies a metabolic reaction network consisting of  $n$  biochemical species and  $m$  reactions converting the species into each other. We further classify the species depending on their physical location and their biological function as either

- *external species*  $Y \in \mathbb{R}_{\geq 0}^{n_y}$  outside of the cell (carbon sources, oxygen, etc.),
- *metabolic species*  $X \in \mathbb{R}_{\geq 0}^{n_x}$  which are intermediates and intracellular products of the metabolism (amino acids, ATP, etc.),
- *storage species*  $C \in \mathbb{R}_{\geq 0}^{n_c}$  which are allowed to accumulate in the model (glycogen, starch, etc.),
- *macromolecules*  $P \in \mathbb{R}_{\geq 0}^{n_p}$  representing biomass components (enzymes, cell walls, DNA, etc.),

with  $n = n_y + n_x + n_p + n_c$ . We measure all species in molar amounts, e.g.,  $[X] = \text{mol}$ .

The macromolecules  $P$  represent the complete reproductive machinery of the organism and can be further divided into a catalytic part, enabling reactions via enzymes and taking care of reproduction via the ribosome, and a non-catalytic part, like cell walls, DNA, etc. To keep the notation simple we address both kinds with  $P$ . Most organisms use some of the available nutrients to create an energy storage, which can be used to survive phases of starvation, e.g. production of starch during day for consumption at night. The storage species  $C$  can either be some macromolecules or simply metabolites allowed to accumulate.

The deFBA assumes the network maximizes biomass accumulation over time. Thus, we assign the accumulating species  $C, P$  their molecular weights  $w_i$ ,  $[w_i] = \text{g/mol}$  and define the *total biomass*  $B$  as

$$B(t) = w_C^T C(t) + w_P^T P(t), \quad (1)$$

depending on the time  $t$ ,  $[t] = \text{h}$ . As recent studies have shown (Waldherr and Lindhorst, 2017) the inclusion of non-catalytic biomass in the objective may lead to unexpected results if these species are very "cheap" to produce in comparison to their weights  $w_C$ . Thus, we additionally define the *objective biomass*  $B_o$  via the *objective weights*  $b_i$ , which in most cases coincide with the molecular weights, but can be set to zero if necessary

$$B_o(t) = b_C^T C(t) + b_P^T P(t). \quad (2)$$

The reactions  $R$  between the species are subdivided into the following types:

- *exchange reactions*  $v_Y \in \mathbb{R}^{m_y}$  exchanging matter with the outside,
- *metabolic reactions*  $v_X \in \mathbb{R}^{m_x}$  transforming metabolites into one another,
- *storage reactions*  $v_C \in \mathbb{R}^{m_c}$  converting metabolites in storage and vice versa,

- *biomass reactions*  $v_P \in \mathbb{R}^{m_p}$  producing macromolecules,

with  $m = m_y + m_x + m_c + m_p$ . We write shortly  $v = (v_Y^T, v_X^T, v_C^T, v_P^T)^T$ ,  $[v] = \text{mol/h}$ . The dynamics of the species are then given by the *stoichiometric matrix*  $S \in \mathbb{R}^{n,m}$

$$\begin{aligned} \frac{d}{dt} \begin{pmatrix} Y(t) \\ X(t) \\ C(t) \\ P(t) \end{pmatrix} &= \begin{pmatrix} S_{Y,Y} & 0 & 0 & 0 \\ S_{X,Y} & S_{X,X} & S_{X,C} & S_{X,P} \\ 0 & 0 & S_{C,C} & 0 \\ 0 & 0 & 0 & S_{P,P} \end{pmatrix} \begin{pmatrix} v_Y(t) \\ v_X(t) \\ v_C(t) \\ v_P(t) \end{pmatrix} \\ &= \begin{pmatrix} S_Y \\ S_X \\ S_C \\ S_P \end{pmatrix} \begin{pmatrix} v_Y(t) \\ v_X(t) \\ v_C(t) \\ v_P(t) \end{pmatrix} = Sv(t), \end{aligned} \quad (3)$$

with the submatrices  $S_{I,J} \in \mathbb{R}^{n_I, m_J}$ ,  $I, J \in \{Y, X, C, P\}$ . Following (Waldherr et al., 2015), the metabolism is modelled to operate in quasi steady-state. This translates to the constraint

$$\frac{d}{dt} X(t) = S_X v(t) = 0, \quad \forall t \geq 0. \quad (4)$$

The enzymatic biomass catalyzes the reactions in the network and the maximal rates are determined by the reaction-specific *catalytic constants* (or turnover numbers)  $k_{\text{cat}, \pm j}$ ,  $j \in \{1, \dots, m\}$ ,  $[k_{\text{cat}, \pm j}] = \text{h}^{-1}$  and the amount of the respective enzyme  $P_i$ . We differentiate between the *forward value*  $k_{\text{cat}, +j}$  and the *backward value*  $k_{\text{cat}, -j}$ .

