Neuro-dynamic Programming to Optimal Control of a Biotechnological Process

Tatiana Ilkova[1], Mitko Petrov[2*]

[1]*Faculty of Engineering*
South-West University "Neofit Rilski"
66 Ivan Mihaylov Str., 2700 Blagoevgrad, Bulgaria
E-mail: tanjasilkova@gmail.com

[2]*Department Bioinformatics and Mathematical Modelling*
Institute of Biophysics and Biomedical Engineering
Bulgarian Academy of Sciences
105 Acad. George Bonchev Str., 1113 Sofia, Bulgaria
E-mail: mpetrov1959@gmail.com

[*]*Corresponding author*

Received: July 04, 2024 **Accepted: December 12, 2024**

Published: December 31, 2024

Abstract: *Dynamic programming (DP) is an elegant way to solve problems related to optimization and optimal control of processes. DP, however, has one major drawback, namely the "curse of dimensionality". To overcome this shortcoming, an approach called neuro-dynamic programming (NDP) has been developed. This approach solves the "curse of dimensionality" problem of DP. For this purpose, a neural network is used in NDP, which ignores the poor results of the utility criterion. In this way, the time for solving the specific task is significantly shortened. In this work, an NDP algorithm is presented for the optimal control of a fed-batch biotechnological process for the production of L-lysine by the strain Brevibacterium flavum 22LD. Application of the NDP algorithm ensures maximum productivity of the L-lysine.*

Keywords: *Optimal control, Dynamic programming, Neuro-dynamic programming, Fed-batch biotechnological processes, L-lysine production, Brevibacterium flavum 22LD strain.*

Introduction

Biotechnological processes (BTP), as objects of optimal control (OC), are significantly different from processes occurring in a non-living nature. This is due to the interaction of processes of different nature. The growth of the microorganisms and metabolic transformations taking place in them is a multistage process with complex interrelated physiological, biochemical, genetic, physical, and other factors that can act simultaneously, influencing each other. BTP optimization and OC problems are usually associated with the presence of a complex, nonlinear dynamic model of the system. Difficulties arise even with offline OC, especially with a large number of control variables, as well as technological and other limitations of BTP [36, 52].

Many different methods have been used for optimization and optimal control of BTP over the years: dynamic programming (DP) [2], Pontryagin maximum principle [44, 46], green theorem [34, 49], calculus of variations [6, 42], etc.

In recent years, more and more new methods for optimization and OC have been introduced, such as fuzzy sets theory (FST) [8, 9, 11, 55, 56], genetic algorithms (GAs) [10, 35, 39, 40, 43, 53], artificial neural networks (ANN) [7, 38, 45, 54], Q-learning [8], model predictive control (MPC) [32], etc.

The calculus of variations [41] has been successfully applied to optimal control of BTP. Pontryagin's maximum principle has been used [46, 47] for optimal control in reactive processing for fine chemicals and polymers in stirred jacketed batch and semi-batch reactors.

DP is a powerful method for solving optimization problems, but it has several limitations that limit its use in solving low-dimensional problems [1]. To overcome these limitations, Luus [33] has proposed iterative dynamic programming through which a global optimum can be found for several nonlinear systems. The method has several limitations, the most important of which is that the optimal actions are a function of time only and are valid only for a fixed starting point.

Bertsekas and Tsitsiklis [5] propose another suboptimal method called neuro-dynamic programming (NDP), which allows to alleviate the "curse of dimensionality" of DP. The name NDP expresses the dependence of the method on both DP and the concept of artificial neural networks. Reinforcement learning is developed to address nonlinear and stochastic optimal control problems and is used in training the neural network in the NDP algorithm [4]. NDP is used to improve the performance provided by an initial suboptimal policy by iteratively approximating the *cost-to-go function*. The initial approximate *cost-to-go* function is further improved by an iterative procedure based on the Bellman equation. In this context, the role of simulation is twofold. First, by simulating the process under a judiciously chosen suboptimal policy and all possible operating parameters/disturbances, it provides a set of data points that define the relevant region in the state space. Second, the simulation provides the *cost-to-go* of moving under the suboptimal policy for each set of points at which to start an iteration of the Bellman equation.

