Modeling Enzyme Controlled Metabolic Networks in Rapidly Changing Environments by Robust Optimization*

Henning Lindhorst[1], Sergio Lucia[2], Rolf Findeisen[3] and Steffen Waldherr[4]

Abstract—**Constraint based methods, such as the Flux Balance Analysis, are widely used to model cellular growth processes without relying on knowledge of regulatory features. Regulation is instead substituted by an optimization problem to maximize a biological objective such as biomass accumulation. A recent extension to these methods is called dynamic enzyme-cost Flux Balance Analysis (deFBA). This fully dynamic modeling method allows to predict optimal enzyme levels and reaction fluxes under changing environmental conditions. However, this method was designed for well defined deterministic settings in which dynamics of the environment are exactly known. In this work, we present a theoretical framework called the robust deFBA which extends the deFBA to handle uncertainty in nutrient availability. We achieve this by combining deFBA with multi-stage Model Predictive Control which explicitly captures the evolution of uncertainty by a scenario tree. The resulting method is capable of predicting robust optimal gene expression levels for rapidly changing environments. We apply these algorithms to a model of the core metabolic process in bacteria under alternating oxygen availability.**

Index Terms—**Computational methods; Metabolic systems; Robust control**

I. Introduction

MICROORGANISMS are constantly encountering variations in their environments, like changes in temperature, pH-value, or pressure. But most importantly the availability of nutrients can change rapidly. Therefore, microbes have developed complex regulatory mechanisms to control their gene expression and adapt on the fly or prepare for changes preemptively. While some regulation mechanisms, e.g. the lac operon, are very well understood, for the majority we only have a vague understanding. This led to the development of constrained based models which, instead of including regulatory features in the model, formulate an optimization problem to describe the effects of regulation. The most prominent approach utilizing this concept is the *Flux Balance Analysis* (FBA) [1], which selects the reaction fluxes in quasi steady-state, such that a biomass producing flux is maximized.

This approach has led to a number of different extensions like the *iterative FBA* presented in [2] which allows to capture changes in the environment. However, it lacks the interplay between enzymes and associated reaction rates. This interplay is considered in the *Resource Balance Analysis* (RBA) [3], which optimizes the growth rate in a fixed environment while including enzymatic flux constraints to limit uptake and metabolic fluxes. The RBA allows to predict flux rates and the enzyme distribution under a quasi-stationary assumption.

The next step in constrained based modeling, called the *dynamic enzyme-cost FBA* (deFBA) [4], can predict all reaction rates and the enzyme levels dynamically over a time course. It is especially useful to investigate adaptation during nutrient changes, e.g. the switch from aerobic to anaerobic growth conditions or the depletion of a nutrient source. However, it is quite complex to generate deFBA models as they rely on a largely increased number of parameters in comparison to original FBA models. In [5] it is outlined how to generate deFBA models starting with a gene annotated metabolic model. Furthermore, solving genome-scale deFBA models on large time horizons is computationally costly. To compensate this we developed the *short-term deFBA* (sdeFBA) [6], which utilizes the idea of a receding prediction horizon to segment the problem into smaller and easier to solve problems. We deliver a basic introduction to sdeFBA in Section II.

Any of these methods rely on the assumption that cells adapt such that their growth rate is maximized at any given time or over a specified time under some nominal environmental conditions. However, as experiments have shown, this approach can not explain all metabolic regulation strategies, such as leaky gene expression [7]. We suggest to model nutrient availability as an uncertainty(see Section III) and use a robust receding horizon optimization approach to cope with the uncertainty. This allows to adapt the objective used in the model to an *overall fitness* as suggested in [8]. We call the method realizing this idea *robust deFBA* (rdeFBA) and present it in Section IV.

Finally, we showcase the differences between classic sdeFBA and the robust deFBA in a numerical example switching between aerobic and anaerobic growth in Section V.

II. Short-term deFBA

Looking at the deFBA as an extension of the regular FBA, the following description contains additionally reactions for macromolecule production and some new constants, which

*H.L. is funded by ERANET for Systems Biology ERASysApp, project ROBUSTYEAST, BMBF grant ID 031L0017A and Research Center Dynamic Systems: Systems Engineering, Otto-von-Guericke-University Magdeburg.

[1] Otto-von-Guericke-Universität Magdeburg, Institute for Automation Engineering henning.lindhorst@ovgu.de

[2] TU Berlin, Department of Electrical Engineering and Computer Science sergio.lucia@tu-berlin.de

[3] Otto-von-Guericke-Universität Magdeburg, Laboratory for Systems Theory and Automatic Control rolf.findeisen@ovgu.de

[4] KU Leuven, Department of Chemical Engineering steffen.waldherr@kuleuven.be

need to be collected from external sources. For a detailed guide on how to build deFBA models we refer to [9].

