Modeling metabolic networks including gene expression and uncertainties

H. Lindhorst\textsuperscript{a,}\textsuperscript{*}, S. Lucia\textsuperscript{a}, R. Findeisen\textsuperscript{a}, S. Waldherr\textsuperscript{b}

\textsuperscript{a}Otto-von-Guericke-University Magdeburg, Institute for Automation Engineering, Universitätsplatz 2, D-39112 Magdeburg, Germany

\textsuperscript{b}KU Leuven, Department of Chemical Engineering, Celestijnenlaan 200F bus 2424, 3001 Heverlee, Belgium

Abstract

Flux Balance Analysis methods are widely used to model cellular growth processes without relying on extensive information on the regulatory features of the cells. The regulation is instead substituted by an optimization problem usually aiming at maximal biomass accumulation. A recent extension to these methods called the dynamic enzyme-cost Flux Balance Analysis (deFBA) is a full dynamic modeling method allowing for the prediction of necessary enzyme levels under changing environmental conditions. However, this method relies on the assumption that we have full knowledge of all system parameters and the environmental dynamics. In this work, we present a framework to extend the deFBA by ideas from Model Predictive Control (MPC). Namely, we use the multi-stage nonlinear MPC as basic tool for the integration of uncertainties to the deFBA method. The main feature is the representation of the evolution of uncertainties by an exponentially growing scenario tree leading to significant computational effort. To counter this growth, we reduce the numerical effort by two means. First, we pick up on the idea of a receding prediction horizon and reshape the standard deFBA to the short-time deFBA (sdeFBA). This leads us, along with further simplification of the scenario tree, to the robust deFBA (rdeFBA). This framework is capable of handling the uncertainties in the model itself as well as uncertainties experienced by the modeled system. We applied these algorithms to two case-studies: a minimal enzymatic nutrient uptake network, and the abstraction of the core metabolic process in bacteria. For the robust deFBA we show the impact of possible measurement errors on the catalytic constants used in the models.

Keywords: Flux Balance Analysis, Metabolic Network, Robust Optimization, Predictive Control

1. Introduction

Bacteria are encountering a vast array of uncertainty in their natural living spaces. These are mainly changes in their environmental conditions such as temperature, nutrient sources, pH-value, etc. To cope with these uncertainties, they have developed complex regulatory interactions between sensing those environments, their metabolic state and levels of gene expression. Unfortunately, most of these interactions are either unknown or not fully understood yet. A widely used approach is the substitution of the regulation by an optimality principle. The most prominent approach utilizing this idea is the Flux Balance Analysis (FBA) \cite{1}. This method optimizes the fluxes in steady state, such that the biomass production is maximized. This simple approach has led to a number of different extensions like the iterative schemes presented in \cite{2,3}. These still static approaches can resolve feasible nutrient and biomass dynamics under metabolic constraints but are not fully dynamic as they do not resolve the problem over the whole timescale of interest. The work presented in \cite{3} also contains an algorithm, whose objective function depends on the...
dynamics over the whole time scale. While this delivers a consistent solution to the dynamic optimization problem, the model does not resolve the biomass composition fully. Enzymatic flux constraints are included in the Resource Balance Analysis (RBA) [4], which is again a static optimization tool for identification of the interaction between different environments and related optimal enzyme levels. A combination of the enzyme production with the dynamic FBA led to the dynamic enzyme-cost FBA [5], which is especially useful in investigating the optimal enzyme levels over the course of nutrient changes, e.g. the switch from aerobic to anaerobic growth conditions.

All of these modeling methods and solutions rely on nominal values for model parameters, like stoichiometry, flux bounds, or catalytic constants, and perfect knowledge of environmental dynamics. Thus, they are not directly capable of handling uncertainties in the model itself. There are some approaches for the FBA via inclusion of uncertainties in the stoichiometry of the network [6] and the steady-state assumption [7]. But these are constrained to static problems and are lacking the inclusion of enzyme levels.

To realize the utilization of robust optimization for the deFBA, we suggest the application of some ideas originating from robust model predictive control (MPC) [8]. As most MPC applications naturally include uncertainties, the MPC community is especially versatile in handling this problem. The most classic approach in this field is the min-max MPC [9]. The result of this method is a series of control inputs minimizing the objective for a worst-case realization of the uncertainties. This approach could be included in the deFBA, but for more complex networks the problem may lead to a very conservative solution or even become infeasible as shown in [10]. Another possible class of robust solvers is the tube-based MPC [11]. Here the solution of the nominal control problem is combined with a so called ancillary-controller ensuring the evolution of the real uncertain system is constrained to a predefined tube centered around the nominal trajectory. This method guarantees stability and the solution satisfies all constraints under the uncertainties. Yet, optimality of the solution is not addressed, and can therefore not be used in the biological setting, as it does not fit to the assumption of optimality. Hence, the main contribution of this paper is the adaptation of the multi-stage nonlinear MPC (NMPC) [12], [13] to metabolic networks including gene expression. The multi-stage NMPC is generally relying on the representation of uncertainties by means of a scenario tree on a finite prediction horizon. For the construction of the tree we assume the effects of all uncertainties can be modeled as discrete events, and each branching of the tree creates another set of scenarios. Hence, the complexity of the tree grows exponentially in the number of scenarios. Furthermore, the deFBA is constructed to solve one large optimal control problem over a specific time span in contrast to the relative short prediction horizon needed to keep the amount of scenarios manageable. Consequently, we first have to apply the idea of the receding prediction horizon to the deFBA to reduce the size of the scenario tree. Instead of using a large time window for the optimization as in deFBA, we propose here to use the receding horizon paradigm. That is, an optimization problem is solved repeatedly, using shorter time-windows, and thus reducing the computational complexity. We call this method the short-term deFBA (sdeFBA), which is an iterative scheme to solve the same problem class as the original deFBA. In this context, we suggest a methodical way to choose the prediction horizon for the sdeFBA such that the results are comparable between the methods. This extension already leads to various benefits. These are mainly the possibility to stop using a specified end-time for the process by checking during the run-time whether a stop condition is met, e.g. all nutrients are depleted. We can as well include new measurements after an iterative step in the simulation. Hence, the sdeFBA is generally usable as a predictor inside an MPC scheme. The next step is the application of the scenario tree to the sdeFBA, which we call the robust deFBA. In this framework we are able to handle all kinds of uncertainties that can occur inside a metabolic network. Yet, we focused on measurement errors in the catalytic constants needed for the deFBA. We consider a minimal example, showing the effects of the different methods and a larger example describing the core cellular process in bacteria.

In summary, our proposed framework is an extension of the deFBA, allowing us to include different kinds of uncertainties within the model of metabolic networks. The result follows a strict optimality principle and is guaranteed to be admissible under the influence of the constraints. Furthermore, we show in the numerical examples the effects of measurement errors on the robust predictions, as these can be qualitatively distinct from the non-robust predictions.
2. Dynamic enzyme-cost FBA

Let us quickly recap the deFBA and its mathematical description [5]. This method was developed to analyze the biomass fluxes under changing environmental conditions as well as the enzyme levels necessary to realize these.