The successful application of NDP has been shown in publications [30, 31, 35, 48, 50]. In number of our publications, DP, NDP, and rollout NDP have been successfully applied to optimal control of various batch and fed-batch BTP [12, 20, 23, 26, 29].

L-lysine is an essential amino acid, the content of which is high in animal protein, but relatively low in vegetable protein. It has been found that the average plant contribution of L-lysine (about 30 %) is not sufficient for animal organisms. An insufficient amount of L-lysine in feed reduces the biological value of feed doses, reduces weight gain and further productivity of farm animals, increases the amount of feed used per kilogram of grain, and reduces the quality of animal production. In addition to animal husbandry, L-lysine is used in the food industry, in medicine as a component of infusion solutions (blood substitutes), and as a fortifier of patent medicines [41, 42]. NDP was applied to the production of L-lysine by strain *Brevibacterium flavum* 22LD. To ensure feedback and stability of the optimal control profile, we have applied the model of predictive control [21].

In this work, we are going to apply the NDP algorithm for offline optimal control of fed-batch cultivation of strain *Brevibacterium flavum* 22LD for L-lysine production.

Materials and methods

Formulation of the optimization problem

In this work, the approach proposed by Kaisare et al. [30] is used. The problem of the dynamic optimization includes minimization of the following expression:

$$\min_{u_0,\ldots,u_{p-1}} \sum_{i=1}^{p-1} \varphi\ (x_i, u_i) + \bar{\varphi}(x_p) \quad (1)$$

with

path constraint: $g\ (x_i, u_i) \geq 0, \quad 0 \leq i \leq p-1;$
terminal constraint: $\bar{g}\ (x_p) \geq 0;$
model constraint: $\dot{x} = f(x, u).$

For a given initial state x_0 and a function constant input is

$$u(i) = u_i i \cdot h \leq \tau < (i+1) \cdot h,$$

where h is the sampling time, x_i represents the value of x at the stage i^{th} (i.e., $x(t)$ in $t = h.i$), ϕ is the single state cost function and $\bar{\varphi}$ is the terminal state cost function and they are defined on the multitude of the real numbers, and f is a continuous and differentiated function.

Such a problem may be solved in the context of finding an open-loop input trajectory offline for a fixed finite-time process.

Dynamic programming

Dynamic programming is related to multistage decision processes that correspond to multistage real processes. Very often, some single-step operations are artificially presented as multi-step by applying the DP method. DP is based on the "*Principle of optimality*", which was formulated by Bellman and states as follows [3]: "*The optimal strategy has such a characteristic that regardless of the initial state and the decision of the next stage it must determine the optimal strategy to a condition that is obtained from the decision taken at the initial stage*."

The DP involves the stage-wise calculation of the *cost-to-go* function to find the solution, not just for the specific x_0 but in general the x_0. In Eq. (1), the *cost-to-go* has been defined for each stage [1-3]:

$$J_i = \min_{u_{p-i},\ldots,u_{p-1}} \sum_{j=p-i}^{p-1} \varphi\ (x_j, u_j) + \bar{\varphi}(x_p). \quad (2)$$

In conformity with the "*Principle of the optimality*" of Bellman [3] "*The tail policy is optimal for the tail subproblem*", the calculation of the *cost-to-go* function can be done for each stage as follows:

$$J_i(x) = \min_u\ \varphi\ (x, u) + J_{i-1}[F_h(x, u)], \quad (3)$$

where $F_h(x, u)$ is the resulting state after integrating the differential equation for one sample interval with the starting state of x and constant input of u. Eq. (3) is calculated from $i = 1$ to $i = p$ with the initialization of $J_0 = \bar{\varphi}(x)$. Such as the pertinent terminal needs to be imposed at each stage. Once received, the *cost-to-go* function has an optimal decision for a general state x_0 according to the "*Principle of the optimum*".