The short-term deFBA [6] is a discretized version of the original deFBA utilizing a receding prediction horizon. As already mentioned this can lead to a decrease in computation time for large-scale systems on long time scales.

At the heart of sdeFBA models lies a metabolic reaction network consisting of n biochemical species and m reactions converting the species into one another. We 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.),
- *macromolecules* $P \in \mathbb{R}_{\geq 0}^{n_p}$ representing biomass components (enzymes, cell walls, ribosome, etc.).

The macromolecules P represent the 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 as P. We measure all species in their molar amounts. This ensures that we end up with a linear optimization problem.

SdeFBA assumes that the network maximizes biomass accumulation over time. Thus, we assign each macromolecule their molecular weight w_i, with a unit of g/mol. The *total biomass* B is computed by

$$B(t) = w^T P(t), \tag{1}$$

depending on the time t. The *growth rate* is given as

$$\mu(t) = \frac{1}{B(t)} \frac{dB(t)}{dt}. \tag{2}$$

The reactions are introduced as independent variables and their rates are determined via optimization. We distinguish between the following reaction 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,
- *biomass reactions* $v_P \in \mathbb{R}^{m_p}$ producing macromolecules,

with $m = m_y + m_x + m_p$. We write shortly $v = (v_Y^T, v_X^T, v_P^T)^T$ and assign the unit mol/h.

As with all FBA methods, we assume the metabolism operates on a shorter time-scale than the environmental changes and biomass production. Therefore, it is not necessary to model the metabolic species X explicitly. We use instead a quasi steady-state constraint on the rates in the form

$$S_X v(t) = 0, \ \forall t \geq 0, \tag{3}$$

with the *stoichiometric matrix* $S_X \in \mathbb{R}^{n_X, m}$ (cf. [4]).

We collect the other species in the vector $Z = (Y^T, P^T)^T \in \mathbb{R}^{n_y + n_p}$ and define their dynamics as

$$\dot{Z}(t) = \begin{bmatrix} S_Y \\ S_P \end{bmatrix} v(t), \tag{4}$$

with the stoichiometric matrices $S_Y \in \mathbb{R}^{n_y, m}$ and $S_P \in \mathbb{R}^{n_p, m}$.

The maximal achievable rates for the reactions are determined by the reaction-specific *catalytic constants* (or *turnover numbers*) $k_{\text{cat},\pm j}$, $j \in \{1, \dots, m\}$ and the amount of the catalyzing enzyme P_i. We differentiate between the *forward value* $k_{\text{cat},+j}$ and the *backward value* $k_{\text{cat},-j}$.

The reaction rate bound for an enzyme that catalyzes a single reaction is given in the form

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

Some enzymes are capable of catalyzing multiple reactions, which we describe by the sets

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

The corresponding constraint with respect to reversibility of the reactions becomes

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

We collect all enzymatic constraint in the *enzyme capacity constraint* as

$$H_c v(t) \leq H_e P(t), \ \forall t \geq 0, \tag{8}$$

with the filter matrix H_e. For more detail on the construction of these matrices see [4]. Equation (8) describes the central sdeFBA constraint as it limits the reaction rates and as a consequence the growth rate (2) as well.

In regular FBA, rates are constrained by biomass independent relations

$$v_{\min} \leq v(t) \leq v_{\max}, \ \forall t \geq 0, \tag{9}$$

derived from measured reaction rates. Due to our formulation in molar amounts all reactions can reach arbitrarily large rates given enough enzyme is present (cf. (8)). Hence, we use (9) to define only the reversibility of 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

$$\psi_s B(t) \leq P_s(t), \ \forall t \geq 0. \tag{10}$$

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

$$H_b P(t) \leq 0, \ \forall t \geq 0, \tag{11}$$

where the rows of H_b are derived from (10). We call (11) the *biomass composition constraint*.

Furthermore, we can enforce specific reaction rates

$$v_a(t) \geq \phi_a B(t), \ \forall t \geq 0 \tag{12}$$

with the *maintenance coefficient* $\phi_a \in \mathbb{R}_{\geq 0}$ to model maintenance reactions scaling with biomass. These are typically ATP sinks, which are not modeled individually. The matrix form is called *maintenance constraint*

$$H_f v(t) \geq H_a P(t), \tag{13}$$

with the rows of H_a corresponding to $\phi_a w^T$ (cf. (12)) and a suitable filter matrix H_f.