The bounds for the reactions rates are given by

$$-v_j \leq k_{\text{cat}, -j} P_i, \quad v_j \leq k_{\text{cat}, +j} P_i. \quad (5)$$

Furthermore, some enzymes are capable of catalyzing multiple reactions, which we describe with the sets

$$\text{cat}(P_i) = \{v_j \mid P_i \text{ catalyzes } v_j\}. \quad (6)$$

The corresponding constraint with respect to reversibility of the reactions then reads

$$\sum_{v_j \in \text{cat}(P_i)} \left| \frac{v_j(t)}{k_{\text{cat}, \pm j}} \right| \leq P_i(t), \quad \forall t \geq 0. \quad (7)$$

We call the matrix form the *enzyme capacity constraint*

$$H_c v(t) \leq H_e P(t), \quad \forall t \geq 0, \quad (8)$$

with the filter matrix  $H_e$ . For more detail on the construction of these matrices see (Waldherr et al., 2015). The constraint (8) is the central constraint in deFBA as it limits growth. In regular FBA the growth rate is constrained by biomass independent constraints

$$v_{\min} \leq v(t) \leq v_{\max} \quad (9)$$

derived from measured reaction rates. Because all reactions can reach arbitrarily large rates given enough enzyme is present (cf. (8)), we make the following assumption.

*Assumption 1.* The biomass independent constraints (9) are only used to define the reversibility of the reactions with  $v_{\min}, v_{\max} \in \{\pm\infty, 0\}^m$ .

Any organism needs structural macromolecules to keep working, e.g., the cell wall separating it from the outside. We express this necessity by enforcing certain fractions  $\psi_s \in [0, 1)$  of the total biomass  $B(t)$  to be made of structural components, e.g., for a structural macromolecule  $P_s$

$$\psi_s B(t) \leq P_s(t), \quad \forall t \geq 0. \quad (10)$$

The extension of (10) to the network level can be expressed by collecting the individual constraints into the *biomass composition matrix*  $H_b$  with

$$H_b \begin{pmatrix} C(t) \\ P(t) \end{pmatrix} \leq 0, \quad (11)$$

where the rows of  $H_b$  are derived from (10). We call (11) the *biomass composition constraint*. Furthermore, we can enforce specific reaction rates

$$v_m(t) \geq \phi_m B(t), \quad \forall t \geq 0 \quad (12)$$

with the *maintenance coefficient*  $\phi_m \in [0, 1]$  to model maintenance reactions scaling with biomass, e.g., re-synthesis of lipids. Hence, we call (13) the *maintenance constraint*

$$v(t) \geq H_m \begin{pmatrix} C(t) \\ P(t) \end{pmatrix}, \quad (13)$$

with the rows of  $H_m$  corresponding to  $\phi_m(w_C^T, w_P^T)$  (cf. (12)). To construct the full deFBA problem, we introduce an *end-time*  $t_{\text{end}} > 0$  and define the objective function as accumulation of the objective biomass (2) as

$$\begin{aligned} \max_{v(t)} \int_0^{t_{\text{end}}} B_o(t) dt \\ \text{s.t. (4), (8), (9), (11), (13); } \forall t \in [0, t_{\text{end}}]. \end{aligned} \quad (14)$$

This dynamic optimization problem can be solved by discretization with a collocation method. The result is a linear program (LP) for which efficient, specialized solvers are available. With respect to the computational and numerical details of solving such problems, we refer the reader to (Waldherr et al., 2015), and to (Reimers et al., 2017a) for a large scale example. We provide an implementation of the deFBA model class in Python 2.7<sup>1</sup>, which imports/exports models using libSBML (Bornstein et al., 2008) and the resource allocation modeling (RAM) annotations (Lindhorst et al., 2017). A step-by-step guide for the generation of deFBA models is described in (Reimers et al., 2017b).

