Real-time economic optimization for a fermentation process using Model Predictive Control

Petersen, Lars Norbert; Jørgensen, John Bagterp

Published in:
Proceedings of European Control Conference (ECC) 2014

Link to article, DOI:
10.1109/ECC.2014.6862270

Publication date:
2014

Document Version
Peer reviewed version

Real-time economic optimization for a fermentation process using Model Predictive Control

Lars Norbert Petersen[1] and John Bagterp Jørgensen[2]

***Abstract*— Fermentation is a widely used process in production of many foods, beverages, and pharmaceuticals. The main goal of the control system is to maximize profit of the fermentation process, and thus this is also the main goal of this paper. We present a simple dynamic model for a fermentation process and demonstrate its usefulness in economic optimization. The model is formulated as an index-1 differential algebraic equation (DAE), which guarantees conservation of mass and energy in discrete form. The optimization is based on recent advances within Economic Nonlinear Model Predictive Control (E-NMPC), and also utilizes the index-1 DAE model. The E-NMPC uses the single-shooting method and the adjoint method for computation of the optimization gradients. The process constraints are relaxed to soft-constraints on the outputs. Finally we derive the analytical solution to the economic optimization problem and compare it with the numerically determined solution.**

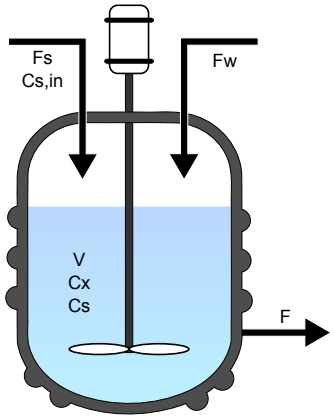

Fig. 1. Simple sketch of the fermentation process.

I. Introduction

Maximizing profit has been and will always be the primary purpose of optimal process operation. Within conventional process control, the economic optimization considerations of a plant are usually indirectly addressed or addressed in a separate real-time optimization (RTO) layer that performs a steady-state economic optimization of the process variables [1]. Recent advances have focused on optimizing the higher-level objectives, such as economics, directly in the process control layer. Model Predictive Control (MPC) has for long time been the preferred framework in both industry and academia because of its flexibility, performance and ability to handle constraints on the inputs as well as the states [2]. Many researchers have also proposed nonlinear MPC which handles nonlinear systems and constraints. Much research has, therefore, been focused on extending the MPC framework to also handle optimization of process economics. The idea of optimizing economics directly has been reported in many works [1], [3]–[5]. Research has also been performed on stability theory, showing that limit cycles may arise because these are economically favourable [6].

The use of fermentation in industry is widely used, and the ability to control a fermentation process at its optimal state is of considerable interest to many fermentation industries. Optimal control reduces the production costs and increases yield while maintaining proper quality of the product. Optimal open-loop time profiles of the feed rates are well known and used, but these solutions depend heavily upon uncertainties in the initial conditions and system parameters which can lead to large errors and thereby profit loss [7], [8]. Therefore, it is advantageous to develop a closed-loop optimization scheme which attenuates uncertainties and is independent of the initial conditions [8].

In this work we address the issue of optimizing the economics directly in the controller. We present a simple fermenter example to demonstrate our results and show the applicability of E-NMPC in a tutorial fashion. The example is simple enough for us to derive the exact solution to the control problem, and thus compare the achieved performance of the discrete nonlinear controller with the analytical solution. We aim at developing an economic nonlinear MPC which optimizes the closed loop performance with respect to an economic objective function for a nonlinear system. We compute the gradients of the optimization problem using the single-shooting method to reduce the computational load. The controller handles both input and (soft) output constraints. The design of the MPC is based on receding horizon control. It is easy to implement and can also easily be used for control of other nonlinear processes to maximize the profit of operation.

We introduce a fermentation model that is based on engineering first principles. It describes the fermentation process as illustrated in Figure 1. The model is partly taken from [9] and describes the fermentation of single cell proteins using Methylococcus Capsulatus. The model is simplified in order to only render the fundamental properties of fermentation. The model describes the hold-up, the biomass concentration and the substrate concentration in a well stirred tank, as a

[1]L. N. Petersen is with the Department of Applied Mathematics and Computer Science, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark and GEA Process Engineering A/S, Søborg, Denmark lnpe@dtu.dk

[2]J. B. Jørgensen is with the Department of Applied Mathematics and Computer Science, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark jbjo@dtu.dk

function of water- and substrate-inlets.