Short-term deFBA is based on an optimization over a receding time horizon. This *prediction horizon* $t_\mathrm{p} \in \mathbb{R}_{>0}$ describes how far in the future the optimizer plans ahead using a step size $h \in \mathbb{R}_{>0}$ for time discretization $\Delta t(h) = \{t_k = kh \mid k \in \mathbb{N}\}$. The horizon t_p can be very influential on the simulation results. A systematic way to ensure specific properties can be found in [6]. We discretize all species and flux trajectories on Δt leading to the approximation $Z_k \approx Z(t = kh)$ and $v_k \approx v(t = kh)$.

We choose the accumulation of biomass (1) as the objective to ensure both fast and high growth while avoiding certain types of non-uniqueness in the flux solutions (cf. [4]). Summarizing a single problem at time $t = t_g = gh$ on the receding horizon is constructed as

$$\max_{(v_k)_{k\in\mathcal{K}}} \sum_{k\in\mathcal{K}} B_k \tag{14a}$$

$$\text{with given} \quad Z_g = Z(t = t_g) \tag{14b}$$

$$\text{subject to:} \; Z_{k+1} = Z_k + hSv_k \tag{14c}$$

$$S_X v_k = 0 \tag{14d}$$

$$H_a Z_{k+1} - H_f v_k \leq 0 \tag{14e}$$

$$H_b Z_{k+1} \leq 0 \tag{14f}$$

$$H_c v_k - H_e Z_k \leq 0 \tag{14g}$$

$$Z_k \geq 0 \tag{14h}$$

$$v_\mathrm{min} \leq v_k \leq v_\mathrm{max}, \; \forall k \in \mathcal{K}, \tag{14i}$$

with the index set

$$\mathcal{K} = \{k \in \{g + ih\} \mid i = 0, 1, \dots, p-1; \; hp = t_\mathrm{p}\}. \tag{15}$$

For easier reading we substituted the dynamics (4) with a simple forward Euler scheme. In real applications, we strongly advise to use a higher order collocation schemes, e.g., Radau collocation. For given initial values Y_0, P_0, we solve the problem (14) iteratively starting at time zero and connecting the iterations via (14b). The discrete solution trajectories Z^*, v^* are generated by appending the calculated values Z_{g+1}, v_g for the time slice $[t_g, t_{g+1}]$ after each iteration.

Each problem of the form (14) is a linear program (LP), for which efficient, specialized solvers exist. With respect to the computational and numerical details of solving such problems, we refer the reader to [4], and [5] for a genome scale example. We provide an implementation of the sdeFBA model class in form of the `Python-deFBA package` [10], which imports/exports models using `libSBML` [11] and the *resource allocation modeling annotations* (RAM) [12]. This package also contains methods to simulate regular deFBA models and the robust deFBA as presented in Section IV.

III. Constructing uncertain environments

Fluctuations in the availability of nutrients are encountered by all microorganisms; either due to simple depletion of resources or by unpredictable external effects. We call a cell population *environmentally robust*, if it does not adapt perfectly to each set of conditions but instead uses some resources to be prepared for a nutritional shift [13]. Our goal is to approximate this robust behavior and the decision process behind it with the sdeFBA. The first step is to find a suitable description for uncertain availability of nutrients.

Solving large-scale sdeFBA problems can be quite demanding in itself. Thus, the formulation of the uncertain environment should be as simple as possible. The foremost relevant basis for any decision made by the cell is whether the nutrient is available or might become available in the near future. So we decided to limit any uncertainty d regarding the environment to binary values $d \in \{0, 1\}$; each corresponding to either possible presence or absence of the nutrient.

To model the decision process inside the cell, we have to distinguish two settings. (i) A nutrient N_a is currently available and might run out. (ii) A nutrient N_u is currently unavailable but might become available again.

Let us first consider case (i). To model the possible depletion of N_a we add a new constraint to (14)

$$d_a v_{\mathrm{in},a}(t - t_g) \leq c_\mathrm{limit}(t - t_g), \; \forall t \geq t_g \tag{16}$$

with $v_{\mathrm{in},a}$ being the only uptake flux for N_a and the current time t_g. The direction of $v_{\mathrm{in},a}$ is chosen such that positive (negative) rates correspond to uptake (secretion). This constraint is only active for $d_a = 1$ and can simulate the depletion of N_a by forcing $v_{\mathrm{in},a}$ to zero. But setting c_limit simply to zero can quickly lead to an infeasible problem, e.g., if enzymes necessary for the uptake of an alternative nutrient can not be produced without N_a. This is especially true if maintenance reactions are constantly consuming energy. To avoid this problem we give the cell some time to adapt to the upcoming change in the environment by choosing c_limit as