## 2.2 Important growth modes

There are multiple reasons to discard the large end-time  $t_{\text{end}}$  in favor of a shorter prediction horizon  $0 < t_p \ll t_{\text{end}}$  and implement an iterative version of the original problem (14). Foremost, the deFBA can produce *linear phases*, defined as

$$dB_o(t)/dt = \lambda, \quad (15)$$

with the constant linear growth rate  $\lambda \geq 0$ . These phases can occur if some macromolecules are very "cheap" in comparison to others. The model uses all resources to solely produce the cheap molecules, regardless of their utility. These phases can either be observed when using very small end-times or as mean to top off the objective value near nutrient depletion or the end-time  $t_{\text{end}}$  (Waldherr and Lindhorst, 2017). We regard the linear phases as mathematical artifacts of the optimization method itself as we do not know of biological examples for this behavior. Thus, one goal of the prediction horizon is to eliminate these linear arcs in the solutions.

Another important growth mode, called a *balanced phase*, is defined by

$$dB_o(t)/dt = \mu_{\text{bal}} B_o(t), \quad (16)$$

with the constant exponential growth rate  $\mu_{\text{bal}} \in \mathbb{R}_{\geq 0}$  depending on nutrient availability and the current biomass composition. In these phases the composition of the biomass stays fixed as it is already optimal for the environment. A dynamic solution generated by the deFBA typically consists of a series of balanced growth phases and the transitions between these.

## 3. SHORT-TERM DEFBA

### 3.1 Implementing the receding time horizon

The implementation of the receding prediction horizon  $t_p$  is straightforward. We split the time interval  $[0, t_{\text{end}}]$  into intervals  $[t_k, t_{k+1}]$  using the time grid  $\Delta_t(t_c) = \{t_k = kt_c \mid k \in \mathbb{N}\}$  defined by the *iteration time*  $t_c \in (0, t_p)$ . Then we replace the original deFBA problem (14) with a series of small problems we call the *short-term deFBA* (sdeFBA). With given values  $Y^{t_k}, C^{t_k}, P^{t_k}$ , these read

$$\max_{v(t)} \int_{t_k}^{t_k+t_p} B_o(t) dt \quad (17a)$$

$$\text{s.t. } \forall t \in [t_k, t_k + t_p] \quad (17b)$$

$$\frac{d}{dt} \begin{pmatrix} Y(t) \\ C(t) \\ P(t) \end{pmatrix} = \begin{pmatrix} S_Y \\ S_C \\ S_P \end{pmatrix} v(t) \quad (17c)$$

$$S_X v(t) = 0 \quad (17d)$$

$$H_c v(t) \leq H_c P(t) \quad (17e)$$

$$H_b \begin{pmatrix} C(t) \\ P(t) \end{pmatrix} \leq 0 \quad (17f)$$

$$v(t) \geq H_m \begin{pmatrix} C(t) \\ P(t) \end{pmatrix} \quad (17g)$$

$$v_{\min} \leq v(t) \leq v_{\max} \quad (17h)$$

$$Y(t), C(t), P(t) \geq 0 \quad (17i)$$

$$Y(t_k) = Y^{t_k}, C(t_k) = C^{t_k}, P(t_k) = P^{t_k}. \quad (17j)$$

For given initial values  $Y_0, C_0, P_0$ , we solve the problem iteratively starting at time zero and connecting the iterations via (17j). The solution trajectories  $Y^*(t), C^*(t), P^*(t), v^*(t)$ ,  $0 \leq t \leq t_{\text{end}}$  are generated by appending the calculated slices over the iteration time  $[t_k, t_k + t_c]$  after each iteration.

### 3.2 Choosing the prediction horizon

We already stated that the native growth mode for metabolic networks is exponential growth, while linear phases are undesired. Our analysis in (Waldherr and Lindhorst, 2017) shows, that linear solutions can arise on very short time scales as exponential solutions need a longer time horizon to outperform them. Hence, we must ensure to choose the prediction horizon  $t_p$  large enough such that linear solutions become sub-optimal. At the same time we want to keep  $t_p$  as small as possible to minimize computational cost. We suggest to determine the prediction horizon by comparison of a strict upper bound on linear growth with an arbitrary balanced growth phase. The idea is sketched in Figure 1. This way we ensure the existence of at least piece-wise exponential solutions on the time horizon  $t_p$ . This calculation is dependent on two sets of variables; the nutrients available and the initial biomass composition  $P_{\text{init}}, C_{\text{init}}$  at time zero (or  $t_k$  in the sdeFBA).