The paper is organized as follows: In Section II we set-up the control problem including discretization and derivation of the gradients of the optimization problem, in Section III we present the fermentation model. Section IV contains the derivation of the analytical solution to the control problem. In Section V we present simulation studies to show the benefit of optimizing the fermenter operation compared to the analytical solution. Conclusions are given in Section VI.

II. Optimization of Dynamical Systems

In this paper, we consider systems of differential equations in the form

$$\frac{d}{dt}g(x(t)) = f(x(t), u(t), d(t)) \qquad t \in [t_0, t_f[\tag{1}$$

where $x(t_0) = x_0$. The form is natural for a large number of systems in process engineering, petroleum engineering, electrical engineering and mechanical engineering. This system representation is also natural for modelling fermenter dynamics. The state function $g(x(t))$ typically represents mass, energy and momentum and $x(t)$ represents the states. The differential equation may also be represented as the index-1 differential equations

$$\frac{d}{dt}h(t) = f(x(t), u(t), d(t)) \qquad x(t_0) = x_0 \tag{2a}$$
$$h(t) = g(x(t)) \qquad h(t_0) = g(x_0) \tag{2b}$$

There are no numerical difference in these two representations. Assuming dg/dt is non-singular, we can rewrite and get

$$\frac{d}{dt}x(t) = \left(\frac{dg}{dx}(x(t))\right)^{-1} f(x(t), u(t), d(t)) \quad x(t_0) = x_0 \tag{3}$$

which is not numerically equivalent to (2). In particular, the representation in (3) does not guarantee conservation of $g(x(t))$, e.g. mass energy and momentum when solved numerically. We will therefore use the representation in (2) when solving the initial value problem of the fermenter model.

The objective function of our optimization is given in Bolza form

$$\phi = \int_{t_0}^{t_f} l(x(t), u(t), d(t))dt + \bar{l}(x(t_f)) \tag{4}$$

where $l(x(t), u(t), d(t))$ is the stage cost and $\bar{l}(x(t))$ is the end cost. In order to optimize the profit of operation, the cost function must represent the cost (or profit) of operating the system in the period $[t_0, t_f]$

The manipulated variables and states are restricted by the constraints

$$c(x(t), u(t)) \geq 0 \qquad t \in [t_0, t_f[\tag{5a}$$
$$\overline{c}(x(t_f)) \geq 0 \tag{5b}$$

where $c(x(t), u(t))$ are the stage constraints and $\overline{c}(x(t_f))$ is the end constraint.

The optimal trajectory for the manipulated variables, $u(t)$, and the states, $x(t)$, are obtained by solution of the continuous-time constrained optimal control problem in Bolza form

$$\min_{[x(t),u(t)]_{t_0}^{t_f}} \quad \phi = \int_{t_0}^{t_f} l(x(t), u(t), d(t))dt + \bar{l}(x(t_f)) \tag{6a}$$

s. t.
$$x(t_0) = x_0 \tag{6b}$$
$$\frac{d}{dt}g(x(t)) = f(x(t), u(t), d(t)), t \in [t_0, t_f[\tag{6c}$$
$$c(x(t), u(t)) \geq 0, t \in [t_0, t_f[\tag{6d}$$
$$\overline{c}(x(t_f)) \geq 0, \tag{6e}$$

A. Discretization

The continous-time constrained optimal control problem is infinite-dimensional. To solve it numerically it must be approximated by a finite-dimensional optimal control problem. The manipulated variables are made finite-dimensional by approximating the input profile by a piecewise constant profile

$$u(t) = u_k \qquad t_k \leq t < t_{k+1} \qquad k \in \mathcal{K} \tag{7}$$

with this discretization, the dynamics may be represented as

$$\frac{d}{dt}g(x(t)) = f(x(t), u_k, d_k), \qquad t_k \leq t \leq t_{k+1} \tag{8}$$

for $\mathcal{K} = 0, 1, ..., N-1$ where $x(t_k) = x_k$. The numerical solution to (8) can be determined by using a simple Forward Euler method. We write it in the residual form

$$R_k(x_k, x_{k+1}, u_k, d_k) = g(x_{k+1}) - g(x_k) - \Delta t_k f(x_k, u_k, d_k) = 0$$