$$c_\mathrm{limit}(t) = \begin{cases} (1 - \frac{t}{t_\mathrm{b}})v_{\mathrm{in,max}}, & t \leq t_\mathrm{b} \\ 0, & t_\mathrm{b} \leq t, \end{cases} \tag{17}$$

with the *preparation time* $t_\mathrm{b} \in \mathbb{R}_{\geq 0}$ and the current maximal uptake rate $v_\mathrm{in,max}$. The preparation time determines how fast the cell needs to adapt to changes in the environment. We suggest using half the prediction horizon as preparation time to put enough stress on the network. As we want to use (16) on a receding time horizon we have to choose a suitable value for $v_\mathrm{in,max}$ in each iteration. On the one hand we want to enable the cell to improve their uptake rate for N_a for a short time span as this be might beneficial for reacting to the upcoming shift. On the other hand we assume that shifts can happen abruptly and want to put enough stress on the cell to prepare as soon as possible. We suggest choosing the value depending on the calculated uptake rates $v_{\mathrm{in},g-1}$ from the previous time step t_{g-1}

$$v_\mathrm{in,max} = v_{\mathrm{in},g-1} e^{\mu_\mathrm{max} h}, \tag{18}$$

with the maximal growth rate μ_max determined by an RBA. This allows for exponential increase in the uptake rate, but still puts enough nutritional stress on the cell. In summary, we model either the original problem for $d_a = 0$ or enforce an adaption process to another nutrient with $d_a = 1$.

In the same fashion we handle case (ii), in which a nutrient source might become available (again). This case is much simpler as we do not need to worry about the need to prepare for the switch as sticking to the available resources is still viable. Hence, we use a simplified version of constraint (16)

$$(1 - d_u) v_{\text{in},u}(t - t_g) \leq 0, \ \forall t \geq t_g. \tag{19}$$

The value $d_u = 0$ represents again the nominal situation: N_u is depleted and no uptake can take place. To allow uptake for $d_u = 1$ can be realized by increasing the amount of N_u artificially. But it is difficult to predict the achievable uptake rates $v_{\text{in},u}$ and setting N_u too low could influence the decision process as it may not be viable to invest for the cell. Hence, we decouple the decision process to metabolize N_u from its present amount by not balancing it. Assuming $v_{\text{in},u}$ is limited by an enzymatic constraint (8), we can realize this by either setting the amount of N_u to infinity or by deleting the entries of the stoichiometric matrix S with respect to N_u. This means $v_{\text{in},u}$ is only constrained by the amount of transporter enzymes present for $d_u = 1$.

This formulation of the uncertain environment allows to easily construct sdeFBA models by fixing values for d. But the solutions of these models are still only valid for the chosen d. To generate robust solutions, we will need to consider all possible value combinations at the same time.

IV. Robust deFBA

A suitable way to generate robust predictions for this kind of uncertain environments is *multi-stage Model Predictive Control* (mMPC) [14], [15]. MMPC is a *scenario-based optimization* technique, in which the scenarios are generated from sampled parameter values. In our case, we sample the values for d.

Assuming our model contains n_d uncertain nutrients, we can distinguish 2^{n_d} realizations $d^j \in \{0,1\}^{n_d}, j \in 0, \ldots, 2^{n_d} - 1$. From here on, we will use the upper index to distinguish between scenarios and keep the lower index for time steps. We write shortly

$$\mathcal{J} = \{0, \ldots, 2^{n_d} - 1\}. \tag{20}$$

To uniquely identify a scenario we use a binary mapping of the values to their resp. vector d^j, e.g., $n_d = 3$

$$j = 6 \sim \text{ binary } 110 \rightarrow d^6 = [1, 1, 0]. \tag{21}$$

This way d^0 always corresponds to the scenario in which the current environmental condition does not change.

The idea for handling these scenarios all at once is sketched in Figure 1. Each branch (or scenario) of this tree corresponds to a value of d^j, in this case for three uncertain environments. This tree is in fact a simplified representation of a more complex structure branching at every node, representing a sequence of different d^j values over time. We stick to the simple tree as it keeps the computational complexity low and still produces robust results (cf. [14]). To optimize over all scenarios at once, we couple them by two means.