<sup>1</sup> <https://bitbucket.org/hlindhor/defba-python-package>

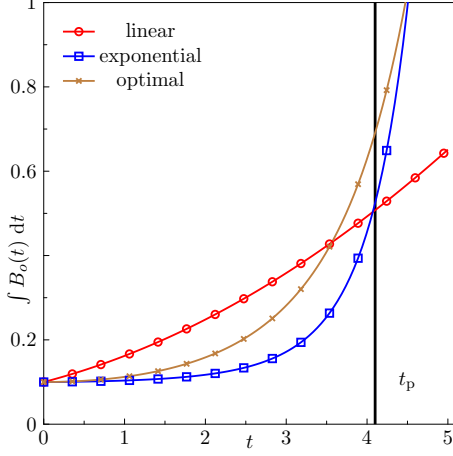


Fig. 1. Illustration for choosing the prediction horizon. Upper bound on linear growth shown in red ( $\circ$ ), balanced growth in blue ( $\square$ ), and optimal solution in brown ( $\times$ ).

To eliminate the influence of nutrient availability in this first investigation we make the following assumption.

*Assumption 2.* All external components  $Y$  are limitlessly available.

We define the *initial objective biomass* as

$$B_{\text{init}} = b_C^T C_{\text{init}} + b_P^T P_{\text{init}}. \quad (18)$$

First we identify a strict upper bound on linear growth dependent on the initial biomass amount by constructing an optimization problem inspired by the regular FBA (Varma and Palsson, 1994a). We assume a linear growth phase  $dB_o(t)/dt = \lambda$  and maximize the linear growth rate

$$\lambda = b_C^T S_C v_{\text{lin}} + b_P^T S_P v_{\text{lin}}. \quad (19)$$

Following Assumption 2, we ignore the nutrient dynamics. The optimization problem is then constructed as

$$\lambda_s(B_{\text{init}}) = \max_{v_{\text{lin}}, P_{\text{lin}}, C_{\text{lin}}} b_C^T S_C v_{\text{lin}} + b_P^T S_P v_{\text{lin}} \quad (20a)$$

$$\text{s.t. } S_X v_{\text{lin}} = 0 \quad (20b)$$

$$H_c v_{\text{lin}} - H_e P_{\text{lin}} \leq 0 \quad (20c)$$

$$H_b \begin{pmatrix} C_{\text{lin}} \\ P_{\text{lin}} \end{pmatrix} \leq 0 \quad (20d)$$

$$w_C^T C_{\text{lin}} + w_P^T P_{\text{lin}} = B_{\text{init}} \quad (20e)$$

$$v_{\text{lin}} \geq H_m \begin{pmatrix} C_{\text{lin}} \\ P_{\text{lin}} \end{pmatrix} \quad (20f)$$

$$v_{\text{min}} \leq v_{\text{lin}} \leq v_{\text{max}}, \quad (20g)$$

with (20e) fixing the initial amount of biomass to  $B_{\text{init}}$ . The value of the *specific growth rate*  $\lambda_s(B_{\text{init}})$  is dependent on the amount of biomass. Instead we use the *regularized rate*

$$\lambda_r = \frac{\lambda_s(B_{\text{init}})}{B_{\text{init}}}. \quad (21)$$

For easier reading we omit the dependency of  $\lambda_s$  on the biomass.

We construct the linear solution as

$$P(t) = P_{\text{lin}} + S_P v_{\text{lin}} t, \quad C(t) = C_{\text{lin}} + S_C v_{\text{lin}} t. \quad (22)$$

This solution is usually not feasible for the original sdeFBA problem (17) with  $t_p > 0$  as violations of (17f) and (17g) are to be expected with increase in biomass over time.