2.1. Modeling metabolic networks

The networks we are investigating consist of a set of molecular species, which we divide into external species \( Y \in \mathbb{R}^{n_y} \) and internal species \( X \in \mathbb{R}^{n_x} \), a set of macromolecules \( P \in \mathbb{R}^{n_p} \) and the set of \( n_r \) reactions \( V \) between these components. We denote the reactions directly by the resulting reaction flux \( V = (V_y, V_x, V_p)^T \), which is divided into three classes

- Exchange reactions \( V_y \in \mathbb{R}^{n_y} \) transport matter between the inside and the outside of the cell;
- Metabolic reactions \( V_x \in \mathbb{R}^{n_x} \) convert the metabolites among each other;
- Biomass reactions \( V_p \in \mathbb{R}^{n_p} \) convert the metabolites into macromolecules or vice versa.

The effect of a flux \( V_i \) on the time evolution of the species is given by the stoichiometric matrix \( S \in \mathbb{R}^{n_y+n_x+n_p,n_r} \) as

\[
\begin{pmatrix}
Y \\
X \\
P
\end{pmatrix} = S
\begin{pmatrix}
V_y \\
V_x \\
V_p
\end{pmatrix} =
\begin{pmatrix}
S_yV_y \\
S_xV_x \\
S_pV_p
\end{pmatrix},
\]

(1)

where the \( S_{i,j} \) describes the stoichiometry of species \( i \) in reaction \( V_j \). The macromolecules \( P \) are usually composed of a large number of the small metabolites \( X \), which may lead to ill-conditioned dynamics with large coefficients in the matrix \( S_p \). We tackle this by scaling the macromolecules and the according fluxes by a large factor \( \alpha \) as

\[
\tilde{P} = \alpha P \\
\tilde{V}_p = \alpha V_p.
\]

Furthermore, we assume the internal metabolites to be in a quasi-steady-state \( \dot{X} = 0 \). This assumption is valid, because the metabolic reactions \( V_x \) are much faster in comparison to the biomass reactions \( V_p \) and the external species \( Y \) are slow variables due to the large external volume (cf. [5]). This gives us the boundary layer condition of the scaled system as

\[
S_yV_y + S_xV_x + \alpha^{-1}S_p\tilde{V}_p = 0,
\]

(2)

so that \( S_p \) is normalized with \( \alpha \). For convenience we will write \( \tilde{V} = (V_y, V_x, \tilde{V}_p) \) for the scaled fluxes. Let us first discuss the other constraints we wish to include into our model. The most important one is the enzyme capacity constraint. Most of the macromolecules \( P \) are enzymes whose amounts limit the possible reaction fluxes. We assume that the first \( m \) entries in the vector \( P \) correspond to these enzymes. The maximal reaction rates are determined by the reaction specific catalytic constant \( k_{\text{cat}, \pm j}, j \in \{1, \ldots, r\} \). We differentiate between the forward constant \( k_{\text{cat},+j} \) and the backward constant \( k_{\text{cat},-j} \), if the reaction is reversible and both directions are catalyzed by the same enzyme. Since some enzymes are capable of catalyzing different reactions, we denote the set of reactions catalyzed by the enzyme \( P_i \) as

\[
\text{cat}(i) = \{ j \in \mathbb{N} \mid P_i \text{ catalyzes } V_j \}.
\]

The constraint for the enzyme \( P_i \) then reads in short

\[
\sum_{j \in \text{cat}(i)} \left| \frac{V_j}{k_{\text{cat}, \pm j}} \right| \leq P_i,
\]

(3)
To eliminate the absolute value in constraint (3) and transform it into a linear constraint, we have to include all possible sign-combination of the occurring $k_{\text{cat}}$-values. As an illustration consider $P_1$ catalyzes $V_1$ reversibly and $V_2$ irreversibly. Then we can write (3) as

$$H_{c,1}V = \begin{pmatrix} k_{\text{cat},+1} & k_{\text{cat},+2} & 0 & \cdots & 0 \\ -k_{\text{cat},-1} & k_{\text{cat},+2} & 0 & \cdots & 0 \end{pmatrix} V \leq \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \end{pmatrix} P = H_{E,1}P.$$  

The concatenations of the matrices $H_{c,1}$ to $H_{c,m}$, resp. $H_{E,1}$ to $H_{E,m}$, are written as $H_c (H_E)$ in short. We include the scaling of $\tilde{P}$ by splitting this into the different kinds of fluxes as

$$\alpha H_{c,y}V_y + \alpha H_{c,x}V_x + H_c \tilde{V}_p \leq H_E \tilde{P}.$$  

and write this shortly as

$$\tilde{H}_c v - H_E \tilde{P} \leq 0.$$  

(4)

Usually, the cell needs a certain amount of structural macromolecules to keep working, for example the membrane separating the cell from the outside. We portray this biological necessity with the help of the biomass composition constraint. Hence, we define the scaled total biomass as $b \tilde{P}$, where $b \in \mathbb{R}^{n_p}$ contains the weight of the macromolecules. We assume that the structural macromolecules need to add up to a certain percentage of the biomass, e.g. for the structural protein $\tilde{P}_s$

$$\psi_s b \tilde{P} \leq \tilde{P}_s,$$  

(5)

with $\psi_s$ being the minimal fraction of the total biomass for $\tilde{P}_s$. As before, the extension of (5) to the network level can be expressed by collecting the individual constraint into

$$H_B \tilde{P} \leq 0,$$

where the rows of $H_B$ correspond to $\psi_s b^T - e_s^T$ for different indices $s$. Moreover, we consider the positivity constraint of all species

$$Y \geq 0, \ X \geq 0, \ \tilde{P} \geq 0$$

and biomass-independent flux constraints

$$V_{\text{min}} \leq V \leq V_{\text{max}}.$$  

(6)

The constraint (6) is mostly used to express the irreversibility of a reaction when appropriate.

We reduce the states $(Y, X, \tilde{P})^T$ and the fluxes $V$ by incorporating equation (2). Consider the matrix $T = (S_x^p, S_x^s, \alpha^{-1}S_p^x) \in \mathbb{R}^{n_x+n_r}$ with rank $n_t \leq \min(n_x, n_r)$. We decompose $T$ via singular value decomposition (SVD) into

$$T = E \Sigma F^* = E (\text{diag}(\sigma_1, \ldots, \sigma_t), 0^{n_x+n_r-n_t}) \begin{pmatrix} W^T \\ M^T \end{pmatrix}.$$  

Following this we can express $T$ as

$$T = T_r W^T = E (\text{diag}(\sigma_1, \ldots, \sigma_t)) W^T,$$

with the full rank matrix $T_r \in \mathbb{R}^{n_x+n_t}$ and the unitary matrix $W \in \mathbb{R}^{n_r+n_t}$. For $M \in \mathbb{R}^{n_r+n_t}$ it holds

$$W^T M = 0.$$  

Exploiting this, we can substitute the scaled fluxes $v$ as

$$v = W w + M u,$$  

4
with \( w \in \mathbb{R}^{n_t} \) and \( u \in \mathbb{R}^{n_r - n_t} \). The boundary layer condition is then expressed by
\[
0 = TV = TW w + TM u = T_r W w + T_r W^T M u = T_r w.
\]
Since \( T_r \) has full rank, the only solution is \( w = 0 \). Hence, we can remove \( w \) from the substitution.