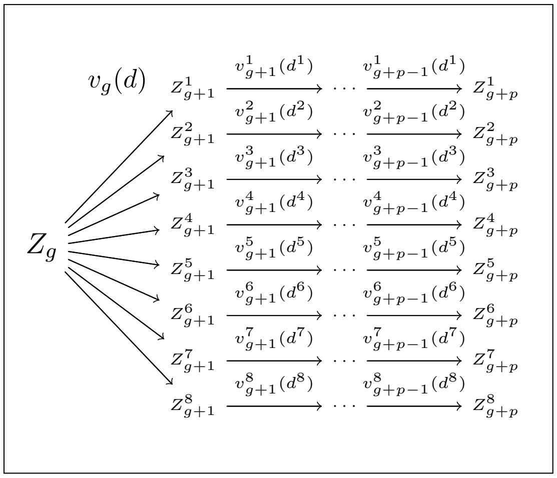
$v_g(d)$
Z_g
Z^1_{g+1} $v^1_{g+1}(d^1)$ … $v^1_{g+p-1}(d^1)$ Z^1_{g+p}
Z^2_{g+1} $v^2_{g+1}(d^2)$ … $v^2_{g+p-1}(d^2)$ Z^2_{g+p}
Z^3_{g+1} $v^3_{g+1}(d^3)$ … $v^3_{g+p-1}(d^3)$ Z^3_{g+p}
Z^4_{g+1} $v^4_{g+1}(d^4)$ … $v^4_{g+p-1}(d^4)$ Z^4_{g+p}
Z^5_{g+1} $v^5_{g+1}(d^5)$ … $v^5_{g+p-1}(d^5)$ Z^5_{g+p}
Z^6_{g+1} $v^6_{g+1}(d^6)$ … $v^6_{g+p-1}(d^6)$ Z^6_{g+p}
Z^7_{g+1} $v^7_{g+1}(d^7)$ … $v^7_{g+p-1}(d^7)$ Z^7_{g+p}
Z^8_{g+1} $v^8_{g+1}(d^8)$ … $v^8_{g+p-1}(d^8)$ Z^8_{g+p}

Fig. 1. Representation of the scenario tree with uncertain vector $d^j \in \{0,1\}^3$. Rates $v_g(d)$ must be viable in all scenarios.

Firstly, the optimization utilizes a shared objective function

$$J = \sum_{j \in \mathcal{J}} o^j_w \sum_{k=g}^{g+p} B^j_k(t) \ \mathrm{d}t, \tag{22}$$

with objective weights $o^j_w \in \mathbb{R}_{\geq 0}$ and total biomass B^j for each scenario. Usually, the weights o^j_w are all set to one, but they can be used to implement additional information on the likelihood of certain scenarios. When used in this context, one can directly translate probability to the weights by using the constraint

$$\sum_{j \in \mathcal{J}} o^j_w = 1. \tag{23}$$

The second connection between the scenarios is called the *nonanticipativity constraint* [16]. As suggested by Figure 1 it enforces that the first set of rates v_g must be feasible in all scenarios

$$v^j_g = v^i_g = v_g, \ \forall i, j \in \mathcal{J}. \tag{24}$$

This is done for two reasons. On the modeling side, constraint (24) expresses the fact, that the cell must decide what strategy to implement for the time being. On the technical side, we still want to calculate a robust solution on the receding time horizon and need a unique set of rates v_g for the solution v^*_k, see Section II.

Adding the known constraints from the sdeFBA and new ones from the uncertain environments, we end up with the *robust deFBA* at time $t_g = gh$ as

$$\max_{(v^j_k), j \in \mathcal{J}, \ k \in \mathcal{K}} \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} B^j_k \tag{25a}$$

$$\text{with given} \quad Z^j_g = Z(t = t_g) \tag{25b}$$

$$\text{subject to:} \ Z^j_{k+1} = Z^j_k + hSv^j_k \tag{25c}$$

$$S_X v^j_k = 0 \tag{25d}$$

$$H_a Z^j_{k+1} - H_f v^j_k \leq 0 \tag{25e}$$

$$H_b Z^j_{k+1} \leq 0 \tag{25f}$$

TABLE I
EXCHANGE AND METABOLIC REACTIONS, CATALYTIC CONSTANTS k_{cat}.