As next step, we identify a balanced growth phase to use as a lower bound for optimal exponential growth by optimizing the static growth rate  $\mu \geq 0$  at  $t = 0$

$$\frac{d}{dt} \begin{pmatrix} C_{\text{init}} \\ P_{\text{init}} \end{pmatrix} = \mu \begin{pmatrix} C_{\text{init}} \\ P_{\text{init}} \end{pmatrix}. \quad (23)$$

The resulting optimization problem reads

$$\mu_{\text{bal}} = \max_{v_{\text{bal}}} \mu \quad (24a)$$

$$\text{s.t. } \mu \begin{pmatrix} C_{\text{init}} \\ P_{\text{init}} \end{pmatrix} = \begin{pmatrix} S_C \\ S_P \end{pmatrix} v_{\text{bal}} \quad (24b)$$

$$S_X v_{\text{bal}} = 0 \quad (24c)$$

$$H_c v_{\text{bal}} - H_e P_{\text{init}} \leq 0 \quad (24d)$$

$$v_{\text{bal}} \geq H_m \begin{pmatrix} C_{\text{init}} \\ P_{\text{init}} \end{pmatrix} \quad (24e)$$

$$v_{\text{min}} \leq v_{\text{bal}} \leq v_{\text{max}}. \quad (24f)$$

The trajectories of the balanced growth phase are derived by solving the initial value problem

$$\frac{d}{dt} \begin{pmatrix} C(t) \\ P(t) \end{pmatrix} = \mu_{\text{bal}} \begin{pmatrix} C(t) \\ P(t) \end{pmatrix}, \quad (25)$$

with  $C(0) = C_{\text{init}}$ ,  $P(0) = P_{\text{init}}$ . These trajectories are realized by the rates  $v(t) = v_{\text{bal}} e^{\mu_{\text{bal}} t}$  and represent a feasible solution to (17), if Assumption 2 holds and the initial values are feasible

$$H_b \begin{pmatrix} C_{\text{init}} \\ P_{\text{init}} \end{pmatrix} \leq 0. \quad (26)$$

We can calculate a suitable time  $t_p$ , by comparing the the balanced solution (25) to the linear one (22). The integral of the biomass curve for (25) is derived as

$$\begin{aligned} IB_{\text{bal}}(t, \mu_{\text{bal}}, B_{\text{init}}) &= \int_0^t b_C^T C(t) + b_P^T P(t) dt \\ &= \mu_{\text{bal}}^{-1} B_{\text{init}} (e^{\mu_{\text{bal}} t} - 1) \end{aligned} \quad (27)$$

and the corresponding integral for the linear case is

$$IB_{\text{lin}}(t, \lambda_r, B_{\text{init}}) = \int_0^t B_{\text{lin}}(t) dt \quad (28)$$

$$= \frac{\lambda_r B_{\text{init}}}{2} t^2 + B_{\text{init}} t. \quad (29)$$

We calculate the prediction horizon by solving

$$IB_{\text{lin}}(t_p, \lambda_r, B_{\text{init}}) - IB_{\text{bal}}(t_p, \mu_{\text{bal}}, B_{\text{init}}) = 0 \quad (30)$$

for  $t_p$ . By looking at the slopes of the biomass curves at time zero, we can deduce that this  $t_p > 0$  only exists if, and only if,  $\lambda_r > \mu_{\text{bal}}$ . Otherwise, the model does not tend to the linear solution and we can chose  $t_p$  arbitrarily.

*Assumption 3.* The linear growth rate is larger than the balanced growth rate  $\lambda_r > \mu_{\text{bal}}$ .

An optimal solution of (17) on  $[0, t_p]$  can only produce an objective value equal or larger than  $IB_{\text{bal}}(t_p)$ , otherwise it would contradict the optimality principle. Hence, we conclude that this optimal solution must contain a superlinear (typically exponential) arc as shown in Figure 1.

*Remark 1.* Calculating  $t_p$  is strongly dependent on the initial biomass  $P_{\text{init}}, C_{\text{init}}$ . Hence, during an sdeFBA run the prediction horizon should be recalculated after each iteration step.

### 3.3 Choosing the iteration time

To keep the computational cost of a sdeFBA run as small as possible we choose the iteration time  $t_c$  as large as possible, such that the solution is still of exponential form. Hence, we show that each solution of (17) starts with an exponential phase. For this we assume a solution starting with a linear phase

$$B_{\text{mix}}(t) = \begin{cases} B_{\text{init}}\lambda_r t + B_{\text{init}} & 0 \leq t \leq t_s \\ B_{\text{init}}(\lambda_r t_s + e^{\mu_{\text{bal}}(t-t_s)}) & t_s < t \leq t_p, \end{cases} \quad (31)$$

with the switching time  $t_s$  and assume Assumption 3 holds. This solution is constructed on the assumption that the linear growth phase does not benefit the autocatalytic capabilities of the system.