The last step to construct the optimization problem is the introduction of the objective functional
\[
J = \int_0^T b^T \tilde{P} e^{-\varphi t} \, dt,
\]
with a discount factor \( \varphi \geq 0 \) and the weight of the individual macromolecules \( b \). Thus, the objective functional represents the accumulation of the total biomass in the system. The discount factor \( \varphi \) is included to reduce the impact of later times, especially the end-time \( T \). If \( \varphi \) is chosen larger than the growth rate, this can be exploited to uniformly bound the objective function value for varying end-times. Finally, we can formulate the full dynamic optimal control problem as
\[
\max_u \int_0^T b^T \tilde{P} e^{-\varphi t} \, dt
\]
subject to
\[
\begin{align*}
\dot{Y} &= \begin{pmatrix} S_y & 0 & 0 \\ 0 & 0 & S_y \end{pmatrix} M u \\
Y(0) &= Y_0, \quad \tilde{P}(0) = \tilde{P}_0 \\
\tilde{H}_e M u - H_E \tilde{P} &\leq 0 \\
H_B \tilde{P} &\leq 0 \\
Y, P &\geq 0 \\
M u &\leq V_{\text{max}} \\
-M u &\leq V_{\text{min}}.
\end{align*}
\]
Depending on the initial conditions \( Y_0, \tilde{P}_0 \), this linear dynamic optimization problem computes the optimal fluxes \( \tilde{V}(t) \) and the corresponding enzyme levels \( \tilde{P}(t) \) at all times. The problem with this approach is, that some artificial results may arise if the end-time \( T \) is chosen either too large or too small. For example, the optimal solution may only include phases in which only the macromolecules with the best cost-benefit ratio are produced. Whereas, this is from a mathematical standpoint an optimal solution, we know that biological system usually do not behave this way. In a later example, we will see this behavior in which the cells focus under nutrient stress on the production of a structure enzyme. But first we present the effects of varying end-times on the deFBA under varying end-times.

2.2. Example: Enzymatic growth

Our first example is composed of one external nutrient \( N \), one internal metabolite \( A \), two macromolecules \( M, E \), and three irreversible reactions \( V_M, V_A, V_E \) between them. The internal metabolite \( A \) corresponds to an energy source like adenosintriphosphate (ATP) necessary for the production of both types of biomass. Accordingly, \( E \) represents a collection of enzymes catalyzing all three reactions and \( M \) is a storage protein, which is usually much cheaper to produce in comparison to the enzymes. The stoichiometric coefficients are given by the reaction network
\[
\begin{align*}
V_M : \quad &b_M N + \tilde{a}_M b_M A \quad \rightarrow M \\
V_A : \quad &N \quad \rightarrow A \\
V_E : \quad &b_E N + \tilde{a}_E b_E A \quad \rightarrow E.
\end{align*}
\]
First we apply the quasi-steady assumption \( 2 \) for the internal metabolite \( x_A \)
\[
\dot{x}_A = V_A - \tilde{a}_E b_E V_E - \tilde{a}_M b_M V_M = 0
\]
\[
\Leftrightarrow \dot{V}_A = \tilde{a}_E b_E V_E + \tilde{a}_M b_M V_M.
\]
Table 1: Numerical values for the parameter in Example 1

| parameter | \( \tilde{a}_M \) | \( \tilde{a}_E \) | \( b_M \) | \( b_E \) | \( k_A \) | \( k_M \) | \( k_E \) |
|-----------|-----------------|-----------------|----------|----------|----------|----------|----------|
| value     | 0.33            | 1               | 1.5      | 1.5      | 2        | 1        |

This gives us the dynamics for the other species as

\[
\begin{pmatrix}
\dot{x}_N \\
\dot{x}_M \\
\dot{x}_E
\end{pmatrix} =
\begin{pmatrix}
-a_M b_M & -a_E b_E \\
1 & 0 \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
v_M \\
v_E
\end{pmatrix},
\]

with \( a_M = \tilde{a}_M + 1 \), \( a_E = \tilde{a}_E + 1 \). As already mentioned, \( E \) is an enzyme catalyzing all the reactions with the catalytic constants \( k_A, k_E, k_M > 0 \). Therefore, the enzyme capacity constraint \(^1\) is given as

\[
\frac{v_M}{k_M} + \frac{v_E}{k_E} + \frac{\tilde{a}_E b_M v_M + \tilde{a}_M b_M v_M}{k_A} \leq x_E.
\]

The biomass vector is given as \( b^T = (b_M, b_E) \) and the objective function reads

\[
J = \int_0^T (b_M x_M + b_E x_E) \exp^{-\varphi t} \, dt.
\]

The example is a very simplified look at a metabolic network. Therefore, we do not include any biomass composition constraint. Please note that we use a discounted objective function as suggested in (7) with a discount factor \( \varphi = 0.25 \). For the numerical solution of the system we used the parameter values presented in Table 1 and initial values \( N_0 = 20 \), \( M_0 = E_0 = 0.1 \). We evaluated this problem for changing end-times \( T \in \{2.5, 5, 7.5\} \) and plotted the results in Figure 1. For none of these solutions the nutrients \( N \) deplete, so we do not include those in the plot. Most importantly, the solution for the end-time \( T = 2.5 \) shows an increase in the macromolecules \( M \), which we call linear growth. This means in fact, that on short times producing only \( M \) leads to a larger objective value. For larger end-times, we see another behavior. In the respective solutions, the deFBA predicts an exponential growth phase in the beginning. This means for this system an expression of only the enzyme \( E \), leading to a quicker uptake of nutrients. The length of the exponential growth phase is depending on the chosen end-time. For larger end-times the mixed behavior becomes optimal. This idea of linear and exponential growth can also be generalized to larger systems. We call every part of an solution curve an exponential growth phase, if enzymes necessary for nutrient uptake are being produced. This is opposed to either linear growth or a stationary phase. Depending on the end-time, the switching time between exponential and linear phase is changing for this example. This is very interesting, as this presents us a weakness of the deFBA for applications in which a suitable end-time \( T \) is a priori unknown. In addition, the deFBA assumes that the biological system has perfect knowledge about the environment and its development over all times. Since this is obviously not true, we introduce the short-term deFBA as a first step.

### 3. Short term deFBA

As the first extension to the deFBA method we pick up some ideas from Model Predictive Control (MPC) \(^14\). Foremost, we are interested in the idea of the receding prediction horizon and how this affects our method. In the previous section we have calculated the optimal fluxes \( \tilde{V}(t) \), \( t \in [0, T] \) for all times at once, now we want to supply the solver only with knowledge on the near future. Therefore, we introduce a prediction horizon \( p > 0 \), \( p \ll T \) representing the time the optimizer is planning ahead. We also introduce a step size \( h > 0 \), \( h \leq p \) as the length of the part of the calculated solution curve we want to use in the final solution. In the original deFBA scheme \(^8\) we have used an discount factor \( \varphi \) to diminish the impact of
Figure 1: Numerical results for example network (9) with changing end-times. The solid line is the amount of the enzyme E and the dashed line presents M. Results for $T = 2.5$ are marked with (□), (circ) marks stand for $T = 5$ and the (x)-mark corresponds to $T = 7.5$.

later times in the optimization. From here on, we set $\varphi := 0$ as this weight is not necessary for short time evaluations. This leads to an iterative scheme over the control times $kh, k \in \{0, \ldots, N\}$.

$$\max_u \int_{kh}^{kh+p} b^T \hat{P} dt$$

s.t. 