Reaction	Enzyme	$k_{\text{cat}}/\text{min}^{-1}$
Exchange reactions		
Carbon → A	$\mathrm{T_C}$	3000
$\mathrm{F_{ext}} \to \mathrm{F}$	$\mathrm{T_F}$	3000
$\mathrm{H} \to \mathrm{H_{Ext}}$	$\mathrm{T_H}$	3000
$\mathrm{O2_{ext}} \to \mathrm{O2}$	S	1000
$\mathrm{D} \leftrightarrow \mathrm{D_{Ext}}$	S	1000
$\mathrm{E} \leftrightarrow \mathrm{E_{Ext}}$	S	1000
Metabolic reactions		
A + ATP → B	$\mathrm{E_B}$	1800
B → C + 2 ATP + 2 NADH	$\mathrm{E_C}$	1800
C ↔ 2ATP + 3D	$\mathrm{E_D}$	1800
C + 4NADH ↔ 3E	$\mathrm{E_E}$	1800
B → F	$\mathrm{E_F}$	1800
C → G	$\mathrm{E_G}$	1800
G + ATP + 2NADH ↔ H	$\mathrm{E_H}$	1800
G → 0.8C + 2NADH	$\mathrm{E_N}$	1800
O2 + NADH → ATP	$\mathrm{E_T}$	1800

TABLE II
BIOMASS PRODUCING REACTIONS CATALYZED BY RIBOSOME R.

Biomass reactions	b/g	$k_{\text{cat}}/\text{min}^{-1}$	P_0/mol
400H+1600 ATP → $\mathrm{T_C}$	4	2.5	6.3039E-3
400H+1600 ATP → $\mathrm{T_F}$	4	2.5	0.0
400H+1600ATP→ $\mathrm{T_H}$	4	2.5	0.0
500H+2000ATP→ $\mathrm{E_B}$	5	2	10.5065E-3
500H+2000ATP→ $\mathrm{E_C}$	5	2	10.0354E-3
1000H+4000ATP→ $\mathrm{E_D}$	10	1	2.8901E-3
1000H+4000ATP→ $\mathrm{E_E}$	10	1	3.6913E-3
2000H+8000ATP→ $\mathrm{E_F}$	20	0.5	4.7100E-4
500H+2000ATP→ $\mathrm{E_G}$	5	2	2.6528E-3
4000H+16000ATP→ $\mathrm{E_H}$	40	0.25	2.6528E-3
500H+2000ATP→ $\mathrm{E_N}$	5	2	0.0
500H+2000ATP→ $\mathrm{E_T}$	5	2	0.0
4500H+1500C+21000ATP→ R	60	0.2	5.4538E-3
250H+250C+250F+1500ATP→ S	7.5	3	46.6976E-3

$$H_c v_k^j - H_e Z_k^j \leq 0 \tag{25g}$$

$$H_d^j v_k^j \leq C_{\text{limit},k}^j \tag{25h}$$

$$Z_k^j \geq 0 \tag{25i}$$

$$v_g^j = v_g^i = v_g, \ \forall i, j \in \mathcal{J} \tag{25j}$$

$$v_{\min} \leq v_k^j \leq v_{\max}, \forall (k, j) \in (\mathcal{K} \times \mathcal{J}), \tag{25k}$$

with (25h) being the combined matrix form of (16), (19) and $\mathcal{K}$ as defined in (15).

As direct consequence of (24) and (25c) we obtain a unique state vector at time $g+1$ for all scenarios

$$Z_{g+1}^j = Z_{g+1}^i = Z_{g+1}, \ \forall i, j \in \mathcal{J} \tag{26}$$

and can construct a unique solution Z^*. Therefore, we can use the robust deFBA as purely predictive method without relying on external state updates.

Furthermore, the nonanticipativity constraint is sufficient to ensure the solutions v^*, Z^* are robust against changes in the environment; meaning optimal and feasible under the considered changes in the environment. We know v_g must obey all constraints in all scenarios (cf. (24)). Because the actual nutrient situation d^0 is always included as a scenario, in all cases of an absent nutrient the intake is forced to zero. If we have chosen $v_{\text{in,max}}$ (17) large enough, no artificial growth limitations are implemented by the robust optimization.

V. EXAMPLE

To showcase the effects of using robust deFBA in comparison to the short-term version we apply them both to an academic example described in Table I and Table II. It represents the core carbon network as found in all cells. We took this model from [4], which describes the problem in more detail.

The most relevant model features are the active transport of a carbon source via the enzyme $\mathrm{T_C}$ and the subsequent aerobic or anaerobic glycolysis. The production of transporters T, enzymes E, and structural component S are all handled by the ribosome R. Production of S is enforced via the biomass composition constraint (11), such that at least 35% of the total biomass are of type S at any time. The system secretes fermentation products D and E, which can also be consumed as nutrients in aerobic conditions. We use the biomass vector b as given by Table II.

For the robust analysis we are investigating the difference between aerobic and anaerobic growth. Hence, we build two different models corresponding to the presence of $\mathrm{O2_{ext}}$ in the medium as described in the previous section. In aerobic growth phases the oxygen is not explicitly modeled and can be consumed in arbitrary amounts. Furthermore, we assume that Carbon is always and limitlessly available. This corresponds to substituting Carbon with the empty set in the first exchange reaction.