*Theorem 1.* If Assumption 3 holds, any optimal solution curve  $B_{\text{mix}}$  (31) consists only of a single linear phase with  $t_s = t_p$ .

**Proof.** We identify the optimal switching time by solving

$$\max_{t_s} \int_0^{t_p} B_{\text{mix}}(t) dt \quad (32)$$

analytically by finding local extrema via the first order derivative with respect to  $t_s$

$$0 = \frac{d}{dt_s} \int_0^{t_p} B_{\text{mix}}(t) dt = B_{\text{init}}(\lambda_r(t_p - t_s) + 1 - e^{\mu_{\text{bal}}(t_p - t_s)}), \quad (33)$$

with the obvious zero  $\bar{t}_s = t_p$ . Evaluating the second derivative at this point gives

$$\frac{d^2}{dt_s^2} \int_0^{t_p} B_{\text{mix}}(t) dt \Big|_{\bar{t}_s} = B_{\text{init}}(\mu_{\text{bal}} - \lambda_r) < 0, \quad (34)$$

with the last inequality following Assumption 3. Hence,  $\bar{t}_s$  is a local maximum and any solution of the  $B_{\text{mix}}$  form does not include an exponential arc. For the sake of completeness, we must also mention that there exists another zero of (33)  $\bar{t}_{s,2} \in [0, t_p]$ , which cannot be given in closed form. But, due to continuity and the intermediate value theorem,  $\bar{t}_{s,2}$  is a local minimum of (32).  $\square$

As we have chosen  $t_p$  such that the balanced growth solution (27) outgrows the maximal linear one, we know that there exists a time frame  $[0, t_c]$  on which the solution of (17) must at least grow exponentially. Thus, we assume the following form for the solution

$$B_{\text{opt}}(t) = \begin{cases} B_{\text{init}}e^{\mu_{\text{bal}}t}, & 0 \leq t \leq t_s, \\ B_{\text{init}}e^{\mu_{\text{bal}}t_s}(\lambda_r(t - t_s) + 1), & t_s < t \leq t_p, \end{cases} \quad (35)$$

with  $B_{\text{init}}\lambda_re^{\mu_{\text{bal}}t_s} = \lambda_s(B_{\text{opt}}(t_s))$ . We want to choose  $t_c$  such that no linear phase occurs in the final solution of the sdeFBA. Otherwise, we can get faulty solutions as shown in the next section.

*Theorem 2.* If Assumption 3 holds, an optimal solution  $B_{\text{opt}}$  (35) of the sdeFBA (17) is growing exponentially on the time frame  $[0, t_c]$ , with

$$0 < t_c < t_p - 2 \left( \frac{1}{\mu_{\text{bal}}} - \frac{1}{\lambda_r} \right). \quad (36)$$

**Proof.** As in the previous proof, we identify the optimal switching time  $t_s$  by solving the optimization problem

$$\max_{t_s} \int_0^{t_p} B_{\text{opt}}(t) dt. \quad (37)$$

The zeros of the first order derivative are given by

$$\frac{d}{dt_s} \int_0^{t_p} B_{\text{opt}}(t) dt = 0 \quad (38)$$

$$\Rightarrow \hat{t}_{s,1} = t_p - 2 \left( \frac{1}{\mu_{\text{bal}}} - \frac{1}{\lambda_r} \right), \quad \hat{t}_{s,2} = t_p. \quad (39)$$

The second-order derivative evaluated at these points is

$$\begin{aligned} \frac{d^2}{dt_s^2} \int_0^{t_p} B_{\text{opt}}(t) dt \Big|_{\hat{t}_{s,1}} &= (\mu_{\text{bal}} - \lambda_r)B_{\text{init}}e^{\mu_{\text{bal}}\hat{t}_{s,1}} < 0, \\ \frac{d^2}{dt_s^2} \int_0^{t_p} B_{\text{opt}}(t) dt \Big|_{\hat{t}_{s,2}} &= (\lambda_r - \mu_{\text{bal}})B_{\text{init}}e^{\mu_{\text{bal}}\hat{t}_{s,2}} > 0. \end{aligned} \quad (40)$$

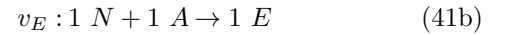
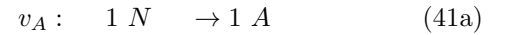
Hence,  $\hat{t}_{s,1}$  maximizes (37) and the solution is of exponential form until  $\hat{t}_{s,1}$ .  $\square$

We strongly advise to choose the iteration time smaller than given by (36) to compensate for numerical errors. Otherwise, we might see solutions mixing linear and exponential phases as shown in Figure 2 (C).