The discretization of $u(t)$ implies the objective function

$$\sum_{k=0}^{N-1} L_k(x_k, u_k, d_k) + L_N(x_N) \tag{9}$$

where we also use Forward Euler for discretization and get

$$L_k(x_k, u_k, d_k) = \Delta t_k l(x_k, u_k, d_k) \tag{10}$$
$$L_N(x_N) = \bar{l}(x(t_f)) \tag{11}$$

The path constraints are relaxed to point constraints

$$C_k(x_k, u_k) = c_k(x(t_k), u(t_k)) \geq 0 \tag{12}$$
$$C_N(x_N) = \overline{c}(x(t_f)) \geq 0 \tag{13}$$

Consequently, the continuous time constrained optimal control problem in (6) is approximated by the following discrete time constrained optimal control problem

$$\min_{\{x_k\}_{k=0}^{N}, \{u_k\}_{k=0}^{N-1}} \quad \phi = \sum_{k=0}^{N-1} L_k(x_k, u_k, d_k) + L_N(x_N) \tag{14a}$$

s. t.
$$x_0 = \overline{x}_0 \tag{14b}$$
$$R_k(x_k, x_{k+1}, u_k, d_k) = 0, k \in \mathcal{K} \tag{14c}$$
$$C_k(x_k, u_k) \geq 0, k \in \mathcal{K} \tag{14d}$$
$$C_N(x_N) \geq 0, \tag{14e}$$

B. Single-Shooting Optimization

The discrete-time finite-dimensional optimal control problem may be solved using single-shooting (control vector parametrization) [10], multiple shooting [11], [12], or the simultaneous method [13]. In these methods, a sequential quadratic programming (SQP) algorithm is typically used for the optimization. Gradient computation is straightforward in the simultaneous method, while either forward sensitivity computation or the adjoint method [14] is used by the single-shooting and the multiple-shooting methods.

We solve the optimization problem in (14) by the single-shooting method (also called vector parametrization, CVP). In this method the system dynamics (14c) are used to eliminate the state variables and express the objective and constraint function as function of the manipulated variables and initial state only. Given the manipulated inputs, $\{u_k\}_{k=0}^{N-1}$, the initial value, x_0, and the requirement, that the system dynamics are observable, the objective function may be expressed as

$$\begin{aligned}\Psi =&\Psi(\{u_k\}_{k=0}^{N-1}; x_0)\\ =& \{\phi = \sum_{k=0}^{N-1} L_k(x_k, u_k, d_k) + l_N(x_N) : \\ & x_0 = \bar{x}_0, \\ & R_k(x_k, x_{k+1}, u_k, d_k) = 0, k \in \mathcal{K}\}\end{aligned} \tag{15}$$

Similarly, the constraint functions may be parametrized

$$\begin{aligned}\chi_k =&\chi_k(\{u_j\}_{j=0}^{k}; x_0)\\ =& \{ C_k(x_k, u_k) : \\ & x_0 = \bar{x}_0, \\ & R_j(x_j, x_{j+1}, u_j, d_j) = 0, j \in \mathcal{K}\}\end{aligned} \tag{16}$$

$$\begin{aligned}\chi_N =&\chi_k(\{u_j\}_{j=0}^{N-1}; x_0)\\ =& \{ C_N(x_N) : \\ & x_0 = \bar{x}_0, \\ & R_j(x_j, x_{j+1}, u_j, d_j) = 0, j \in \mathcal{K}\}\end{aligned} \tag{17}$$

Using (15)-(17), the discrete time constraint optimal control problem may be expressed as

$$\min_{\{u_k\}_{k=0}^{N-1}} \quad \Psi(\{u_k\}_{k=0}^{N-1}; x_0) \tag{18a}$$

$$\text{s. t.} \quad \chi_k(\{u_j\}_{j=0}^{k}; x_0) \geq 0, k \in \mathcal{K} \tag{18b}$$

$$\chi_N(\{u_j\}_{j=0}^{N-1}; x_0) \geq 0 \tag{18c}$$

where $\mathcal{K} = 0, 1, ..., N-1$. It is a nonlinear optimization problem which is solved by using sequential quadratic programming (SQP). In each iteration a convex quadratic problem is solved for which evaluation of Ψ, $\nabla\Psi$, χ_k, χ_N and $\nabla\chi_k$,$\nabla\chi_N$ of the nonlinear problem has to be determined. Ψ and χ are computed directly from (15) and (16) while $\nabla\Psi$ and $\nabla\chi$ is computed using the adjoint method. The system states in the optimization problem are dependent on the manipulated variables, in such a way that past changes have an influence on all the subsequent states. This means that the gradients have to be determined in each iteration. The adjoint method is an efficient method for computation of these gradients.