The objective of DP is to calculate numerically the optimal cost function *J*. This computation can be done offline, i.e., before the real system starts operating. In very few cases we can solve the stage-wise optimization analytically to obtain a closed-form expression for the *cost-to-go* function. The conventional numerical approach to the problem involves gridding the state space, calculating and storing the *cost-to-go* for each grid point as one marches backward from the last stage to the first. An optimal policy, that is, an optimal choice of *u* for each *i*, is computed either simultaneously with *J*, or in real time by minimizing the right-hand side of Bellman's equation. It is well known, however, that for many important problems, the computational requirements of DP are overwhelming, mainly because of a very large number of states and controls (Bellman's "*curse of dimensionality*"). In such situations, a suboptimal solution is required.

Cost-to-go approximation

The traditional approach to solving the Bellman equation involves gridding the state space, solving the optimization Eq. (3) for each grid point, and performing the stage-wise optimization until convergence. Exhaustive sampling of state space can be avoided by identifying relevant regions of the space through simulation under judiciously chosen suboptimal policies. The "*policy improvement theorem"* states that a new policy that is greedy with respect to the *cost-to-go* function of the original policy is as good as or better than the original policy (the new policy is given when the moment value of the *cost-to-go* function is least, i.e., the new policy is defined by the expression [5]:

$$u(x) = arg\min_{\boldsymbol{u}} \{\varphi(x, \boldsymbol{u}) + J_i[F_h(x, \boldsymbol{u})]\} \quad (4)$$

and it is the improvement of the original policy. When the new policy is as good as the original policy, the above equation becomes the same as the Bellman optimality Eq. (3). The use of the Bellman equation to obtain an iterative improvement of *cost-to-go* approximator forms the crux of various methods like NDP [4, 5] and reinforcement learning (RL) [38, 54].

In this paper, the basic idea from NDP is used to improve the development of optimal control. A "*functional approximator"* is used to interpolate between these data points. An improvement is obtained through iterations of the Bellman equation. When the iterations converge, this offline-computed *cost-to-go* approximation is an optimal control calculation for the BTP.

NDP algorithm

NDP idea is [4, 5]:

1. Use an "*approximate*" reward function;
2. At the current state select a decision that maximizes the expected value of the current *"stage reward" + "approximate reward of the next state"*.

The NDP method is a suboptimal method that centers on the approximate evaluation of the optimal cost function *J*, possibly through the use of ANN and/or simulation [31]. For the description of the algorithm the next symbol was used: *J* represents *cost-to-go* values; $\tilde{J}(x)$ is a function approximation relating *J* to corresponding state *x*; $()^i$ is the iteration index for cost iteration loop, *k* is the discrete time. Then it can be written:

$\tilde{J}(k) \equiv \tilde{J}(x(k))$ and $\varphi(k) \equiv \varphi\ (x(k), u(k))$.

The following steps describe the NDP algorithm for approximation of the *cost-to-go* function in optimal control [35, 52]:

1. Perform simulations of the process with chosen suboptimal policies under all representative operating conditions;
2. The *cost-to-go* function is calculated using the simulation data for each state visited during the simulation, as each closed loop simulation yields data $x(0)$, $x(1)$, …, $x(N)$, where N is sufficiently large for the system to reach equilibrium. For each of these points, one-stage cost $\phi(k)$. *Cost-to-go* is the sum of single stage costs from the next point $J(k) = \sum_{i=k+1}^{N} \varphi(i)$;
3. Preparing the fit ANN to the data approximate the *cost-to-go* function – denoted as $\tilde{J}^0(x)$, as a smooth function of the states;
4. To improve the approximation, perform the following iteration (referred to as the cost iteration) until convergence: with the current *cost-to-go* approximation $\tilde{J}^i(x)$ is calculated $\tilde{J}^{i+1}(k)$ for the given points of x by the following equation:

$$\tilde{J}^{i+1} = \min_u \varphi\ (x,u) + \tilde{J}^i(F_h(x,u)) \quad (5)$$

which is based on the Bellman equation. Fit an improved *cost-to-go* approximate $\tilde{J}^{i+1}$ to x and $\tilde{J}^{i+1}(x)$.

