OPTIMIZATION OF SEMI-INDUSTRIAL PROCESS FOR *L-LYSINE* AMINO ACID PRODUCTION

Tatiana Ilkova, Mitko Petrov
Institute of Biophysics and Biomedical Engineering – Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria, E-mails: tanja@biomed.bas.bg, mpetrov@biomed.bas.bg

ABSTRACT

In this paper Neuro-dynamic programming (NDP) is proposed as an alternative to alleviate the "curse of dimensionality" of the Dynamic programming (DP) for optimal control of a fed-batch fermentation process in the *L-lysine* production. We used a kinetic model, obtained in our previous investigations. The model is used for optimal control of the process as a control variable is feeding rate. The results show that the quality of *L-lysine* enhances at the end of the process. Also the substrate and computing time decrease. The proposed method is particularly simple to implement and can be applied for on-line optimization.

Keywords: *neuro-dynamic programming, neural network, L-lysine fermentation*

INTRODUCTION

Dynamic programming (DP) is one of the most applied methods for optimal control. *DP* solves many complicated problems of optimal planning and control. The approach application has been considered impractical in complex dynamic systems since analytical solution is difficult to achieve and the so called "*curse of dimensionality*" occurs in the numerical solution. Actually it is one of the most simple and efficient realizable algorithms

Neuro-dynamic programming (NDP) is proposed as an alternative to lighten the "*curse of dimensionality*" of the DP. The term NDP expresses the reliance of the methods, described in this article with respect to both the DP and the neural network concepts [2].

Amino acids are the basic bioelements of proteins, which are the most important macromolecules for the functions of humans and animals. Out of the 20 L-amino acids, which are found worldwide in most of the living organisms, *L-lysine* is one of the nine essential amino acids for human and animal nutrition [1].

L-lysine is an essential amino acid, but cannot be produced by the humans. For this reason *L-lysine* should be received from food. Amino acids are the building blocks of the protein. Lysine is important for proper growth and it plays an essential role in the production of carnitine [1].

The aim of our paper is to obtain an optimal feed rate strategy of a fermentation process in *L-lysine* production using Neuro-dynamic control.

L-LYSINE PRODUCTION MATHEMATICAL MODEL

The kinetic model of the fermentation process is based on the mass balance equations as a perfect mixing in the bioreactor is accepted and it has the following type [1]:

$$\frac{dX}{dt} = \mu X - \frac{F}{V} X \tag{1}$$

$$\frac{dS}{dt} = \frac{F}{V}(S_{in} - S) - k_5 \mu X - k_6 X - k_7 \eta X \tag{2}$$

$$\frac{dTr}{dt} = \frac{F}{V}(Tr_{in} - Tr) - k_{13} \mu X - \frac{F}{V} Tr \tag{3}$$

$$\frac{dC_L}{dt} = k_l a(C^* - C_L) - k_{14} \mu X - k_{15} X - k_{16} \eta X - \frac{F}{V} C_L \tag{4}$$

$$\frac{dL}{dt} = \eta X \tag{5}$$

$$\frac{dV}{dt} = F \tag{6}$$

The specific grown rate of the biomass and specific consumption rate have the form:

$$\mu = \frac{k_1 Tr\ C_L}{\left[(k_2 + Tr)(k_3 + S_0 - S)(k_4 + C_L)\right]} \tag{7}$$

$$\eta = \frac{k_8 S C_L}{\left[(k_9 + S)(k_{10} + S)(k_{11} + C_L)(k_{12} + C_L)\right]} \tag{8}$$

where: μ – specific rate of *L-lysine* synthesis, h^{-1}; η – specific consumption rate of *L-lysine*, h^{-1}; *X* – biomass concentration, g l^{-1}; *L* – *L-lysine* concentration, g l^{-1}; S – glucose concentration, g l^{-1}; *V* – working liquid volume, l; *F* – feed flow rate, l h^{-1}; *Tr* – Threonine concentration, mg l^{-1}; *t* – process time, h; C_L – dissolved oxygen concentration, g l^{-1}; *C** – equilibrium dissolved oxygen concentration, g l^{-1}; S_{in} – input feed substrate concentration, g l^{-1}; Tr_{in} – input feed Threonine concentration, g l^{-1}; k_La – volumetric liquid mass transfer coefficient, h^{-1}.

