



Using of Dynamic and Rollout Neuro - Dynamic Programming for Static and Dynamic Optimization of a Fed-batch Fermentation Process

Tatiana Ilkova

Centre of Biomedical Engineering “Prof. Ivan Daskalov” – Bulgarian Academy of Science
105 Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
E-mail: tanja@clbme.bas.bg

Received: May 12, 2005

Accepted: November 30, 2005

Published: December 16, 2005

Abstract. A fed-batch fermentation process is examined in this paper for experimental and further dynamic optimization. The static optimization is developed for to be found out the optimal initial concentrations of the basic biochemical variables – biomass, substrate and substrate in the feeding solution. For the static optimization of the process the method of Dynamic programming is used. After that these initial values are used for the dynamic optimization carried out by a submethod of Neuro-dynamic programming-rollout. The general advantage of this method is that the number of the iterations in the cost approximation part is decreased

Keywords: Dynamic programming, Rollout neuro-dynamic programming, Neural network

Introduction

Many industrial and laboratory important fermentation processes (FP), including production of antibiotics, enzymes, organic acids and etc., are carried out in a fed-batch mode of operation in which the substrates are added continuously contrariwise to batch operation. Fed-batch bioreactors are particularly useful when growth and/or metabolite production is inhibited because of catabolite repression, where controlled addition of the substrate is essential to achieve maximum production of the desired product [4, 7]. This work focuses on a laboratory *E. coli* fed-batch fermentation process and its static and dynamic optimization.

One approach for solving problems of optimization is Dynamic programming (DP), which is successfully applied to fermentation processes [1]. However, the approach is largely considered impractical as due to the analytical solution of resulting dynamic program is seldom possible and the numerical solution suffers from the “*curse of dimensionality*” [2].

Neuro-dynamic programming (NDP) is a relatively new class of DP methods for control and further decision making under uncertainties [2, 8]. These methods have the potential to deal with problems that for a long time were thought to be intractable due to either a large state space or the lack of an accurate model. The name NDP expresses the reliance of the methods on both DP and neural networks (NN) concepts. In this case, in the artificial intelligence community, from where the methods originated, the name *reinforcement learning* is also used. There has been a gradual realization *reinforcement learning techniques* can be fruitfully motivated and interpreted in terms of classical DP concepts such as value and policy iteration [6]. Two fundamental DP algorithms, policy iteration and value iteration, are the starting points for the NDP methodology. A new policy is then defined by minimization of Bellman’s equation, where the optimal cost is replaced by the calculated scoring function, and the process repeats [4, 6]. The method is successfully applied for optimal control of FP in the last



years, as the computing time was decreased about 2/3 times and the quantity of the desired products was increased [3, 4].

A type of NDP method, called *rollout* is applied in recent years. It aims to approximate the optimal *cost-to-go* function by the cost of some reasonably good suboptimal policy, called *base policy*. Depending on the context, the cost of the *base policy* may be calculated either analytically, or more commonly by simulation [6]. This submethod will be used for dynamic optimization of the process.

The aim of this paper is optimization of the initial conditions of the basic kinetic variables of a fed-batch fermentation and further future dynamic optimization.

Model of the process

The model of the process (*E. coli* fermentation) is expressed by the following biochemical variables: biomass concentration, substrate concentration, oxygen concentration in liquid phase and oxygen concentration in gas phase [7]:

$$\begin{aligned}
 \frac{dX}{dt} &= \mu(S, C_L) X - \frac{F}{V} X \\
 \frac{dS}{dt} &= \frac{F}{V} (S_0 - S) - K_1 \eta(\mu) X \\
 \frac{dC_G}{dt} &= K_G a (C_L - C_G) \\
 \frac{dC_L}{dt} &= K_L a (C_G - C_L) - K_2 \mu(S, C_L) X - \frac{F}{V} C_L \\
 \frac{dV}{dt} &= F \\
 \mu(S, C_L) &= \mu_{\max} \frac{S^2}{(K_S + S^2)} \frac{C_L^2}{(K_C + C_L^2)}, \quad \eta = \frac{\mu(S, C_L)}{Y} + K_m
 \end{aligned} \tag{1}$$