The experimental setup is structured as following: The initial biomass composition for rdeFBA and sdeFBA are determined via an RBA with deFBA constraints for an anaerobic environment only containing Carbon with $B(0) = 1\text{g}$. The exact values are given in Table II. We simulate both cultures for half an hour in an aerobic setting using a step size of 2min and a prediction horizon of 34min. In the robust simulation we chose the preparation time as 16min. Afterwards, we restart the simulation without oxygen using the final values for the environment $Y(30\text{min})$ and macromolecules $P(30\text{min})$ as initial values. We repeat this switch until we get stable periodic biomass compositions over a set of switches.

The results depicted in Figure 2 were created with the `Python-deFBA package` developed in house [10] and are included as a numerical example. The times in which oxygen is available are shown with a blue background. First, we can observe a periodic behavior after the first hour. This means the biomass composition pattern for 60 min to 120 min in plots (C) and (D) repeats itself afterwards.

While the non-robust solution grows faster in the first 30 min the robust solution overtakes after the first switch and increases its advantage over time (A). This can be explained by plot (C) showing the amount of enzyme $\mathrm{E_E}$ with respect to the total biomass. $\mathrm{E_E}$ is necessary for anaerobic growth. While the robust solution already starts investing again in it after 10min, the short-term solution ignores it. After the

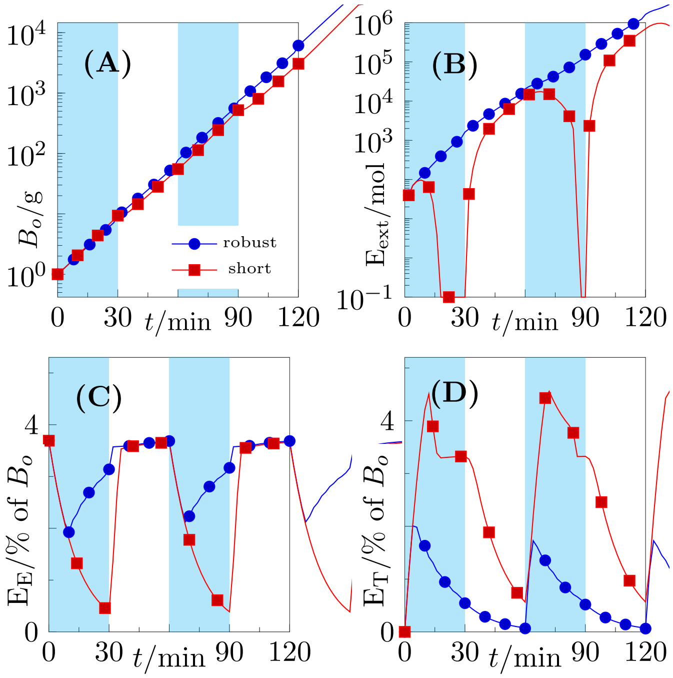

Fig. 2. Comparison of simulation results of the carbon core network. (A) shows the total amount of biomass, (B) the excretion product E_{ext}, (C) and (D) show the percentage share for E_E, resp. E_T of the total biomass.

switch to aerobic conditions the short-term solution invest as much resources as possible into E_E. We know this adaptation phase as *lag-phase* [17]. The inverse of this is depicted in plot (D), which shows enzyme E_T needed for aerobic growth. The robust solution stops investing after only 6min. This leads to the most important difference between the solutions.

The robust solution does not reabsorb the excretion product E_{ext} to boost its growth rate (B). The short-term solution on the other hand reabsorbs the waste fully. A high investment into E_T is needed to handle to excess of NADH when using E as carbon source, see the peak in plot (D) at 12 min and 72 min. With a fifty percent chance of no return on this investment, it is simply not reasonable in the robust optimization.

The next interesting result is the absence of any preparation for the possible return of oxygen availability in the robust solution during anaerobic growth phases. However, robust adaptation to aerobic growth via production of E_T is done in a single time step from 60 min to 62 min as shown in (D). At the moment, it is unclear whether this is an inherent property of the way we construct the scenarios or whether it is specific to this example.

VI. Conclusion

In this work we have presented the robust deFBA by combining the short-term deFBA with multi-stage MPC and introduced a simple way to formulate uncertainty in nutrient availability. The resulting method enables us to predict the behavior of a cell strain adapted to sudden changes in their environment. We envision that these results will help to further the understanding of metabolic regulation by comparison with regular deterministic models and experimental results.