5. Update of the policy. It may be necessary to update $(\min_u \varphi + \tilde{J}^i)$ with suboptimal policy for improvement of *cost-to-go* approximation.

In NDP the approximation procedure has been partially avoided through the simulation one. The simplified scheme is shown in Fig. 1 [22, 25].

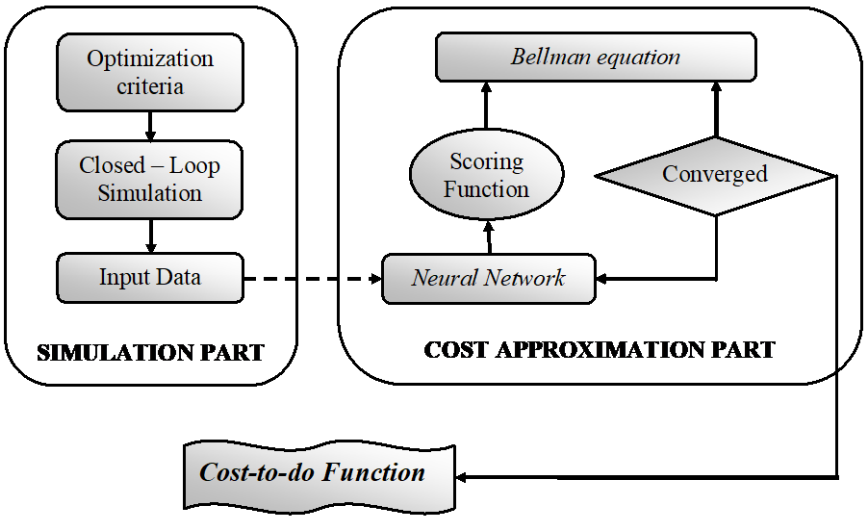


Fig. 1 Block-scheme of NDP algorithm

Assuming that one starts with a fairly good approximation of the *cost-to-go* (which would result from using a good suboptimal policy), the cost iteration should converge fairly fast-faster than the conventional stage-wise *cost-to-go* calculation.

Results and discussion

Experimental investigations

The experimental investigations are done in a 15 L bioreactor that is included in a System of Automatic Control (SAU). The SAU is flexible and includes control of the following parameters of the process: rotation speed, oxygen partial pressure, temperature, pH, foam level, gas flow rate, and flow rates of the main substance. The process is led in the next conditions [18, 24, 51]:

- Temperature, T = 30 °C;
- pH = 6.8-7.6;
- pO_2 = 20-30 %;
- Gas flow rate, Q_G = 60 L/h;
- Rotation speed, n = 450 min^{-1};
- Maximum bioreactor volume, V = 15 L.

The most efficient and cheap method for biosynthesis of L-lysine (in biologically active form) is the microbiological method by direct fermentation. The general scheme of the L-lysine biosynthesis is shown in Fig. 2 [51].