The algorithm for the adjoint method is presented in Algorithm 1 and 2.

Algorithm 1 Adjoint method for $\nabla_{u_k}\Psi(\{u_k\}_{k=0}^{N-1}; x_0)$

Solve for λ_N in $\nabla_{x_N} R_{N-1}\lambda_N = \nabla_{x_N} L_N$
for $k = N-1$ **to** 1 **do**
  Compute
  $\nabla_{u_k}\Psi = \nabla_{u_k} L_k - \nabla_{u_k} R_k \lambda_{k+1}$
  Solve for λ_k in
  $\nabla_{x_k} R_{k-1}\lambda_k = \nabla_{x_k} L_k - \nabla_{x_k} R_k \lambda_{k+1}$
end for
Compute
$\nabla_{u_0}\Psi = \nabla_{u_0} L_0 - \nabla_{u_0} R_0 \lambda_1$

Algorithm 2 Adjoint method for $\nabla_{u_k}\chi(u_k; x_0)_{j=0}^{N-1}$

Solve for λ_N in $\nabla_{x_N} R_{N-1}\lambda_N = \nabla_{x_N} C_N$
for $k = N-1$ **to** 1 **do**
  Compute $\nabla_{u_k}\chi = \nabla_{u_k} C_k - \nabla_{u_k} R_k \lambda_{k+1}$
  Solve for λ_k in $\nabla_{x_k} R_{k-1}\lambda_k = \nabla_{x_k} C_k - \nabla_{x_k} R_k \lambda_{k+1}$
end for
Compute $\nabla_{u_0}\chi = \nabla_{u_0} C_0 - \nabla_{u_0} R_0 \lambda_1$

As we use the Forward Euler method the derivatives simply become

$$\nabla_{x_k} R_k = -\nabla_{x_k} g(x_k) - T_s \nabla_{x_k} f(x_k, u_k, d_k) \tag{19}$$

$$\nabla_{x_k} R_{k-1} = \nabla_{x_k} g(x_k) \tag{20}$$

$$\nabla_{u_k} R_k = -T_s \nabla_{u_k} f(x_k, u_k, d_k) \tag{21}$$

and the derivatives for the cost function become

$$\nabla_{x_k} L_k = T_s \nabla_{x_k} l_k \tag{22}$$

$$\nabla_{u_k} L_k = T_s \nabla_{u_k} l_k \tag{23}$$

To solve the problem, we use Matlab's `fmincon` with an SQP algorithm. A local optimum is reported if the KKT conditions are satisfied with relative and absolute tolerance of 10^{-9}. A non-optimal solution is returned if the relative cost function or step size changes less than 10^{-9}. The sample time of the NMPC is $T_s = 0.20$ hours.

C. Soft Constraints

When both inputs and states are subject to constraints, the solution may become infeasible. For example when disturbances, which cannot be rejected within the given constraints, hits the system. A method for dealing with infeasibility is to introduce so-called soft-constraints. The constraints are softened by using slack variables with the $l_2 - l_1$ penalty function

$$\phi_s = \sum_{k=0}^{N-1} \left(\frac{1}{2} s_k^T S_w s_k + s_k diag(S_w) \right) \tag{24}$$

with $s \geq 0$ and (14d) are relaxed to $C_k(x_k, u_k) - s_k \geq 0$. The optimizer will then find a solution which minimises the original cost function (14) while keeping the constraint

TABLE I
KINETIC PARAMETERS

Symbol	Value	Unit
γ_S	1.777	kg substrate/kg biomass
μ_{max}	0.37	1/hr
K_S	0.021	kg/m^3
K_I	0.38	kg/m^3

violations as small as possible. The constraints can also be handled without the need for slack variables as shown in [12].

III. FERMENTATION MODEL

A mathematical model is needed for the optimal control and economic analysis of the fermentation process. The aim of this study is to provide a model of the Methylococcus Capsulatus fermentation process. The process is sketched in Figure 1

It is a model with variable volume, substrate and biomass hold-up that are governed by Haldane growth kinetics. The model is deliberately kept simple to illustrate key principles of fermenter operation.