where: X is biomass concentration, g/l; S -substrate concentration, g/l; S_0 -substrate concentration in the feeding solution, g/l; C_L -oxygen concentration in liquid phase, %, C_G -oxygen concentration in gas phase, %; $K_G a$ -oxygen mass-transfer coefficient in gas phase, h^{-1} ; $K_L a$ -oxygen mass-transfer coefficient in liquid phase, h^{-1} ; F -feeding rate, l/h; μ -specific growth rate, h^{-1} ; η -specific substrate utilization rate; h^{-1} ; Y -yield coefficient, $g \cdot g^{-1}$; V -bioreactor volume, l; K_S , K_C , K_m , K_i , $i=1,2$ -yield coefficients.

The statistical investigations with comparison between the experimental and the model data show high degree of adequacy of the model. This model will be used for static optimization and further dynamic optimization of the process aiming maximum of biomass concentration at the end of the bioprocess.

Static optimization

The optimization problem can be formulated in the following way: to be found such initial values of the biochemical variables $X(0)$, $S(0)$ and S_0 that maximizing the biomass quantity at the end of the process:

$$\max_{X(0), S(0), S_0} J = \int_{t_0}^{t_f} X(t) dt \tag{2}$$

where: t_0 -initial time, t_f - final time of the fermentation.

The vector of control variables has a type:

$$\mathbf{u} = \mathbf{u} [u_1, u_2, u_3],$$

where: $u_1 = X(0)$ g/l, $u_2 = S(0)$ g/l and $u_3 = S_0$ g/l.

The intervals of variation of variables are:

$$u_{1_{\min}} \leq u_1 \leq u_{1_{\max}} ; u_{2_{\min}} \leq u_2 \leq u_{2_{\max}} ; u_{3_{\min}} \leq u_3 \leq u_{3_{\max}}$$

Based on a few years' experiences it was found that the best reasonable limits of changing of the initial conditions are as follows [7]:

$$(0.08 \leq u_1 \leq 5.00) \text{ g/l}; (2.6 \leq u_2 \leq 3.8) \text{ g/l}; (85.0 \leq u_3 \leq 135.0) \text{ g/l}.$$

To solve this problem DP method was applied. It was established that this method has been successfully applied for finding of initial conditions for desired maximum. An algorithm is synthesized and a program is developed and the following initial conditions of the biochemical variables have been found: $u_1 = 0.13$ g/l; $u_2 = 3.2$ g/l; $u_3 = 110$ g/l.

Using these initial values, the biomass quantity increases with 5.12% in comparison with the original data. Fig. 1 shows the results before and after static optimization for the biomass concentration.

The general problem for such increase of biomass concentration is that in practice the precisely in hundredths quantity initial values of the kinetic variables often cannot infuse descend from measurement errors [8]. Due to that fact it is necessary the stability of the static optimization criterion to be investigated in vicinity of the admissible obtained initial values of X , S and S_0 .

Fig. 2 - 4 present how the criterion depends on control variables. The initial values of biomass concentration, substrate and substrate in the feeding solution are discretized respectively with steps $h_1 = 0.01$, $h_2 = 0.1$ and $h_3 = 1$.