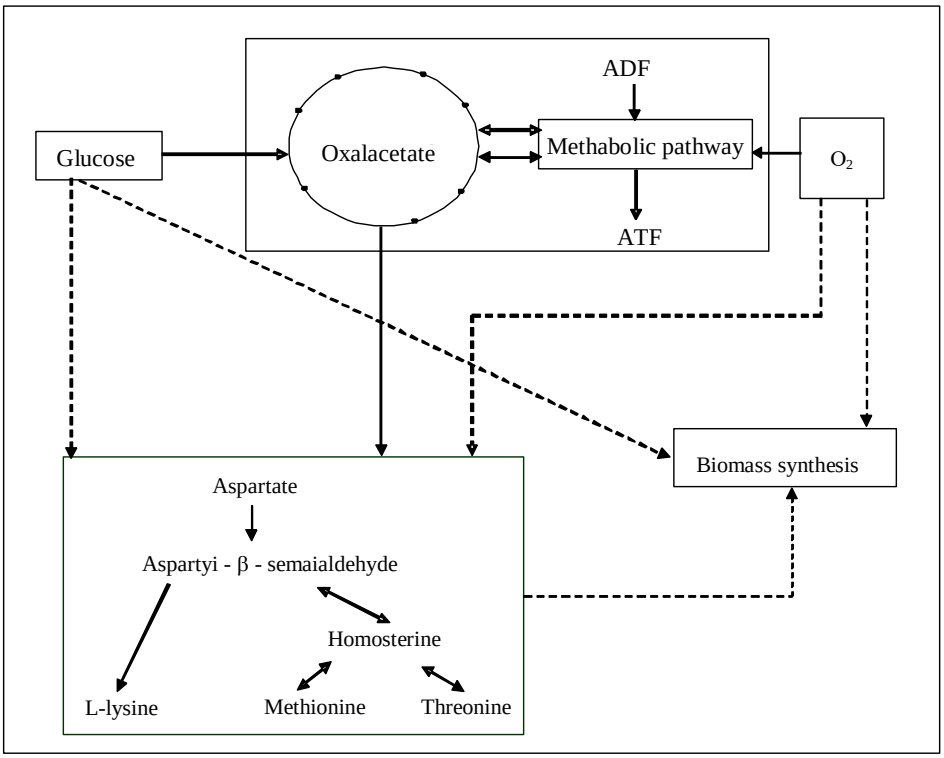


Fig. 2 General scheme of the L-lysine biosynthesis [51]

Process model of Brevibacterium flavum 22LD *for L-lysine production*

The model of the fed-batch processes includes the dependences between the concentrations of the basic variables of the process: cell mass concentration (bacteria *Brevibacterium flavum* 22LD), substrate concentration, L-lysine concentration, threonine concentration and oxygen concentration in the liquid phase. The model of the process has the following type [18, 24]:

$$\begin{aligned} \frac{dX}{dt} &= \mu X - \frac{F}{V}X \\ \frac{dS}{dt} &= \frac{F}{V}(S_0 - S) - K_1\eta\ X \end{aligned} \tag{6}$$

$$\begin{aligned} \frac{dL}{dt} &= K_2\mu X - \frac{F}{V}L \\ \frac{dTr}{dt} &= -K_3\mu X - \frac{F}{V}Tr \end{aligned} \tag{7}$$

$$\frac{dC_L}{dt} = k_l a(C_L^* - C_L) - K_4\mu X - \frac{F}{V}C_L \tag{8}$$

$$\frac{dV}{dt} = F \tag{9}$$

$$\mu = \mu \frac{S}{(K_S+S)} \frac{Tr}{(K_{Tr}+Tr)} \frac{C_L}{(K_C+C_L)_{max}} \tag{10}$$

$$\eta = \frac{\mu}{Y} + K_m \tag{11}$$

where: X is the biomass concentration, g/L; S – substrate concentration, g/L; S_0 – initial substrate concentration, g/L; L – L-lysine concentration, g/L; Tr – threonine concentration, g/L; C_L – dissolved oxygen concentration, g/L; C_L^* – mean oxygen concentration, g/L; t – process time, h; V – volume of the bioreactor, L; F – feeding rate, L/h; $k_l a$ – volumetric oxygen mass-transfer coefficient, h^{-1}; $K_{1, ..., 5}$ – kinetic coefficients, g/g; K_i – inhibition coefficient, g/g; K_S – Monod's saturation constant for substrate, g/g; K_{Tr} – Monod's saturation constant for threonine, g/g; K_C – Monod's saturation constant for oxygen, g/g; K_m – kinetic constant, g/g; Y – yield coefficient, g/g; μ – specific growth rate, h^{-1}; μ_{max} – maximal value of specific growth rate, h^{-1}; η – specific consummation rate of substrate, h^{-1}.