$$\begin{pmatrix} \dot{Y} \\ \dot{P} \end{pmatrix} = \begin{pmatrix} S_y & 0 & 0 \\ 0 & 0 & S_p \end{pmatrix} Mu$$

$$Y(kh) = Y_k, \quad \hat{P}(kh) = \hat{P}_k$$

$$\dot{H}_c Mu - H_E \hat{P} \leq 0$$

$$H_B \hat{P} \leq 0$$

$$Y, P \geq 0$$

$$Mu \leq V_{\text{max}}$$

$$-Mu \leq V_{\text{min}}.$$ 

After solving the problem (12) for fixed $k$, we set $Y_{k+1} = Y((k+1)h), \hat{P}_{k+1} = \hat{P}((k+1)h)$ and save the results to the solution curve $u(t)$. For our numerical results we use an equidistant time grid $\Delta t$ with the step size $h$

$$\Delta t = \{tk = kh \mid k = 0, \ldots, N\}$$

and denote the approximation of the states $Y_k, \hat{P}_k$ and controls $u_k$ on this grid. We further assume that the prediction horizon can be expressed as $p = \hat{p}h$. The dynamics of the system are substituted by a collocation method $f : \mathbb{R}^{n_y} \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_t-n_t} \to \mathbb{R}^{n_y+n_p}$. As an example, the problem (12) can be discretized with an
1-step collocation method $f$ as

$$
\max_{\{u_t\}_{t \in (k, \ldots, k+p)}} \sum_{t=k}^{k+p} b^T \tilde{P}_t
$$

with given $Y_k, \tilde{P}_k$

and $\forall l \in \{k+1, \ldots, k+p-1\}$

$$
\begin{pmatrix}
Y_{l+1} \\
\tilde{P}_{l+1}
\end{pmatrix} = f(Y_l, \tilde{P}_l, u_l)
$$

(13)

We call the scheme (13) the short-term deFBA (sdeFBA). Solving the system starting at the $k$-th time step, gives us the optimal fluxes for the next $\hat{p}$-time steps. However, we only implement the first control input $\{u_k\}$ and continue with the next iteration for $k + 1$. This approach is beneficial in multiple ways. Firstly, the computation can be stopped after each iteration if, for instance, the nutrients $Y$ are depleted. Secondly, we can integrate updates to the values of $Y, \tilde{P}$ after each iteration, if new data is available. Thus, we can implement the sdeFBA as a predictor inside an MPC scheme. The most relevant advantage for us is the availability of algorithms to handle uncertainties in such problems via multi-stage NMPC [12].

Since we have already seen in Example 2.2 that different end-times $T$ may lead to different results, we need to choose a suitable prediction horizon $p$ for the sdeFBA. In most applications of MPC the prediction horizon is dictated by real-time requirements for the computation and is chosen maximal with this regard. For our approach we are interested in a prediction horizon $p_{up}$ such that a solution including exponential growth phases is attainable. This means, we need to choose $p_{up}$ large enough such that no purely optimal linear solutions exist, whilst keeping $p_{up}$ small for the sake of low computational cost.

3.1. Choosing the prediction horizon

Our idea to choose the prediction horizon is based on the results of Example 2.2. We distinguished between linear growth, in which nutrient uptake is constant, and exponential growth, including the production of necessary transport enzymes. By analyzing those kinds of growth behavior, we can systematically choose the prediction horizon. We do this by first specifying some initial values $\tilde{P}_0$ for the biomass composition. This can be done by either choosing $\tilde{P}_0$ directly, or using the Resource Balance Analysis (cf. [1]) to derive an optimal biomass composition for certain environmental conditions. Afterwards, we define an upper bound on linear growth possible under those conditions, $b^T \tilde{P}_{lin}$, and search for a lower bound on exponential growth, $b^T \tilde{P}_{exp}$. The time of intersection of these curves is used as the prediction horizon $p_{up}$. As shown in Figure 2 using this horizon in the sdeFBA leads to exponential growth. Therefore, we first identify an upper bound on linear growth by maximizing the slope of the curve $b^T \tilde{P}_{lin}$ at the single point $t = 0$. Additionally, we set the system dynamics to $\dot{\tilde{P}} = Mu_{lin}$, with $u_{lin}(\tilde{P}_0) \in \mathbb{R}^{t-r}$. This translates to the optimization problem similar to the standard FBA [1]

$$
\max_{u_{lin}} b^T (0, 0, S_p^p) Mu_{lin}
$$

s.t. $\tilde{H}_c Mu_{lin} - H_E \tilde{P}_0 \leq 0$

$Mu_{lin} \leq V_{max}$

$- Mu_{lin} \leq V_{min}$

$\tilde{P} \geq 0$

8
We assume that the biomass flux stays constant over time, as no high-cost transport enzymes are produced, and identify it with the maximal linear growth. Of course, this can only act as an upper bound, as we do not include any nutrient dynamics. We use the solution $u_{\text{lin}}(\bar{P}_0)$ to obtain the according biomass trajectory by solving the differential equation

$$\dot{\bar{P}}_{\text{lin}} = S_p M u_{\text{max}} \Rightarrow \bar{P}_{\text{lin}}(t) = \bar{P}_0 + t S_p M u_{\text{max}},$$

and calculate the integrated biomass curve until time $\tau$ as

$$B_{\text{lin}}(\tau, u_{\text{max}}, \bar{P}_0) = \int_0^\tau b^T \dot{\bar{P}}(t) \, dt = \tau b^T \bar{P}_0 + \tau^2 b^T S M u_{\text{max}}.$$ (14)

This curve is an admissible solution to the deFBA constraints, but it will not reflect the optimal solution for a real application. Additionally, we need an additional bound for purely exponential growth solutions. Hence, we construct another optimization problem inspired by the RBA. In a RBA problem one is looking for enzyme levels which maximize the growth rate $\mu(\bar{P}_0) \in \mathbb{R}$ of the system. We enforce balanced growth \[15\], that is exponential growth in all macromolecules at the same level, via the dynamics $\dot{\bar{P}} = \mu \bar{P}$. As before, we solve this only at time $t = 0$ and identify the flux vector $u_{\text{exp}}(\bar{P}_0) \in \mathbb{R}^{1-r}$ realizing this growth rate.

$$\max_{u_{\text{exp}}, \mu} \mu$$

s.t. $\mu \bar{P}_0 = (0, 0, S_p^T) M u_{\text{exp}}$

$$\bar{H} c M u_{\text{exp}} - H E \bar{P}_0 \leq 0$$

$$M u_{\text{exp}} \leq V_{\text{max}}$$

$$-M u_{\text{exp}} \leq V_{\text{min}}$$

$$\bar{P} \geq 0.$$

The biomass curve with the maximal growth rate $\mu$ can be identified by solving the system dynamics

$$\dot{\bar{P}} = \mu \bar{P} \Rightarrow \bar{P}(t) = \bar{P}_0 \exp(\mu t)$$
and the according integrated biomass curve is then given as

\[ B_{\text{exp}}(\mu, \tau, \tilde{P}_0) = \int_0^\tau b^T \tilde{P}(t) \, dt = \mu^{-1} b^T \tilde{P}_0 \exp(\mu \tau) - \mu^{-1} b^T \tilde{P}_0. \]  
(15)