The initial conditions in the kinetic variables (1) - (8) have the values:

X(0) = X0 = 3.00 g l-1; *S*(0) = S0 = Si = 100.00 g l^{-1}; *Tr*(0) = Tr0 = Tr_{in} = 100.00 mg l^{-1}; *L*(0) = 0.00 g l^{-1}; $C_L(0) = C^* = C_0$ = 6.1x10-3 g l^{-1}; $V(0) = V_0$ = 10.00 l.

The coefficients in the model (1)-(8) have values (after mathematical modeling):

k_1 = 20.8, k_2 = 42.0, k_3 = 28.0, k_4 = 1.1, k_5 = 1.01, k_6 = 0.07, k_7 = 0.51, k_8 = 62.0, k_9 = 28.0, k_{10} = 37.0, k_{11} = 4.0, k_{12} = 0.12, k_{13} = 6.10, k_{14} = 448.0, k_{15} = 22.0, k_{16} = 209.0, k_la = 120.

NEURO-DYNAMIC OPTIMAL CONTROL OF THE PROCESS

The aim of this work is to search the optimal feed flow rate (*F*(*t*)) of a fed-batch process for L-lysine production that will raise *L-lysine* at the final of the process, i.e. [5]:

$$\max_{\mathbf{u}} Q = \int_{t_0}^{t_f} L(t)V(t)\,dt \tag{9}$$

where: t_0 – initial time, t_f – final time of the fermentation=48 h.

A general problem in DP can be defined as follows:

$$\max_{\mathbf{u}_0,\ldots,\mathbf{u}_{k-1}} \sum_{i=1}^{k-1} f\ (\mathbf{W}_i, \mathbf{u}_i) \tag{10}$$

where: W is a vector of the variables of the process, u - vector of control variables, *k* is the present stage.

The objective is to maximize or minimize the combination of the total span and the stagewise, together with the terminal costs subject and the terminal constrains [3].

DP includes a stagewise calculation of the cost-to-go function to reach the solution for the general initial state. The cost-to-go (10) at each stage is defined by:

$$B_i\left(W(t_i), t_i\right) = \max_{u_{\min} \le u_k \le u_{\max}} \Delta t \sum_{k=1}^{N-1} f_k\left(\mathbf{W}_k, \mathbf{u}_k\right) \tag{11}$$

Then the calculation of the cost-to-go function at each stage can be done as:

$$B_i\left(W(t_i), t_i\right) = \max_{\mathbf{u}_{\min} \le \mathbf{u}_k \le \mathbf{u}_{\max}} \left\{f_i\left(W(t_i), \mathbf{u}_i\right) + B\left(W(t_{i+1}), t_{i+1}\right)\right\} \tag{12}$$

Once obtained the cost-to-go function, represents a suitable vehicle to obtain the optimal solution for the total stage.

The classical approach for determining the Bellman equation includes gridding of the state

space, determining the optimization (10) for each grid point and performing the stagewise optimization until accumulation is achieved. The circumstantial sampling of the state space can be avoided by identifying the respective regions of the state space by simulation under prudently chosen suboptimal policies [6, 7, 8].

NDP uses neural network approximations for the approximation of cost-to-go function. The cost-to-go function was not used to generate a determinate control law; instead, it has been used in an on-line optimization to decrease the large horizon problem to a comparatively short horizon problem. The approach was found to be robust to approximation errors. Both deterministic (step changes in kinetic parameters) and stochastic problems (random variations in kinetic parameters and feed composition) were explored [7, 9, 10].

The following notations are used for description of the algorithm:

B – Bellman equation; $\widetilde{B}(x)$ - approximated Bellman equation corresponding to state W;

$()^{i}$ – iteration index for cost iteration loop; k – discrete time.

Finally:

$$\widetilde{B}(k) \equiv \widetilde{B}(W(k)) \text{ and } f(k) = f\left(W(k), \mathbf{u}(k)\right).$$

The general simulation-approximation procedure involves computation of the converged cost-to-go approximation off-line. The architecture of the scheme is shown in **Fig. 1**.