Please note that  $t_c$  is also dependent on the prediction horizon  $t_p$  and the initial biomass composition  $B_{\text{init}}$ . So it should be recalculated together with  $t_p$  after each iteration (cf. Remark 1).

## 4. NUMERICAL EXAMPLE

We present a simple model, analyzed in detail in (Waldherr and Lindhorst, 2017), to give the reader an idea about the impact of end-times, prediction horizons, and iteration times on the quality of the solution. In this minimal example the organism can invest nutrients in either its' auto catalytic capabilities by investing in enzymes or it can produce non-catalytic components yielding a better nutrients-to-biomass ratio. The three irreversible reactions of the network are



The external nutrient  $N$  represents a collection of components necessary for growth, such as carbon, nitrogen, etc. Further processed components made from these nutrients are collected as the internal metabolite  $A$ . We differentiate the macromolecules into the group of enzymes  $E$ , collecting the whole enzymatic machinery needed for growth, and non-enzymatic macromolecules  $M$ . These can be interpreted as storage components such as lipids, starch, or glycogen.

Assuming unlimited nutrients, we would expect a biological system to work exclusively in the exponential phase and produce no storage  $M$  at all. But the defBA model (14) may generate a solution containing linear phases depending on the system parameters and the end-time.

In this work we are only interested in the effects of the time variables and fix the system parameters to the values shown in Table 1. The numerical results using these values were all generated with our Python defBA package<sup>2</sup> using a discretization step size  $d = 0.1$  h and the initial values  $E(0) = M(0) = 0.1$  mol.

<sup>2</sup> Available at [bitbucket.org/hlindhor/defba-python-package](https://bitbucket.org/hlindhor/defba-python-package)

Table 1. Values used in the numerical example

$b_M$ [ $\frac{g}{mol}$ ]	$b_E$ [ $\frac{g}{mol}$ ]	$k_A$ [ $h^{-1}$ ]	$k_M$ [ $h^{-1}$ ]	$k_E$ [ $h^{-1}$ ]
15	10	1.5	2	1

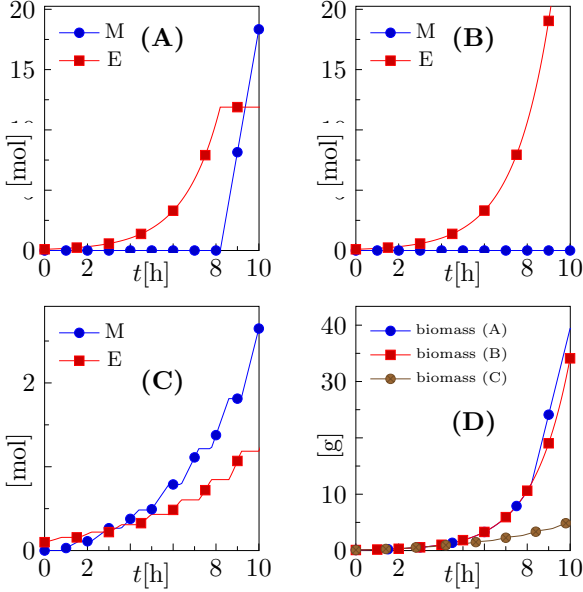


Fig. 2. (A) deFBA solution  $t_{\text{end}} = 10$  h. (B) sdeFBA solution  $t_p = 3.25$  h,  $t_c = 1.45$  h. (C) sdeFBA solution  $t_p = 2.5$  h,  $t_c = 1.5$  h. (D) biomass comparison of methods.

Following (Waldherr and Lindhorst, 2017), we can derive the necessary condition for a single linear phase to be the optimal solution as

$$t_{\text{lin}} \leq \frac{2(k_M b_M - k_E b_E)}{b_M k_M k_E} \approx 1.45 \text{ h.} \quad (42)$$

Choosing any  $t_{\text{end}} > t_{\text{lin}}$  results in a mixed trajectory starting with an exponential phase and ending with a linear one. This behavior can be observed in Figure 2 (A). A purely exponential solution is not attainable with the deFBA as any solution ends in a linear phase producing only  $M$  to top off the objective.