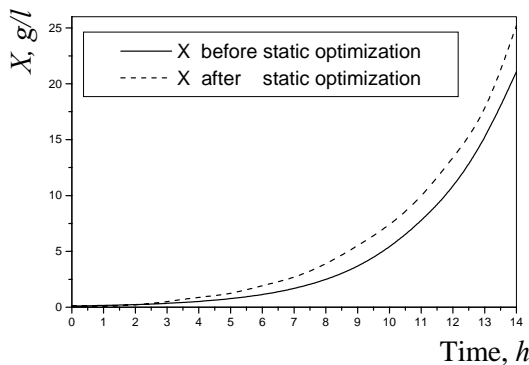


Fig. 1 Results for biomass concentration before and after static optimization

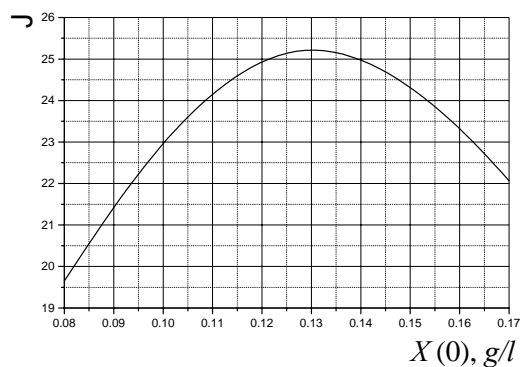


Fig. 2 Criterion J in dependence on $X(0)$

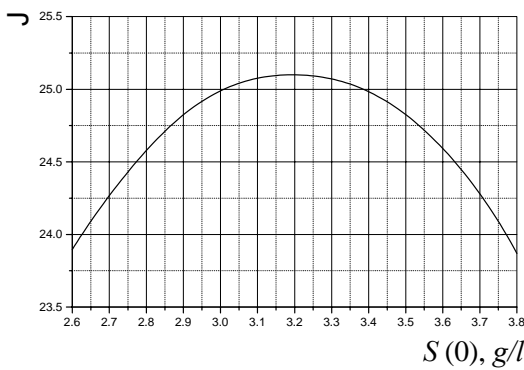


Fig. 3 Criterion J in dependence on $S(0)$

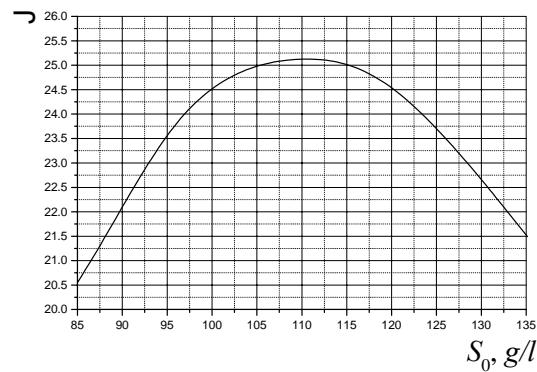


Fig. 4 Criterion J in dependence on S_0

From Fig. 2 - 4 it could be seen that the criterion has a type “plateau” and it is steady in the vicinity of the obtained initial values of X , S and S_0 .

Dynamic optimization

For determination of the optimal control problem of fed-batch fermentation processes maximizing of the optimization criterion at the end of the process $\max J$ (in this case biomass) on the used substrate S is accepted. Thus the optimization problem is reduced to find a profile of the control variable that maximizes the criterion [7]:

$$\max_F J = \int_{t_0}^{t_f} X(t) dt .$$

The feeding rate F is accepted as a control variable.

The optimization problem will be solved by method called *rollout NDP*. *Rollout NDP* is suggested as a method to alleviation the “curse of dimensionality”. It aims to approximate the optimal *cost-to-go* function by the cost of some reasonably good suboptimal policy, called *base policy*. Depending on the context, the cost of the *base policy* may be calculated either analytically, or more commonly by simulation [6].

With this method the model of the process (1) and the vector of the control variable are examined as developing in time processes and they are analyzed of consecutive stages. Admissible values for this bioprocess for the control variable are taken in the interval $0 \leq F \leq 1$, with discrete step $\Delta F = 0.01$. In the fed-batch fermentation process the control variable (F) is limited by [7]:

$$\sum_{i=1}^N F_i = F , \tag{3}$$

then the total quantity feeding substrate is:

$$V_i = \sum_{i=1}^N F_i \cdot h_i , \tag{4}$$

where $h_i = t_i - t_{i-1}$ is the step of discretization.

The control objective is, therefore, to drive the reactor from the low biomass steady state to the desirable high biomass yield state. It may be considered as a step change in the set point at time $t=0$ from the low biomass to the high biomass yield steady state.