A. Constitutive Relations

The biomass growth is limited by substrate. The overall reaction mechanism is

$$\gamma_S S \rightarrow X \tag{25}$$

The cell growth model is governed by the Haldane expression. The reaction rate can be written as follows

$$r = \mu C_X \tag{26}$$

where the specific growth rate is

$$\mu = \mu_{max} \frac{C_S}{K_S + C_S + C_S^2/K_I} \tag{27}$$

Consequently, the growth of biomass is only limited by the substrate concentration. The production rates of biomass and substrate are

$$R_X = r, \qquad R_S = -\gamma_s r \tag{28}$$

The parameters belonging to the growth of Methylococcus Capsulatus are shown in table I.

B. Conservation equations

A mass balance for the hold-up in the fermenter is governed, assuming constant and identical density of all feed streams and the fermenter content. The mass balances for the biomass and the substrate are also governed and gives

$$\frac{d}{dt}(\rho V) = \rho F_s + \rho F_w - \rho F \tag{29a}$$

$$\frac{d}{dt}(V C_X) = -F C_X + R_X V \tag{29b}$$

$$\frac{d}{dt}(V C_S) = F_s C_{S,in} - F C_S + R_S V \tag{29c}$$

with $V(t_0) = V_0$, $C_X(t_0) = C_{X,0}$ and $C_S(t_0) = C_{S,0}$. The state functions in the model are the total mass, $m = \rho V$, the mass of biomass, $m_X = V C_X$, and the mass of substrate, $m_S = V C_S$. The state variables of the model are the volume, V, the biomass concentration, C_X, and the substrate concentration, C_S. The manipulated variables are the water inlet flow rate, F_w, the substrate inlet flow rate, F_s, and the outlet flow rate, F. When the fermenter is operated in fed batch mode, the outlet flow rate is zero, i.e. $F = 0$.

C. Objective function - Profit

The profit of operating the fermenter is given as the value of the produced biomass minus the cost of the used substrate. This profit, in the period $[t_0,t_f]$, may be expressed as

$$\phi_p = \int_{t_0}^{t_f} (p_X R_X V - p_S F_S C_{S,in}) dt \tag{30}$$

The price of biomass and substrate is 3000\$ and 0\$ respectively. Often we neglect the price of the substrate, as $p_X \gg p_S$. The end cost, $\bar{l}(x(t_f))$, is equal to zero. Optimal operation of the fermenter seeks to maximize the profit, ϕ_p, by manipulating the fermenter inputs within the operation constraints.

D. Constraints

The fermentation process is subject to operation restrictions. The manipulated variables are restricted by input constraints

$$0 \leq \quad F_s \quad \leq 30 \tag{31}$$

$$0 \leq \quad F_w \quad \leq 30 \tag{32}$$

$$0 \leq \quad F \quad \leq 60 \tag{33}$$

The outputs, the volume and the concentrations, are restricted by the output constraints

$$0 \leq \quad V \quad \leq 60 \tag{34}$$

$$0 \leq \quad C_X \quad \leq 0.2 \tag{35}$$

These are relaxed in the computations as shown in (24). The end constraint, $\bar{c}(x(t_f))$, is equal to zero.

IV. ANALYTICAL SOLUTION

In this section, we develop an analytical solution to the operation of the fermenter described in section III. As the fermenter is initially almost empty, we start up in fed batch mode, and when the tank is filled, it is operated in continuous mode.

We consider operation of the fermenter in a period $[t_0\ t_f]$ where t_0 and t_f is a large number i.e. $t_f \rightarrow \infty$. In the period $[t_0\ t_N]$ the fermenter is operated in fed batch mode and operated in continuous mode when filled in the period $[t_N\ t_f]$. The profit of such an operation is, when neglecting the cost of substrate ($p_X \gg p_S$) for simplicity,

$$\phi_p = p_X \int_{t_0}^{t_N} R_X(t) V(t) dt + p_X \int_{t_N}^{t_f} R_X(t) V(t) dt \tag{36}$$