The simulation-based approach involves computation of the converged profit-to-go approximation off-line. The following steps describe the general procedure of NDP algorithm:

1. Performing of simulations of the process with chosen suboptimal policies under all representative operating conditions. Starting with a given policy (a 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 to the results of the many simulated system trajectories using that policy.
2. Calculation of the ∞-horizon cost-to-go for each state visited during the simulation, using the simulation data. The solution of one-stage-ahead cost plus cost-to-go problem results in the improving of the cost values. Cost-to-go is the sum of the single state cost from the next point to the end of the horizon: $B(k) = \sum_{i=k+1}^{N}$
3. The deviation, which is a result of the optimality, depends on a variety of factors, principal among which is the ability of the architecture $\widetilde{B}^{i}(W)$ to approximate accurately the cost functions of the various policies.
4. A new policy is then defined by minimizing 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 oscillate in a surrdounding of an optimal policy.
5. Fitting a neural network function approximator to the data to approximate the cost-to-go function as a smooth function of the states.
6. As described above the improved costs are again fitted to a neural network, to obtain subsequent iterations $\widetilde{B}^{1}(k)$, $\widetilde{B}^{2}(k)$, and so on …, until convergence is accomplished.
7. Policy update may sometimes be necessary to increase the coverage of the state space. In this case more suboptimal simulations with the updated policy are used to increase the coverage or the number of the data points in certain region of the state space.
8. The NDP algorithm block- scheme is shown in **Fig.1**.

The following values of feeding rate are examined: $F \in [0.1,\ 0.2,\ 0.3,\ 0.7,\ 0.8]$, that can cover the possible rang of variations.

The converged cost-to-go function mentioned above was used for determine the one-stage-ahead problem. The choice for switch over the one-stage-ahead of the control is computing by:

$$\mathbf{u}(k) = \arg \max_{\mathbf{u}(k)} \left\{ f\ (\frac{Q(t_k)}{t_k}, \mathbf{u}) + \widetilde{B}^6 (\frac{Q(t_k)}{t_k}, \mathbf{u}(k)) \right\} \quad (14)$$

where: u is vector of control variables, k is the optimization stages, B is Bellman equation.

A program on MATLAB 7.0 has been developed for the procedure. In this way an optimal profile of the control variable has been received.

The *L-lysine* with and without optimization is shown in **Fig.2**. **Fig.2** shows the increase of the *L-lysine* production with optimization by 23.34%. The substrate concentration is decreased witt 28% for all fermentation. This is shown in **Fig.3.**

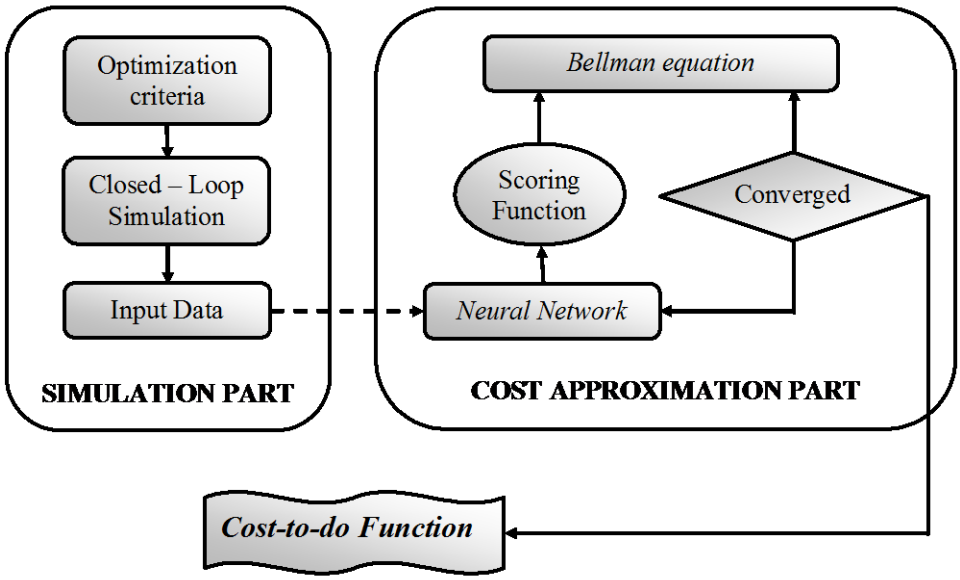

Fig.1. NDP algorithm.