But we can use the short-term deFBA to generate an exponential solution. Using the idea from Section 3 we calculate the initial prediction horizon as  $t_p \approx 3.25$  h and the iteration time as  $t_c \approx 1.45$  h. The sdeFBA generates a purely exponential solution as shown in Figure 2 (B). While this is a more reasonable solution from a biological view, the objective value for this solution is slightly smaller than the one obtained by the deFBA (cf. Figure 2 (D)).

Figure 2 (C) shows a sdeFBA solution using a prediction horizon  $t_p = 2.5$  h and an iteration time  $t_c = 1.5$  h. While this  $t_p$  is capable of producing an exponential phase in each iteration the chosen iteration time is way too large. Hence, we see a solution in which exponential growth and linear phases take turns on each iteration slice. This is neither optimal nor observed in nature.

## 5. CONCLUSION

While our presentation of the sdeFBA focuses on the quality of the solution, this method provides further ad-

vantages in comparison to the original deFBA. Foremost, we can replace the fixed time frame  $[0, t_{\text{end}}]$  in the original deFBA (14) with a variable one dependent on the network's state. As example, the deFBA is not designed to handle starvation scenarios and the optimization problem may become infeasible if the nutrients deplete. But in the sdeFBA we can simply stop iterating once the nutrients deplete or another chosen threshold is reached. Of course, this also means we can update state variables or dynamics while setting up the next iteration. So we can use the sdeFBA as predictor in an online model predictive controller, which maximizes, e.g., some biomass component by changing the nutrient composition.

Lastly, the sdeFBA can be a way to solve large scale deFBA problems on large time-scales more efficiently. The problem lies in the linear programs constructed by the deFBA, whose states can vary several orders of magnitude due to exponential growth phases. This leads to ill-posed problems, which take very long to solve even with sophisticated commercial solvers. By breaking the problem into smaller pieces via the sdeFBA we can reduce the computational time.

## REFERENCES

- Bornstein, B.J., Keating, S.M., Jouraku, A., and Hucka, M. (2008). LibSBML: An API library for SBML. *Bioinformatics*, 24(6), 880–881.
- Goelzer, A., Fromion, V., and Scorletti, G. (2011). Cell design in bacteria as a convex optimization problem. *Automatica*, 47(6), 1210–1218.
- Lindhorst, H., Reimers, A.-M., Bockmayr, A., and Waldherr, S. (2017). RAM: An annotation standard for SBML Level 3. doi:10.15490/fairdomhub.1.sop.304.6.
- Mahadevan, R., Edwards, J.S., and Doyle, F.J. (2002). Dynamic flux balance analysis of diauxic growth in *Escherichia coli*. *Biophysical Journal*, 83(3), 1331–1340.
- Reimers, A.-M., Knoop, H., Bockmayr, A., and Steuer, R. (2017a). Cellular trade-offs and optimal resource allocation during cyanobacterial diurnal growth. *Proceedings of the National Academy of Sciences*, 114(31), E6457–E6465. doi:10.1073/pnas.1617508114.
- Reimers, A.-M., Lindhorst, H., and Waldherr, S. (2017b). A protocol for generating and exchanging (genome-scale) metabolic resource allocation models. *Metabolites*, 7(3)(47). doi:10.3390/metabo7030047.
- Varma, A. and Palsson, B.O. (1994a). Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type *Escherichia coli* w3110. *Applied and Environmental Microbiology*, 60(10), 3724–3731.
- Varma, A. and Palsson, B. (1994b). Metabolic Flux Balancing: Basic Concepts, Scientific and Practical Use. *Nature Biotechnology*, 12(10), 994–998. doi:10.1038/nbt1094-994.
- Waldherr, S. and Lindhorst, H. (2017). Optimality in cellular storage via the Pontryagin Maximum Principle. *Preprints of the 20th IFAC World Congress*, 20, 10305–10311.
- Waldherr, S., Oyarzún, D.A., and Bockmayr, A. (2015). Dynamic optimization of metabolic networks coupled with gene expression. *Journal of Theoretical Biology*, 365, 469–485.