Rollout NDP algorithm

The following steps describe the general procedure of the developed method of *rollout* NDP algorithm:

1. Starting with a given policy (some rule for choosing a decision u at each possible state i), and approximately evaluate the cost of that policy (as a function of the current state) by least-squares-fitting a scoring function $\tilde{J}^j(X)$ to the results of many simulated system trajectories using that policy [4];
2. The solution of *one-stage-ahead cost* plus *cost-to-go* problem, results in improvements of the cost values [1];
3. The resulting deviation from optimality depends on a variety of factors, principal among which is the ability of the architecture $\tilde{J}^j(X)$ to approximate accurately the cost functions of various policies;
4. *Cost-to-go* function is calculated using the simulation data for each state visited during the simulation, as for each closed loop simulation (simulation part).
5. A new policy is then defined by minimization of Bellman's equation, where the optimal cost is replaced by the calculated scoring function, and the process repeats. This type of algorithm typically generates a sequence of policies that eventually oscillates in a neighbourhood of an optimal policy;
6. Fit a neural network function approximator to the data to approximate *cost-to-go function* as a smooth function of the states;
7. The improved costs are again fitted to a neural network, as described above, to obtain subsequent iterations $\tilde{J}^1(X)$, $\tilde{J}^2(X)$, and so on ..., until convergence.

In *rollout* NDP the approximation part is partially avoided with help of the simulation part, a simplified scheme is shown in Fig. 5.

Results

A functional approximation relating *cost function* with augmented state was obtained by using neural network with eight hidden neurons, nine input and one output neurons.

Cost is said to be "converged" if the sum of the absolute error was less than 10% of the maximum cost. The cost converged in 7 iterations for our system. The general advantage of this method is that the number of iterations in the cost approximation part is decreased. In this case the number of iterations is diminished from 1560 to 1340 for one step.

The program is developed using *rollout NDP* method. The results show generally an increase of 22.37 % amount of biomass production after the static and dynamic optimization. These results are shown in Fig. 6.