This solution also satisfies the deFBA constraints (5) but at the same time represents a suboptimal solution to the original deFBA problem. It is surely suboptimal as depending on \( \tilde{P}_0 \) enzymes are produced, which are not needed in the actual environment. Therefore, we interpret (15) as a lower bound for exponential growth. We calculate the intersection time \( p_{\text{up}} \) by solving

\[ B_{\text{lin}}(p_{\text{up}}) = p_{\text{up}} b^T \tilde{P}_0 + p_{\text{up}}^2 b^T S_M u_{\text{max}} = \mu^{-1} b^T \tilde{P}_0 \exp(\mu p_{\text{up}}) - \mu^{-1} b^T \tilde{P}_0 = B_{\text{exp}}(p_{\text{max}}). \]  
(16)

The solution \( p_{\text{up}} = 0 \) always exists for (16). If this is the only solution, the sub-optimal biomass curve \( B_{\text{exp}} \) is larger at all times and we can choose the prediction horizon arbitrarily and still get an exponential solution from the sdeFBA. Secondly, there may exists another solution for (16) with \( p_{\text{up}} > 0 \). This implies that on a horizon \( p \geq p_{\text{up}} \) even an sub-optimal exponential solution outgrows the best linear solution (cf. Figure 2). Thus, as long as nutrients do not deplete and we choose \( p \geq p_{\text{up}} \), the optimal solution for (12) must be of an exponential form as this. For the experiments conducted so far, we can even reproduce the results of the original deFBA by using \( p_{\text{up}} \). We can only stress that this scheme does not include the actual nutrient dynamics, thus, the prediction horizon may be chosen smaller and still reproduce the original deFBA behavior.

3.2. Example: Enzymatic growth

Let us revisit Example 2.2 and show the effects of different predictions horizon. We use exactly the same values as before, see Table 1. Following the idea from Section 3.1 we obtain the linear solution (14) as

\[ B_{\text{lin}}(\tau) = \frac{9}{100} \tau^2 + \frac{1}{4} \tau \]

and the exponential solution (15) as

\[ B_{\text{exp}}(\tau) = \frac{5}{8} e^{0.4 \tau} - \frac{5}{8}. \]

Solving (16) then gives us \( p_{\text{up}} \approx 3.88 \). We are interested in the different behaviors for the system observed with different horizons. Hence, we depict the results for \( p \in \{1, 1.2, 3.9\} \) with an end time of \( T = 8 \) in Figure 3. As calculated by the results of Section 3.1 the trajectories for \( p = 3.9 \) (blue lines) show a behavior comparable to the standard deFBA solution, meaning at early times only the enzyme \( E \) is produced. From time \( t = 5.4 \) onwards the solver predicts also an increase in the amount of \( M \). This is exactly the time at which the prediction horizon includes the time \( t = 7 \) at which the nutrients will deplete. We see the same behavior in the red curve (\( p = 1.2 \)) showing the production of \( M \) only starts if the point of depletion is included in the solution. For \( p \leq 1 \) only linear growth is taking place and no enzymes are produced. While a purely linear solution is not reflecting a real biological system, we can not decide for this simplified example which of the exponential solutions is more realistic.

4. Robust deFBA

All biological systems encounter some form of uncertainty. To name only some of those, one can think of sudden changes in temperature, nutrient situation, errors in gene expression, etc. Our robust deFBA framework is capable of handling these kinds of uncertainties, nevertheless, we constrain ourselves for this work to a second level uncertainty based in the model itself. In case of the deFBA we choose to include measurement errors in the \( k_{\text{cat}} \)-values. While there exists a variety of approaches in the MPC environment to tackle this, we focus here on the multi-stage NMPC approach presented [12, 13], and also presented in e.g. [16] for linear systems. The multi-stage NMPC is based on the assumption that one can represent the
Figure 3: Comparison between sdeFBA results for different prediction horizons. Shown are the results for $p = 1$ (brown, □), $p = 1.2$ (red, ◦), $p = 3.9$ (blue, x).

uncertainties in the system by a branching tree as shown in Figure 4. Each branch represents the effect of an uncertainty $d_j$ and the chosen control $u^j$. A path from the root to a final leaf inside the tree is called a scenario. To make the notation easier to read, we use a upper index for scenarios and a low index for prediction steps. So with each new iterative step $k$, we get another set of scenarios depending on the number of uncertainties $n_d$. This is contrary to the expectation that the real measurement error is fixed for all times. As we are dealing with an unknown factor, which stays unknown over the whole time span, we need to cover all effects which may occur due to the unknowns and the tree is keeping on branching.

The main challenge of this approach is that the scenario tree grows exponentially with the prediction horizon. However, previous results [13] have shown that a full coverage of the scenario tree is usually not necessary and we can assume that the uncertainties stay constant after a certain time span. We choose this time span to be identical to the step size $h$ of the time grid used for the discrete approximation of the system. This way we can simplify the tree to the form presented in Figure 4 (right). Under these assumptions we can further identify the form of each scenario by assuming the uncertainties are all bounded $d^j(t) \in [d^j_{\min}, d^j_{\max}]$, $\forall t \in [0, T]$. Constructing the scenarios with the extreme values for each $d^j$, we can ensure the calculated fluxes $u_0$ are feasible for all scenarios. Thus, considering $n_d$ uncertainties we get a total of $2^{n_d}$ different scenarios, with the length of each scenario is defined by the prediction horizon $p$. Figure 4 shows how to handle a system with three uncertainties where each branch corresponds to a uncertainty $d^j$ of the set

$$d^j \in \{ (d^1_{\min}, d^2_{\min}, d^3_{\min}), (d^1_{\min}, d^2_{\max}, d^3_{\min}), (d^1_{\min}, d^2_{\min}, d^3_{\max}), (d^1_{\min}, d^2_{\max}, d^3_{\max}),$$

$$(d^1_{\max}, d^2_{\min}, d^3_{\min}), (d^1_{\max}, d^2_{\max}, d^3_{\min}), (d^1_{\max}, d^2_{\min}, d^3_{\max}), (d^1_{\max}, d^2_{\max}, d^3_{\max}) \}.$$

Therefore, we get eight different scenarios in this case. The first control $u_0$ is the most important one, as this is the one we want to implement to the final solution in each iterative step. This control has to be feasible for all uncertainties $d^j$, which we denote with $d$.