The initial conditions are [51]:

$X(0) = 3.0$, g/L; $S(0) = 100.0$, g/L; $L(0) = 0.0$, g/L; $Tr(0) = 80.0$, g/L,
$C_L^* = 6.1 \times 10^{-3}$, g/L; V(0) = 10 L.

The coefficients in the model are [17, 18]:

$\mu_{max} = 0.39$, h^{-1}, $K_1 = 0.893$, $K_2 = 0.123$, $K_3 = 6.545$, $K_4 = 41.947$, $C_L^* = 6.1 \times 10^{-3}$, g/L;
$K_S = 0.452$, $K_C = 0.0123$, $K_m = 0.345$, $K_{Tr} = 23.342$, $Y = 0.489$, $k_l a = 139.68$, h^{-1}.

Formulation of the optimization problem for optimal control of Brevibacterium flavum 22LD

As it is well-known for the fermentation process, relatively little change in the speed of feed can cause the process to switch over toward an undesired stability state (especially steep disturbance in F). The control objective is, therefore, to drive the reactor from the low product steady state to the desirable high product yield state. It may be viewed as a step change in the set-point at time $t = 0$ from the low product to the high product steady state.

The optimization criterion has accepted the value of the functional $L(t)$ at the end of the process ($t_f = 48$ h) which means the quantity forms L-lysine after 48-hour fermentation.

The criterion of quality has a type:

$$\max_u J = L(t_f), \tag{12}$$

where $L(t_f)$ is the L-lysine concentration in final time, g/L, t_f – final time, h.

For the optimal control variable, the feeding rate (F) is taken. The optimal problem aims to find such a profile of the optimal control F that maximizes the criterion of quality (12). Then, the criterion (12) can be written as:

$$max\, J = \sum_{k=0}^{N} \int_{t_k}^{t_{k+1}} L(t)dt. \tag{13}$$

For this aim the problem has been discretisation in the interval $[t_0, t_f]$ – the initial and final time of the fermentation, where the monotonous crescent row has formed from the type: t_0, t_1, t_2, …, t_f. The total time is 48 hours. Therefore the optimisation criterion is:

$$J = \sum_{k=0}^{N} f_k(L_k, F_k)\Delta t, \tag{14}$$

where $L_k \equiv L(t_k)$ – L-lysine concentration; $F_k = F(t_k)$, for $\kappa = 0, ..., N$; $\Delta t = 6$ h, and $N = 48$ h.

The restriction of feeding flow rate $F \in [0 \div 3]$ L/h for the optimal control variable is forced, it is:

$$\sum_{i=1}^{N} F_i = F \text{ and } \boldsymbol{V_i} = \sum_{i=1}^{N} \boldsymbol{F_i} \cdot \boldsymbol{h_i}. \tag{15}$$

NDP for optimal control of Brevibacterium flavum 22LD

In this part of our work, by applying *NDP*, the optimal feed flow rate (F) of a fed-batch process is determined for the *Brevibacterium flavum* 22LD model (6)-(11) to increase L-lysine concentration at the end of the process, criterion J, Eq. (12).

The simulation-based approach involves the computation of the converged *profit-to-go* approximation *offline*.

The optimal control variable is the feeding rate (F). The following values of F are examined: $F_k \in [0.1,\ 0.2,\ 0.3,\ 0.5,\ 0.7]$ L/h, that covers the possible range of variation.

Improvement to the *cost-to-go* function is obtained through the iterations of the Bellman equation. This method is known as a value iteration.