The only observed downside of the robust deFBA is the high computational cost. Small scale models, like the carbon core example, can be evaluated with the rdeFBA on a typical office computer using our Python package [10] in mere seconds per iteration. But larger models, and especially genome-scaled ones, can be very challenging as every uncertainty doubles the problem size. Our experience so far shows a limit of roughly 750 metabolic reactions and 500 biomass reactions for the sdeFBA, see [5], and a suitable reduction of these numbers for the rdeFBA. Hence, we are currently investigating alternative problem formulation to further reduce the computational cost and make the rdeFBA viable for genome-scale models.

References

[1] A. Varma and B. Palsson, “Metabolic Flux Balancing: Basic Concepts, Scientific and Practical Use,” *Nature Biotechnology*, vol. 12, no. 10, pp. 994–998, 1994.

[2] R. Mahadevan, J. S. Edwards, and F. J. Doyle, “Dynamic flux balance analysis of diauxic growth in Escherichia coli,” *Biophysical Journal*, vol. 83, no. 3, pp. 1331–1340, 2002.

[3] A. Goelzer, V. Fromion, and G. Scorletti, “Cell design in bacteria as a convex optimization problem,” *Automatica*, vol. 47, no. 6, pp. 1210–1218, 2011.

[4] S. Waldherr, D. A. Oyarzún, and A. Bockmayr, “Dynamic optimization of metabolic networks coupled with gene expression,” *Journal of Theoretical Biology*, vol. 365, pp. 469–485, 2015.

[5] A.-M. Reimers, H. Knoop, A. Bockmayr, and R. Steuer, “Cellular trade-offs and optimal resource allocation during cyanobacterial diurnal growth,” *Proceedings of the National Academy of Sciences*, vol. 114, no. 31, pp. E6457–E6465, 2017.

[6] H. Lindhorst, A.-M. Reimers, and S. Waldherr, “Dynamic modeling of enzyme controlled metabolic networks using a receding time horizon,” *Preprints on 10th IFAC International Symposium on Advanced Control of Chemical Processes*, 2018.

[7] A. M. New, B. Cerulus, S. K. Govers, G. Perez-Samper, B. Zhu, S. Boogmans, J. B. Xavier, and K. J. Verstrepen, “Different levels of catabolite repression optimize growth in stable and variable environments,” *PLoS biology*, vol. 12, no. 1, p. e1001764, 2014.

[8] D. Molenaar, R. Van Berlo, D. De Ridder, and B. Teusink, “Shifts in growth strategies reflect tradeoffs in cellular economics,” *Molecular systems biology*, vol. 5, no. 1, p. 323, 2009.

[9] A.-M. Reimers, H. Lindhorst, and S. Waldherr, “A protocol for generating and exchanging (genome-scale) metabolic resource allocation models,” *Metabolites*, vol. 7(3), no. 47, 2017.

[10] H. Lindhorst, *Python-deFBA: A toolbox for simulating enzyme controlled metabolic networks*, Institute for Automation Engineering, Otto-von-Guericke-Universität, Magdeburg, Germany, 2017. [Online]. Available: bitbucket.org/hlindhor/defba-python-package

[11] B. J. Bornstein, S. M. Keating, A. Jouraku, and M. Hucka, “LibSBML: An API library for SBML,” *Bioinformatics*, vol. 24, no. 6, pp. 880–881, 2008.

[12] H. Lindhorst, A.-M. Reimers, A. Bockmayr, and S. Waldherr, “RAM: An annotation standard for SBML Level 3,” 2017, doi: 10.15490/fairdomhub.1.sop.304.6.

[13] T. F. Hansen, “The evolution of genetic architecture,” *Annu. Rev. Ecol. Evol. Syst.*, vol. 37, pp. 123–157, 2006.

[14] S. Lucia, T. Finkler, and S. Engell, “Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty,” *Journal of Process Control*, vol. 23, no. 9, pp. 1306 – 1319, 2013.

[15] D. Bernardini and A. Bemporad, “Scenario-based model predictive control of stochastic constrained linear systems,” in *Proceedings of the 48th IEEE Conference on Decision and Control*, 2009, pp. 6333–6338.

[16] T. Ding, Y. Hu, and Z. Bie, “Multi-stage stochastic programming with nonanticipativity constraints for expansion of combined power and natural gas systems,” *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 317–328, 2018.

[17] M. D. Rolfe, C. J. Rice, S. Lucchini, C. Pin, A. Thompson, A. D. Cameron, M. Alston, M. F. Stringer, R. P. Betts, J. Baranyi, *et al.*, “Lag phase is a distinct growth phase that prepares bacteria for exponential growth and involves transient metal accumulation,” *Journal of bacteriology*, vol. 194, no. 3, pp. 686–701, 2012.