We implement the scenario tree into the discrete sdeFBA [13] by optimizing over all scenarios at once.
The new objective function is chosen as the sum of the individual objectives \( \tilde{J} \)

\[
\hat{J}(t_0, t_f) = \sum_{j=1}^{2^d} J^j = \sum_{j=1}^{2^d} \int_{t_0}^{t_f} \omega^j b^T \bar{P}^j \, dt,
\]

with start-time \( t_0 \), end-time \( t_f \), and the weights \( \omega^j \). The weights can be used if additional information on the likelihood of a scenario is given, but usually we use \( \omega^j = 1 \). Further, we define an index set \( \mathcal{J} = \{ j \in \mathbb{N} \mid 1 \leq j \leq 2^d \} \) for the sake of clarity. The robust approach then defined as an iterative scheme over \( k \in \{ 0, \ldots, N \} \)

\[
\max_{u^j \in Y} \sum_{j=1}^{2^d} \int_{kh}^{kh+p} b^T \bar{P}^j \, dt
\]

with given \( Y(hk) = Y_k, \bar{P}(hk) = \bar{P}_k \)

and \( \forall j \in \mathcal{J} \)

\[
\begin{pmatrix}
\dot{Y}^j \\
\dot{\bar{P}}^j
\end{pmatrix} = \begin{pmatrix}
S^y_g & 0 \\
0 & S^y_p
\end{pmatrix} M u^j
\]

\[
\begin{align*}
\bar{H}^j M u^j & - \bar{H}_E \bar{P}^j \leq 0 \\
\bar{H}_B \bar{P}^j & \leq 0 \\
Y^j, \bar{P}^j & \geq 0 \\
M u^j & \leq V_{\text{max}} \\
-M u^j & \leq V_{\text{min}}.
\end{align*}
\]

Additionally we have to ensure that the first interval of fluxes \( u^j(t) \) for each predictive step is identical for all scenarios by enforcing

\[
\forall a, b : 1 \leq a, b \leq 2^d, \quad u^a(t) = u^b(t) = u(t), \quad \forall t \in (kh, (k+1)h).
\]
We assume all three horizon can be beneficial for the rdeFBA. Changes in the catalytic constants are expected to slow down the system. Therefore, a larger prediction horizon can be too large for just the recreation of the deFBA results of the nominal system, which may include some nonlinear constraints.

For the problem, we are using a tool for nonlinear problems, as we are planning some enhancements to the deFBA, also offers an efficient implementation of multi-stage NMPC. While the deFBA constructs a strictly linear problem, we are using a tool for nonlinear problems, as we are planning some enhancements to the deFBA, which may include some nonlinear constraints.

We calculate the prediction horizon for the robust deFBA in the way suggested by Section 3.1. While this prediction horizon can be too large for just the recreation of the deFBA results of the nominal system, changes in the catalytic constants are expected to slow down the system. Therefore, a larger prediction horizon can be beneficial for the rdeFBA.

4.1. Example: Enzymatic Growth

We visit Example 2.2 a last time to see the effects of the robust deFBA on the small enzyme system. We assume all three $k_{\text{cat}}$-values for the reactions include an error up to 20% of their nominal values. Hence, we are looking at eight different scenarios consisting of all combinations of extremal values

$$k_{\text{cat},j} \in \{ (k_{A,\text{min}}, k_{E,\text{min}}, k_{M,\text{min}}), (k_{A,\text{min}}, k_{E,\text{max}}, k_{M,\text{min}}), (k_{A,\text{min}}, k_{E,\text{min}}, k_{M,\text{max}}), (k_{A,\text{min}}, k_{E,\text{max}}, k_{M,\text{max}}), (k_{A,\text{max}}, k_{E,\text{min}}, k_{M,\text{min}}), (k_{A,\text{max}}, k_{E,\text{max}}, k_{M,\text{min}}), (k_{A,\text{max}}, k_{E,\text{min}}, k_{M,\text{max}}), (k_{A,\text{max}}, k_{E,\text{max}}, k_{M,\text{max}}) \}.$$ 

In Figure 5 we compare the solutions for the nominal values (blue) with the robust solution (red). We used a prediction horizon $p = 3.9$ as predicted by the results in (2.2) and keep the step size at $h = 0.2$. Obviously, the inclusion of the uncertainties decreases the overall growth rate. This is due to the fact that each implemented flux vector must obey all the enzyme capacity constraint of each individual scenario at the same time. By these means the scenario in which the values $\{k_{A,\text{min}}, k_{E,\text{min}}, k_{M,\text{min}}\}$ are used is acting as an overall upper bound to the fluxes. We call this the minimal scenario. We also plotted the results of the sdeFBA in which we used exactly those values for comparison in brown with □-marks. The small difference between the red (○) and the brown curve (x) are simple results of numerical errors. Thus, we conclude that for such a simple system we do not need the robust simulation as the results for the minimal scenario are also the optimal solution for the robust problem. But this is not the limit to the effects one can observe when including measurement errors inside the rdeFBA. In the upcoming section we present a larger network in which multiple pathways can be used to produce biomass. In larger scale networks the impact of uncertainties is naturally higher and one can expect a qualitative change in the robust solution compared to the nominal solution for any of the individual scenarios.

4.2. Example: Core Carbon Network

In this section, we consider a more sophisticated model for the uptake of different carbon sources and oxygen and the transformation of these into biomass. It corresponds to the core network inside all forms of bacteria. While the previous example has shown already the differences between the deFBA, sdeFBA and its robust counterpart we like to stress the differences with this example. This example was previously studied in [5] with help of the deFBA under changing environmental conditions. We only present one environmental setting and compare the results for the different methods. The biomass in this example has more details as we consider ribosomes $R$ and structural components $S$ as macromolecules. The ribosomes are responsible for the production of all proteins, enzymes and more ribosomes. One can interpret $S$ as the membrane of the cell and therefore the surface area gives us a limit to the uptake of certain nutrients or products. This
Further calls for a biomass composition constraint, which we use to enforce 35% of all biomass is needed to be of type $S$

$$0.35b^T P \leq S.$$  

All transport, metabolic or biomass reactions are shown in the Tables 2-3. The different enzymes are either labeled $E_j$ for catalyzing metabolic reactions or $T_j$ for transport reactions. The catalytic constants $k_{cat}$ for the enzymes are derived from typical values in metabolism ([19] [20] [21]), while the constants for the biomass reactions are taken from measurements inside E.coli [22]. For the rdeFBA we assume that three of the $k_{cat}$-values include a measurement error. We have chosen the values

$$k_{cat,E_F} = 1800 d_{E_F}, \quad k_{cat,E_T} = 1800 d_{E_T}, \quad k_{cat,R,R} = 0.2 d_{R,R},$$

where $d_{E_F} \in [1, 1.3], d_{E_T} \in [0.7, 1.3]$ and $d_{R,R} \in [1, 10]$. The first two affect the efficiency of growth subject to oxygen availability and act in respect to the efficiency of the enzymatic capacity. The $k_{cat,R,R}$-value describes the efficiency of self-reproduction of the ribosomes. Larger values lead to a quicker growth and deFBA simulations have shown that this may also impact the decision if the cell prefers to stay in exponential growth or goes to a linear growth phase in which only $S$ is produced. The structure molecule $S$ is the most efficient macromolecule in terms of biomass efficiency, meaning the cells gains more weight by producing this than any other molecule. But it is also the only molecule in need of the enzyme $E_F$ for its production. So usually, the cell will only produce enough $E_F$ and $S$ to fulfill the biomass composition constraint (18) in an exponential growth phase.

This system is still very simple in comparison to genome-scale models. Yet, the system already captures a large variety of possible behaviors and includes some decisions of the biomass composition. Varying the environmental conditions, especially the presence of oxygen, leads to very different enzyme levels and pathway usage. Thus, we add dynamics to the oxygen in the medium as

$$\frac{d}{dt} O^2_{ext} = V_0 - \gamma_0 O^2_{ext},$$
where $V_0$ is the oxygen inflow and $\gamma_0$ the ventilation rate. These dynamics correspond to an aerated batch process. We use the same objective functional as before, namely

$$J(t_0, t_f) = \int_{t_0}^{t_f} b^T \dot{P} \, dt.$$  

(19)

We used the Resource Balance Analysis [4] to identify suitable initial values. For this we set the initial amounts of Carb1(0) = 2 mol, Carb2(0) = 30 mol and $O_2^{\text{ext}} = 50$ mol and searched for the enzyme distribution leading to maximal growth under these conditions. We constrained the amount of macromolecules at time zero to be $b^T \dot{P}(0) = 0.5$ with the scaling factor for the macromolecules chosen to be $\alpha = 100$. The initial value is shown in Table 3.