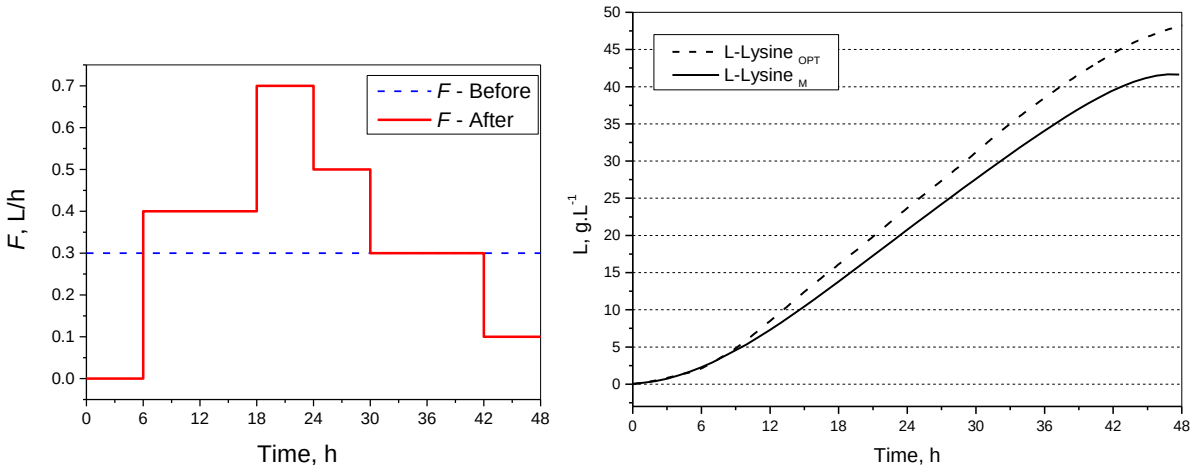
A functional approximation relating the *cost-to-go* function with the augmented state is obtained by using ANN with three layers. Five values of (F) are examined, that can cover the possible range of variation. For each of the parameter values, the reactor is started at three different $L(0)$ values around the low L-lysine steady state. We have obtained 100 data points for each run. Thus a total of 1200 data points have been obtained. A functional approximation relating *cost-to-go* with the augmented state is obtained by using a neural network with five hidden nodes, six input, and one output node. The ANN shows a good fit with the mean square error of 10^{-3} after training for 1000 epochs. Improvement to the *cost-to-go* function is obtained through iterations of the Bellman equation. This method is known as cost iteration (or value iteration). The solution of the one-stage-ahead cost plus *cost-to-go* problem results in improvements in the cost values. 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 etc., until convergence. Cost is said to be "converged" if the sum of absolute error is less than 5% of the maximum cost. The cost has converged in 4 iterations for our system. The converged *cost-to-go* function is used in solving the *one-stage-ahead* problem:

$$\boldsymbol{u}(k) = arg \max_{\boldsymbol{u}(k)} \{f[L(t_k), \boldsymbol{u}] + \tilde{B}^6[L(t_k), \boldsymbol{u}(k)]\}, \tag{16}$$

where $\boldsymbol{u}$ is the vector of the control variable (F), k is the optimization stages; B^6 is the Bellman equation; $L(t_k)$ represents *cost-to-go* values for stage k; the superscript 6 represents the iteration index.

An algorithm under MATLAB R2019b program was developed for optimal process control in the production of L-lysine by *Brevibacterium flavum* 22LD. All calculations are done on an octal core AMD FX-8320, 3500 MHz processor, 32 GB Memory (RAM), with a Windows XP Pro (32 bit) operating system.

The optimal profile of feed flow rate F and L-lysine concentration before and after optimization is shown in Fig. 3.



A) B)

Fig. 3 Optimal profiles of L-lysine before and after optimization with NDP:
A) feed flow rate F; B) concentration of L-lysine.

By using the NDP algorithm for optimal control of the process, L-lysine production increases more than 13% at the end of the process, which is shown in Fig. 3B.