In order to maximize the production and profit, $R_X(t)$ and $V(t)$ must be maximized in both periods. From sec. III-A we have

$$R_X^* = r^* = \mu(C_S^*)C_X^* \tag{37}$$

Thus, the maximal production of biomass is obtained when the biomass and substrate concentrations are kept constant at their optimal values

$$C_X^* = C_{X,max} \qquad t \in [t_0, t_f] \tag{38}$$

$$C_S^* = \max \mu(C_S) = \sqrt{K_I K_s} \qquad t \in [t_0, t_f] \tag{39}$$

In this case the production rates, R_X and R_S, become constant values which attain their maximal values at

$$R_X^* = r^*, \qquad R_S^* = -\gamma_s r^* \tag{40}$$

such that the total profit of the biomass production (36) can be expressed as

$$\phi_p = p_X R_X^*(V_{max} - V_0) + p_X R_X^* V_{max}(t_f - t_N) \tag{41}$$

A. Fed batch operation

In order to determine the optimal trajectory for the manipulated inputs we set up the model equations, utilizing that the biomass and substrate concentrations are now constant. $F = 0$ in fed batch mode and the density ρ is constant, which reduces the model to

$$\dot{V}(t) = F_s(t) + F_w(t) \qquad V(t_0) = V_0 \tag{42a}$$

$$\dot{V}(t)C_X^* = R_X V(t) \qquad C_X(t_0) = C_X^* \tag{42b}$$

$$\dot{V}(t)C_S^* = F_s C_{S,in} + R_S V(t) \qquad C_S(t_0) = C_S^* \tag{42c}$$

Substitution of (42a) in (42b) and (42c) yields

$$\begin{bmatrix} C_X^* & C_X^* \\ C_S^* - C_{S,in} & C_S^* \end{bmatrix} \begin{bmatrix} F_s(t) \\ F_w(t) \end{bmatrix} = \begin{bmatrix} R_X^* \\ R_S^* \end{bmatrix} V(t) \tag{43}$$

Solving for $\begin{bmatrix} F_s(t) & F_w(t) \end{bmatrix}^T$ we get

$$\begin{bmatrix} F_s(t) \\ F_w(t) \end{bmatrix} = \begin{bmatrix} \frac{C_S^* R_X^*}{C_{S,in} C_X^*} - \frac{R_S^*}{C_{S,in}} \\ -\frac{R_X^*(C_S^* - C_{S,in})}{C_{S,in} C_X^*} + \frac{R_S^*}{C_{S,in}} \end{bmatrix} V(t) \tag{44}$$

As we have restricted F_w to be greater than zero we require

$$R_X^*(C_S^* - C_{S,in}) \leq R_S^* C_X^* \Rightarrow \tag{45}$$

$$C_{S,in} \geq C_S^* + \gamma_s C_X^* \tag{46}$$

As the volume in the fermenter is given by the simple differential equation in (29a), we can now construct the trajectory of the states and the inputs in fed batch operation. The equation becomes

$$\frac{d}{dt}V = F_s + F_w = \frac{R_X^*}{C_X^*}V(t) \qquad V(t_0) = V_0 \tag{47}$$

Leading to the state evolution

$$V(t) = V_0 \exp\left(\frac{R_X^*}{C_X^*}t\right) \tag{48}$$

and the time, t_N, where the fermenter is filled, $V(t_N) = V_{max}$

$$t_N = \frac{C_X^*}{R_X^*} \log\left(\frac{V_{max}}{V_0}\right) \tag{49}$$

Consequently, the optimal operation of the inputs are given by substituting (48) into (44) and the switch to continuous operation is given in (49).

B. Continuous operation

During continuous operation, the production is optimized by letting the state assume its optimal values

$$V^*(t) = V^* = V_{max} \qquad t \in [t_N, t_f] \tag{50}$$

$$C_X^*(t) = C_X^* = C_{X,max} \qquad t \in [t_N, t_f] \tag{51}$$

$$C_S^*(t) = C_S^* = \sqrt{K_I K_s} \qquad t \in [t_N, t_f] \tag{52}$$

Solving for $\begin{bmatrix} F_s(t) & F_w(t) & F(t) \end{bmatrix}^T$ in (29) utilizing the above we get the optimal input trajectory

$$\begin{bmatrix} F_s^*(t) \\ F_w^*(t) \\ F^*(t) \end{bmatrix} = \begin{bmatrix} \frac{C_S^* R_X^*}{C_{S,in} C_X^*} - \frac{R_S^*}{C_{S,in}} \\ -\frac{R_X^*(C_S^* - C_{S,in})}{C_{S,in} C_X^*} + \frac{R_S^*}{C_{S,in}} \\ \frac{R_X^*}{C_X^*} \end{bmatrix} V^* \tag{53}$$

As the volume, biomass and substrate concentration in the fermenter are measured, we can now construct the optimal inputs for continuous operation.