The result for the different methods are shown in Figure 6. At the top left the biomass $b^T \dot{P}(t)$ is plotted for the different optimizers. The sdeFBA delivers qualitatively the same results as the deFBA with a slightly decreased growth rate. This is to be expected from the sdeFBA as the solver is working with the finite prediction horizon $p \ll T$. Nevertheless, we only show the sdeFBA results in more detail, since the enzyme plots for the deFBA are indistinguishable from the ones of sdeFBA. The overall growth rate of the robust deFBA is much lower than the other two. The cause of this becomes clear, when looking at the other plots in Figure 6. In the top right plot we compare the time development of the nutrients. While we can also observe a time delay as in the left plot, we also see that the robust solution does not deplete the Carb1 source at first. This is very surprising as the parameters are chosen in a fashion to make Carb1 the preferred source. We explain this by the reduced growth rate in the robust solution, as the uptake of Carb2 is supplying enough $A$ to maximize production. The two plots at the bottom of Figure 6 show the time development of the biomass composition. Here we can see why the growth rate of the robust solution diverges from the other solution. While the sdeFBA predicts an initial increase in Ribosomes along with an increase in enzyme production capacity, the rdeFBA focuses on the production of structure $S$. This behavior can be explained with the uncertainty $d_{R \cdot R}$. The robust solution optimizes for all scenarios at once and four of those include a 10-fold increased efficiency for the self replication of the ribosomes. Therefore, the solver decides against an initial production of ribosomes. The sdeFBA solution on the other hand keeps the amount of structural proteins to the enforced minimum until the prediction horizon includes the time of nutrient depletion. Only then the solution predicts an increase in structure molecules $S$ as it is the most effective biomass component in terms of nutrients to biomass conversion. Overall it is very interesting to see that all solutions reach an identical end-value in total biomass. So in general the effectiveness of the production is not affected by our changes in the catalytic constants. But at the same time we can observe changes in the quality of the solutions and even more in the value of the objective (19).

In Example 4.1 we have presented an equivalence between the rdeFBA results and the sdeFBA result for the minimal example. This effect can not be reproduced in this larger network. We have checked the results of individual sdeFBA problems for all scenarios, whereas, none of these results are identical to the robust solution. Hence, the effects of the uncertainties can only be observed via the full robust problem including all scenarios at once.

Most importantly, it is obvious that the prediction from the non-robust methods violate the enzyme capacity constraints including the measurement errors. Therefore, the modeled cells can not reach the growth rate predicted by the non-robust methods. Additionally, in this example we have seen that the inclusion of measurement errors led to different usage of the pathways. This shows that the rdeFBA can serve as a tool to analyze the sensitivity of a model to measurement errors. We are expecting even larger effects with increased network size and are confident this method can give further insight into the regulation of gene expression.

5. Discussion

In this paper, we have presented two strongly connected new methods for the simulation of metabolic networks coupled with gene expression. The proposed short-term deFBA method (12) can be applied to processes in which the end-time of the process is a priori unknown. By using this iterative scheme, one can
Table 2: Exchange and metabolic reactions, together with their rate constants and catalytic constants $k_{\text{cat}}$. For reversible reactions we use the same value for forward and backward reactions.

| Reaction | Enzyme | $k_{\text{cat}}$ |
|----------|--------|------------------|
| Exchange reactions | | |
| Carb1 $\rightarrow$ A | $T_{C1}$ | 3000 |
| Carb1 $\rightarrow$ A | $T_{C2}$ | 2000 |
| $F_{\text{ext}}$ $\rightarrow$ F | $T_{F}$ | 3000 |
| $H$ $\rightarrow$ $H_{\text{Ext}}$ | $T_{H}$ | 3000 |
| $O_{2}\text{ext} \rightarrow$ O2 | S | 1000 |
| D $\rightarrow$ $D_{\text{Ext}}$ | S | 1000 |
| E $\leftrightarrow$ $E_{\text{Ext}}$ | S | 1000 |
| Metabolic reactions | | |
| A + ATP $\rightarrow$ B | E_B | 1800 |
| B $\rightarrow$ C + 2 ATP + 2 NADH | E_C | 1800 |
| C $\leftrightarrow$ 2ATP + 3D | E_D | 1800 |
| C + 4NADH $\leftrightarrow$ 3E | E_E | 1800 |
| B $\rightarrow$ F | E_F | 1800 |
| C $\rightarrow$ G | E_G | 1800 |
| G + ATP + 2NADH $\leftrightarrow$ H | E_H | 1800 |
| G $\rightarrow$ 0.8C + 2NADH | E_N | 1800 |
| O2 + NADH $\rightarrow$ ATP | E_T | 1800 |

Table 3: List of biomass producing reactions. All reactions are catalyzed by the Ribosomes R. Included are the according $k_{\text{cat}}$ values the weights $b$ of the macromolecules.

| Biomass reactions | $b$ | $k_{\text{cat}}$ | $\tilde{P}(0)$ / (µ mol) |
|-------------------|-----|-----------------|------------------------|
| 400 H + 1600 ATP $\rightarrow$ T_{C1} | 4 | 2.5 | 20.434 |
| 1500 H + 6000 ATP $\rightarrow$ T_{C2} | 15 | 0.67 | 1.177 |
| 400 H + 1600 ATP $\rightarrow$ T_{F} | 4 | 2.5 | 3.373 |
| 400 H + 1600 ATP $\rightarrow$ T_{H} | 4 | 2.5 | 3.373 |
| 500 H + 2000 ATP $\rightarrow$ E_B | 5 | 2 | 33.996 |
| 500 H + 2000 ATP $\rightarrow$ E_C | 5 | 2 | 31.424 |
| 1000 H + 4000 ATP $\rightarrow$ E_D | 10 | 1 | 11.588 |
| 1000 H + 4000 ATP $\rightarrow$ E_E | 10 | 1 | 1.509 |
| 2000 H + 8000 ATP $\rightarrow$ E_F | 20 | 0.5 | 2.876 |
| 500 H + 2000 ATP $\rightarrow$ E_G | 5 | 2 | 15.914 |
| 4000 H + 16000 ATP $\rightarrow$ E_H | 40 | 0.25 | 14.892 |
| 500 H + 2000 ATP $\rightarrow$ E_N | 5 | 2 | 2.4176 |
| 500 H + 2000 ATP $\rightarrow$ E_T | 5 | 2 | 32.534 |
| 4500 H + 1500 C + 21000 ATP $\rightarrow$ R | 60 | 0.2 | 29.268 |
| 250 H + 250C + 250F + 1500 ATP $\rightarrow$ S | 7.5 | 3 | 233.59 |
Figure 6: Comparison of Simulation results for the different methods. (Top left) Plot of biomass $b^T\hat{P}$ over time for the three methods. (Top right) Comparison of the results for the external nutrients. (□) represents the results for sdeFBA and (∗) shows the rdeFBA results. (Bottom left) Biomass composition for sdeFBA. (Bottom right) Biomass composition for robust deFBA.
check for break conditions during the run-time and stop the simulation. Furthermore, the sdeFBA can make large scale networks with large final times manageable by reducing the numerical effort for each iteration. To make this even more useful we are further investigating the effects of varying prediction horizons. Basically, we want to find the minimal prediction horizon such that the sdeFBA can reproduce the according deFBA results. We want to identify this as a property of the network without inclusion of the environmental dynamics, as sketched in Section 3.4. On the other hand, there are some experiments investigating the actual time span bacteria plan ahead. The work [23] observes a decrease in the CO2 production rate of yeast cells 1.5 hours before the main nutrient source, glucose, depletes. The yeast is therefore capable of sensing the nutrient concentration and their gradients. This shows that the cells are preemptively reacting to environmental changes, which can be identified as the implementation of a prediction horizon inside natural metabolic networks. At the same time the experimental results show the cells stay in an exponential growth phase for as long as possible. This puts us in a difficult place, when using the sdeFBA for setting in which all nutrients deplete. While a shorter prediction horizon leads to a prolonged exponential growth phase (cf. Example 3.2) and decreased numerical effort, most biological systems do in fact plan further ahead. This calls for a larger prediction horizon.