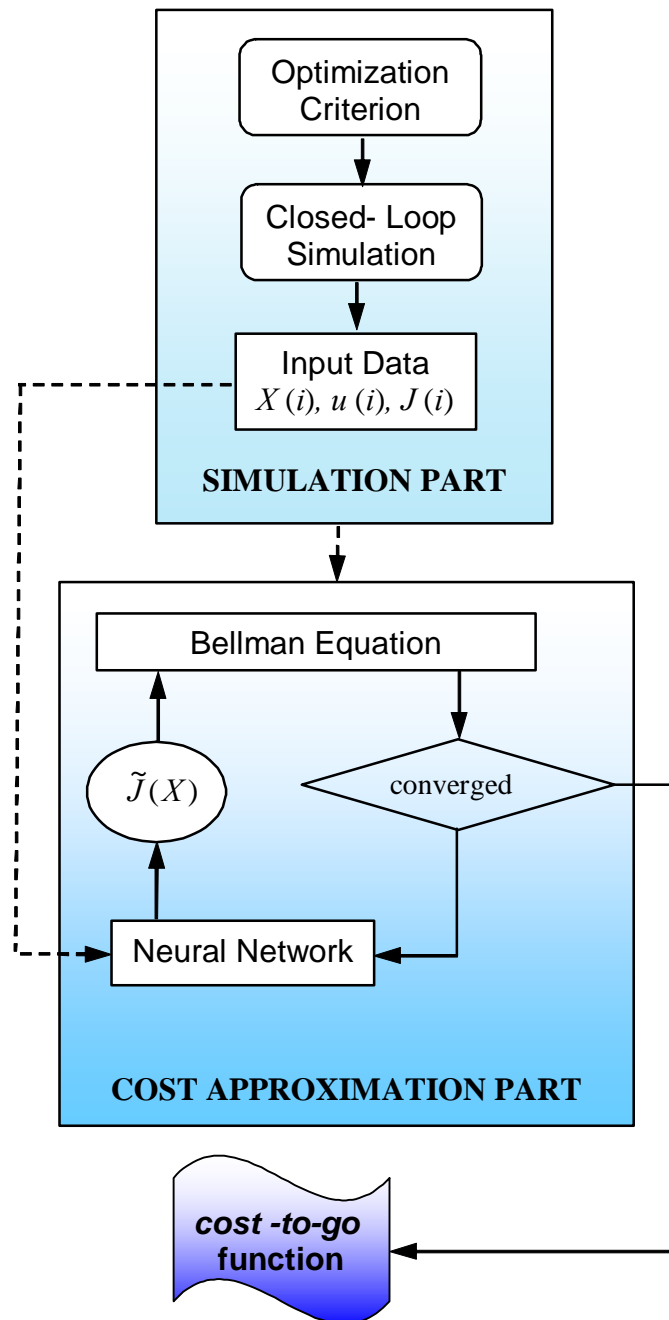


Fig. 5 Dynamic optimization algorithm

Conclusion

A static optimization of an *E. coli* fed-batch fermentation process is performed in order to be found the optimal initial values of the basic biochemical variables. The static optimization criterion is investigated and the results show synonymity and stability of the decision. After that these initial values are used for the dynamic optimization. For this aim a new method - *rollout* NDP is developed and is applied for dynamic optimization of the process. With this method the number of iterations in the cost approximation part are decreased using of the simulation part. In this way number of iterations is diminished. The results shows that *rollout* NDP approach is particularly simple to implement and is applicable for *on-line* implementation.

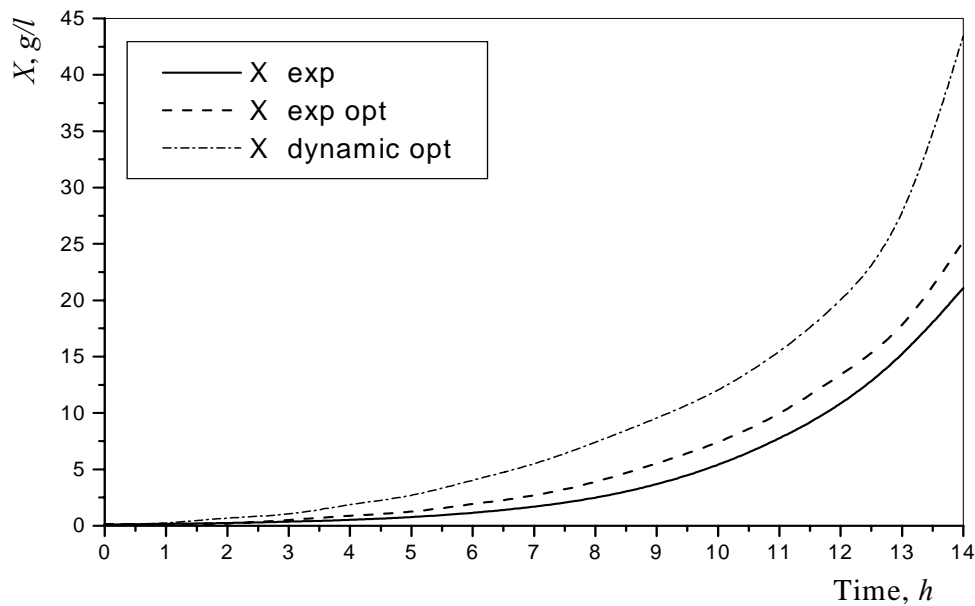


Fig. 6 Optimization results

Acknowledgments

The investigations are partially supported by Bulgarian National Science Fund by grant TH-1314/2003.

References

1. Bertsekas D. (2000). Dynamic Programming and Optimal Control, Vol. II, Athena Scientific, Belmont, MA, 2nd Edition.
2. Bertsekas D., T. Tsitsiklis (1996). Neuro-Dynamic Programming, Athena Scientific, Belmont, MA.
3. Ilkova T. (2004). Application of the Neuro-Dynamic Programming to a Fed-batch Fermentation Process, Technical Ideas, XLI, 80-88.
4. Kaisare N., J. M. Lee, J. H. Lee (2003). Simulation Based Strategy for Nonlinear Optimal Control: Application to a Microbial Cell Reactor, International Journal Robust and Nonlinear Control, 13, 347-363.
5. Mayne, D., J. Rawlings, C. Rao, P. Scokaert (2000). Constrained Model Predictive Control: Stability and Optimality, Automatica, 36, 789-814.
6. Tsitsiklis J., B. Van Roy (1997). An Analysis of Temporal-Difference Learning with Function Approximation, IEEE Transactions on Automatic Control, 42, 674-690.
7. Weuster-Botz D., E. Franco-Laraand (2005). Estimation of Optimal Feeding Strategies for Fed-batch Bioprocesses, Bioprocess and Biosystems Engineering, 27, 255-262.
8. Zheng A., M. Morari (1995). Stability of Model Predictive Control with Mixed constraints, IEEE Transactions on Automatic Control, 40, 1818-1823