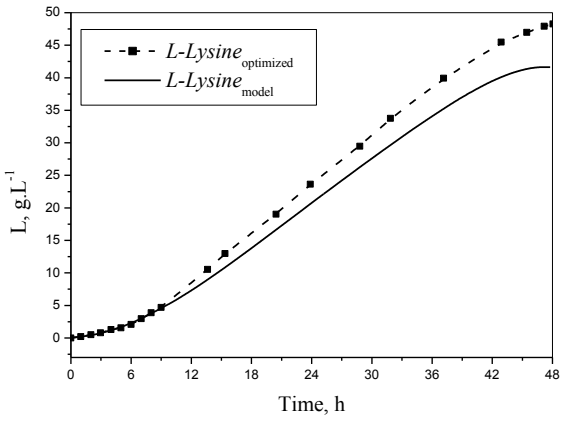

Fig.1. *L-lysine* concentration – model and optimized data.

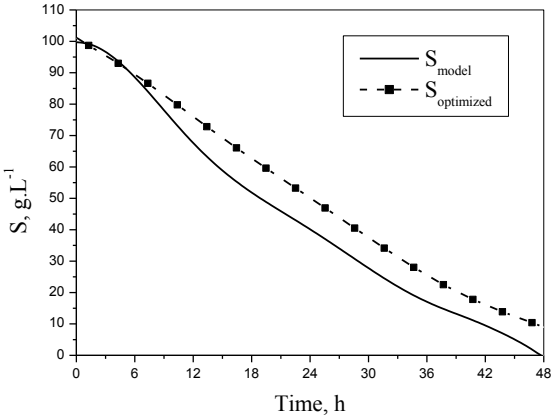

Fig.2. Substrate concentration before and after optimization.

CONCLUSION

1. In this paper we developed an optimal control of processes for an *L-lysine* fed-batch fermentation for receiving an optimal feed rate profile with Neuro-dynamic control.
2. The result shows that NDP is a convenient and easy to putting into practice application method

for optimization. Also it should be used for on-line fulfillment.

3. The results show that the *L-lysine* quantity is highly raised at the end of the process which is the desired criterion for process quality. Also the quantity of the substrate decrease, that is important problem in an industrial processes with expensive substrates.
4. At this point an optimal profile of the feeding rate is received and with this profile an optimized *L-lysine* production is obtained. However, the optimization algorithm does not have a feedback and it does not guarantee robustness to process disturbances. Therefore in future work we will apply suitable methods that will guarantee robustness to process disturbances.

ACKNOWLEDGEMENT

This work was published with financial support of the "Human Resources Development" Operational Programme, "Science–Business" Scheme grant BG051PO001-3.3-05/0001), Project: DO3-146/02.04.2014 Index: P-8-01/14.

REFERENCES

1. Anastassiadis, S. 2007. L-Lysine fermentation. Recent Patents on Biotechnology, 1, 11-24.
2. Bertsekas ,D., M. Tsitsiklis, 1996. Neuro-Dynamic Programming.Massachusetts: Athena Scientific Belmont.
3. Barto, A. G., S. J. Bradtke, S. P. Singh, 1995. Real-Time Learning and Control Using Asynchronous Dynamic Programming. Artificial Intelligence, 72, 81-138.
4. Driessens, K., S. Dzeroski, 2004. Integrating Guidance intro Relational Reinforcement Learning. Machine Learning, 57, 217-304.
5. Ilkova, T., M. Petrov, 2011. L-lysine Neuro-dynamic Control. Journal of Medical Research, 11, 55-60.
6. Kaisare, N. S., J. M. Lee, J. H. Lee, 2003. Simulation based Strategy for Nonlinear Optimal Control: Application to a Microbial cell Reactor. International Journal of Robust and Nonlinear Control, 13, 347-363.
7. Lee, J. M., N. S. Kaisare, J. H. Lee, 2006. Choice of Approximator and Design of Penalty Function for an Approximate Dynamic Programming Based Control Approach. Journal of Process Control, 16, 135-156.
8. Soni, A. 2002. A Multi-Scale Approach to Fed-Batch Bioreactor Control. University of Pittsburgh Press, PA.
9. Sutton, R. S., 1998. Learning to Predict by the Methods of Temporal Differences, Machine Learning, 3, 9-44.
10. Tosukhowong, T., J. H. Lee, 2009. Approximate Dynamic Programming Based Optimal Control Applied to an Integrated Plant with a Reactor and a Distillation Column with Recycle. AIChE Journal, 55, 919-930.