The second method we presented is the extension of the sdeFBA to utilizing a robust optimization technique. We called this the robust deFBA [17]. It inherits the idea of the multi-stage nonlinear model predictive control [12] and makes them usable for metabolic networks. We have chosen this approach over other robust approaches for its simplicity and easy implementation in the sdeFBA. Furthermore, the description of uncertainties by discrete events is perfectly suited for our first application with measurement errors. Additionally, the implementation do-mpc is capable to handle nonlinear constraints, which may arise in future projects. It is very simple to see, that is sufficient to use only the maximal deviations from the nominal values of the $k_{cat}$-values for the creation of the different scenarios. While already leading to a considerable reduction in size of scenario tree, the amount of scenarios still increases two-fold with each additional uncertainty. Hence, it is only possible to use either a short prediction horizon or a limited number of uncertainties inside the rdeFBA. Assuming an optimal choice for prediction horizon will arise from further analysis of the sdeFBA, we are focusing on a decrease in the number of the scenarios. We are already investigating this by using a sensitivity analysis on the scenario level. Another solution to the large computation cost can be the application of scenario decomposition approaches such as progressive hedging algorithms (PHA), as suggested by [24]. Their basic idea is the independent solution of each scenario by relaxation of the non-anticipativity constraint [17b].

References
[1] A. Varma, B. Palsson, Metabolic flux balancing: Basic concepts, scientific and practical use Nat Biotech 12 (10) (1994) 994–998.
URL http://dx.doi.org/10.1038/nbt1094-994
[2] A. Varma, B. O. Palsson, Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type Escherichia coli w3110., Applied and environmental microbiology 60 (10) (1994) 3724–3731.
[3] R. Mahadevan, J. S. Edwards, F. J. Doyle, Dynamic flux balance analysis of diauxic growth in Escherichia coli, Biophysical journal 83 (3) (2002) 1331–1340.
[4] A. Goelzer, V. Fromion, G. Scorletti, Cell design in bacteria as a convex optimization problem, Automatica 47 (6) (2011) 1210–1218.
[5] S. Waldherr, D. A. Oyarzún, A. Bockmayr, Dynamic optimization of metabolic networks coupled with gene expression, Journal of Theoretical Biology 365 (2015) 409–485.
[6] M. M. Zavlanos, A. A. Julius, Robust flux balance analysis of metabolic networks, in: American Control Conference (ACC), 2011, IEEE, 2011, pp. 2915–2920.
[7] E. Almaasa, E. Gruberc, A. Holderb, A. Koc, M. MacGillivrayc, M. Sawyerc, Robust analysis of metabolic pathways, submitted.
[8] A. Bemporad, M. Morari, Robust model predictive control: A survey, in: Robustness in identification and control, Springer, 1999, pp. 207–226.
[9] P. J. Campo, M. Morari, Robust model predictive control, in: American Control Conference, 1987, IEEE, 1987, pp. 1021–1026.
[10] P. Scokaert, D. Mayne, Min-max feedback model predictive control for constrained linear systems, Automatic Control, IEEE Transactions on 43 (8) (1998) 1136–1142.
[11] S. V. Rakovic, B. Kouvaritakis, M. Cannon, C. Panos, R. Findeisen, Fully parameterized tube mpc, in: Proceedings of the 18th IFAC world congress, 2011, pp. 197–202.
S. Lucia, T. Finkler, S. Engell, Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty, Journal of Process Control 23 (9) (2013) 1306 – 1319.

S. Lucia, J. Andersson, H. Brandt, M. Diehl, S. Engell, Handling uncertainty in economic nonlinear model predictive control: a comparative case-study, Journal of Process Control 24 (2014) 1247–1259.

J. Rawlings, D. Mayne, Model Predictive Control Theory and Design, Nob Hill Pub, 2009.

C. Baroukh, R. Muñoz-Tamayo, J.-P. Steyer, O. Bernard, Drum: A new framework for metabolic modeling under non-balanced growth. Application to the carbon metabolism of unicellular microalgae, PloS one 9 (8) (2014) e104499.

D. Muñoz de la Peña, A. Bemporad, T. Alamo, Stochastic programming applied to model predictive control, in: Proc. of the 44th IEEE Conference on Decision and Control, 2005, pp. 1361–1366.

A. Wächter, L. Biegler, On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming, Mathematical Programming 106 (2006) 25–57.

J. Andersson, J. Åkesson, M. Diehl, CasADi – A symbolic package for automatic differentiation and optimal control, in: S. Forth, P. Hovland, E. Phipps, J. Utke, A. Walther (Eds.), Recent Advances in Algorithmic Differentiation, Lecture Notes in Computational Science and Engineering, Springer, Berlin, 2012, pp. 297–307.

A. Bar-Even, E. Noor, Y. Savir, W. Liebermeister, D. Davidi, D. S. Tawfik, R. Milo, The moderately efficient enzyme: evolutionary and physicochemical trends shaping enzyme parameters, Biochemistry 50 (21) (2011) 4402–4410.

I. Schomburg, A. Chang, D. Schomburg, Brenda, enzyme data and metabolic information, Nucleic acids research 30 (1) (2002) 47–49.

R. Milo, P. Jorgensen, U. Moran, G. Weber, M. Springer, Bionumbersthe database of key numbers in molecular and cell biology, Nucleic acids research 38 (suppl 1) (2010) D750–D753.

R. Young, H. Bremer, Polypeptide-chain-elongation rate in escherichia coli b/r as a function of growth rate, Biochemical Journal 160 (2) (1976) 185–194.

G. G. Zampar, A. Kümmel, J. Ewald, S. Jol, B. Niebel, P. Picotti, R. Aebersold, U. Sauer, N. Zamboni, M. Heinemann, Temporal system-level organization of the switch from glycolytic to gluconeogenic operation in yeast, Molecular systems biology 9 (1) (2013) 651.

R. T. Rockafellar, R. J.-B. Wets, Scenarios and policy aggregation in optimization under uncertainty, Mathematics of operations research 16 (1) (1991) 119–147.