V. Results

We illustrate the applicability of the E-NMPC by considering a combined simulation, in which the fermenter is started in fed batch mode and then goes into continuous mode when it is full. The analytical solution and simulated states, inputs and the value of the objective function to the fermenter are shown in Figure 2, 3 and 4 respectively.

A. Fed batch operation

The fermenter is started in fed batch mode. We, first of all, notice that the controller is able to control the system to the optimal state when comparing with the analytical solution. Furthermore, the volume of the fermenter increases exponentially. The NMPC uses a simple Forward Euler method with constant step size for state prediction, and we, therefore, see a small offset arise in the substrate and biomass concentrations while the volume increases. A reduced sample time will decrease the deviation from the analytical solution. On the other hand, the deviation do not lead to a significant loss of profit. The violations of the constraints are small and by any practical means do not pose a problem.

B. Continuous operation

Continuous mode is reached when production is continued after the fermenter is filled. Again we note that the correct optimal solution is found compared to the analytical solution. The before mentioned small offset in the substrate and biomass concentrations have vanished due to the constant hold-up. Only small numerical errors on the NMPC solution arise. These errors are due to the Forward Euler method, but are also negligible.

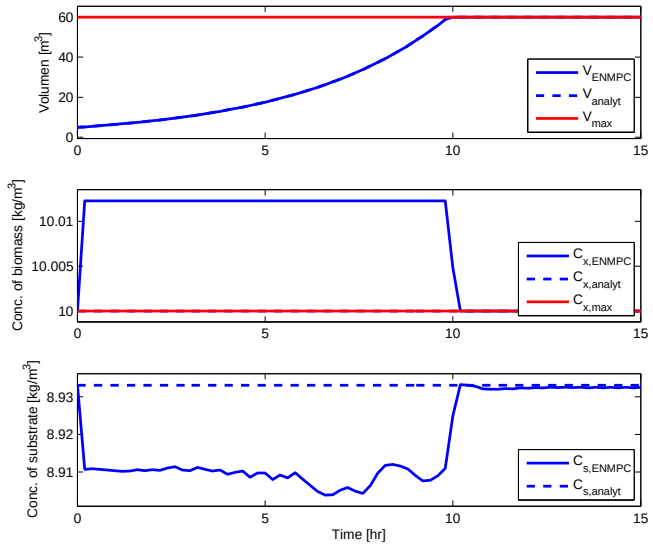

Fig. 2. States of the fermenter. The controller fills the tank and then opens the outlet valve, i.e. starts up in fed batch mode and then continue in continuous mode. The concentrations are kept at almost optimal values during the compete operation.

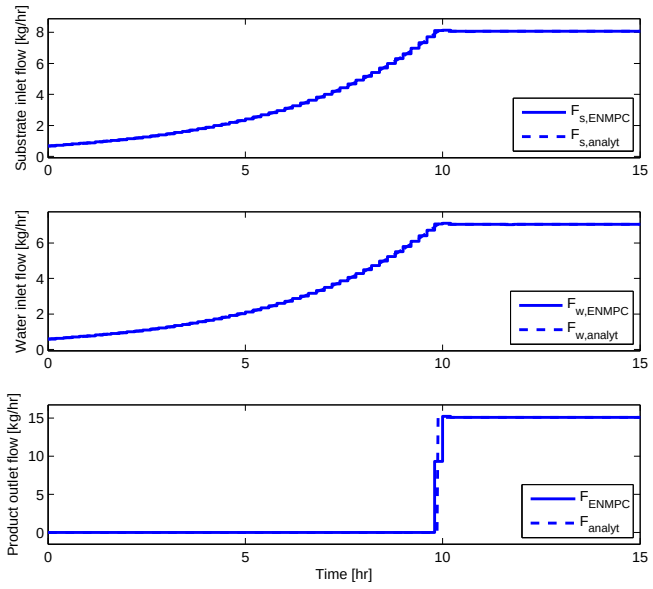

Fig. 3. Inputs to the fermenter. In fed batch mode the flow of water and substrate increases exponentially in order to stabilize the biomass and substrate concentrations. The flows stabilize when the fermenter is filled.

VI. Conclusion

In this paper, we have derived and demonstrated an economically optimizing nonlinear model predictive controller (NMPC) for a fermentation process. The performance of the controller is by any practical means deemed identical to the analytical solution. Only minor economic loss was observed due to the Forward Euler state integration method. A model for the fermentation process of single cell proteins, using Methylococcus Capsulatus, was stated and used directly in the control algorithm. The process was studied under feedback control using the proposed controller in a reseeding

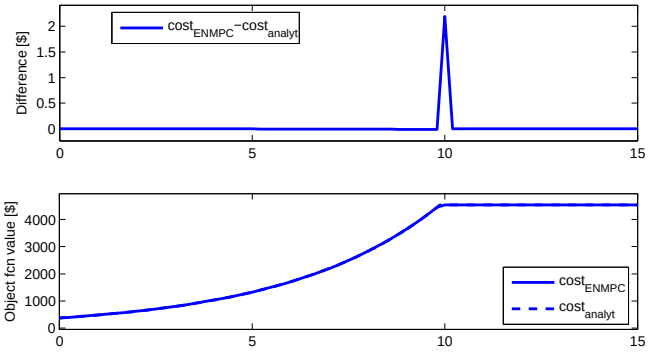

Fig. 4. Objective function of the fermenter compared to the analytical solution. As seen the losses of running an E-NMPC are very small compared to the analytical solution.

horizon setup. The performance of the closed-loop system was studied in fed batch and continuous operation. We have presented a through description of the NMPC algorithm and a fermentation example, which can be used in a tutorial fashion.

References

[1] J. B. Rawlings, C. N. Bates, and D. Angeli, "Fundamentals of economic model predictive control," *Proceedings of the IEEE Conference on Decision and Control*, pp. 3851–3861, 2012.

[2] M. L. Darby, M. Harmse, and M. Nikolaou, "MPC: Current practice and challenges," *IFAC Proceedings Volumes*, vol. 7, no. 1, pp. 86–98, 2009.

[3] R. Amrit, "Optimizing process economics in model predictive control," Ph.D. dissertation, University of WisconsinMadison, September 2011.

[4] R. Halvgaard, N. K. Poulsen, H. Madsen, and J. B. Jørgensen, *Economic Model Predictive Control for Building Climate Control in a Smart Grid*. IEEE, 2012.

[5] T. G. Hovgaard, L. F. Larsen, and J. B. Jørgensen, "Flexible and cost efficient power consumption using economic mpc," *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 848–854, 2011.

[6] D. Angeli, R. Amrit, and J. B. Rawlings, "Receding horizon cost optimization for overly constrained nonlinear plants," *Proceedings of the IEEE Conference on Decision and Control*, pp. 7972–7977, 2009.

[7] J. M. Modak and H. C. Lim, "Feedback optimization of fed-batch fermentation," *Biotechnology and Bioengineering*, vol. 30, no. 4, pp. 528–540, 1987.

[8] B. Srinivasan, D. Bonvin, E. Visser, and S. Palanki, "Dynamic optimization of batch processes: II. Role of measurements in handling uncertainty," *Computers & Chemical Engineering*, vol. 27, pp. 27–44, 2003.

[9] D. F. Olsen, J. B. Jørgensen, J. Villadsen, and S. B. Jørgensen, "Modeling and simulation of single cell protein production," vol. 11, no. 1, pp. 502–507, 2010.

[10] M. Schlegel, K. Stockmann, T. Binder, and W. Marquardt, "Dynamic optimization using adaptive control vector parameterization," *Computers & Chemical Engineering*, vol. 29, no. 8, pp. 1731–1751, 2005.

[11] H. G. Bock and K. J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," *9th IFAC World Congress Budapest. Pergamon Press*, pp. 242–247, 1984.

[12] A. Capolei and J. B. Jørgensen, "Solution of constrained optimal control problems using multiple shooting and esdirk methods," *Control Conference (ACC), 2012*, pp. 295–300, 2012.

[13] L. T. Biegler, "Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation," *Computers & Chemical Engineering*, vol. 8, no. 3, pp. 243–248, 1984.

[14] J. B. Jørgensen, "Adjoint sensitivity results for predictive control, state- and parameter-estimation with nonlinear models," *European Control Conference 2007*, pp. 3649–3656, 2007.