Discussion

In the monograph [52], the differences between different BTPs as objects for modelling and control are examined. The application of the DP and NDP for OC is also considered. DP is applied in one and multiple cases for the industrial processes OC of fed-batch for *E. coli*, L-lysine, and lactose oxidation from a natural substratum from a strain *Kluyveromyces marxianus var. lactis* MC 5. The developed investigations have shown the inclusion of more control variables raises the process effectiveness. The application of the DP method for optimal control of wit multi criteria decreases the substrate feeding rate values. This is a good reason for expensive substrates. The second monograph part presents an application of NDP for OC for the same processes. NDP gives a more flexible procedure and decreasing of computing time. These advantages facilitate its online applications.

In [36], Chapter 6 "Neuro-dynamic, rollout, and model predictive control of fermentation processes", some of the methods related to DP, such as NDP and rollout algorithms, are discussed. The possibilities of MPC in the optimal management of the fed-batch fermentation process of the lactose oxidation from a natural substrate by strain *Kluyveromyces marxianus var. lactis* MC5 is described.

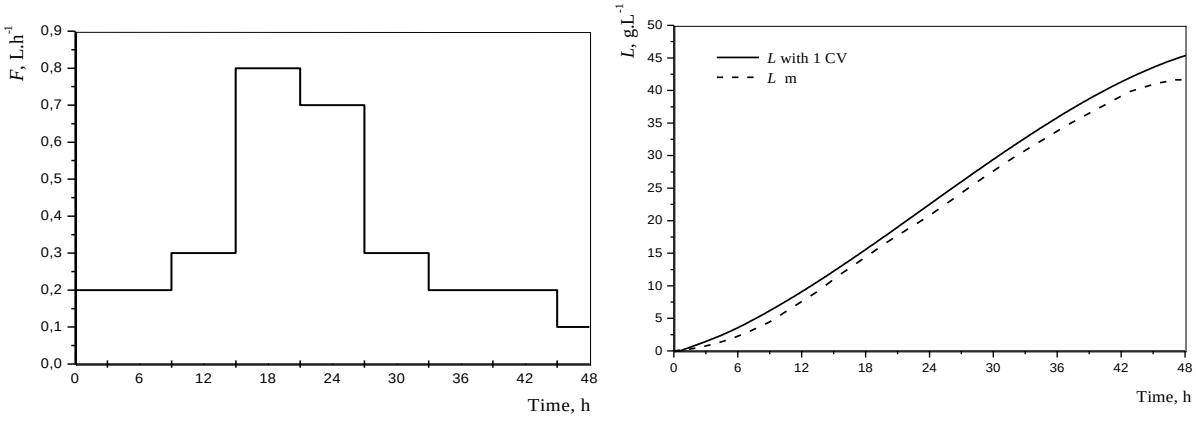
In a publication by Ilkova and Petrov [24], it is proved that the process does not need to be continued for more than 48 hours. We have studied the process up to the 54th hour. The obtained results have shown that after the 48th hour the process stops and its continuation is economically unprofitable. Therefore, the fixed right end of the process at the 48th hour is appropriate.
In [19] multi-objective optimization of an aerobic fed-batch cultivation of *Brevibacterium flavum* 22LD for the L-lysine production is performed. The single-objective functions reflect the L-lysine production and the degree of the substrates utilizing – glucose and threonine. The multiple-objective optimization is transformed into a standard problem for optimization with a single-objective function. The obtained results from the paper have shown that multiple-objective optimization is a more complex approach minimizing the risk in the procedure of making decisions and maximizing the formulated objective.

In [17], an efficient fuzzy optimization algorithm is developed and applied to BTP for the production of L-lysine by strain *Brevibacterium flavum* 22LD. The developed "*flexible*" model better reflects the influence of the kinetic parameters of the process on the optimization criterion. For the first time, in this process, the initial conditions are optimized using fuzzy set theory. With the initial conditions thus determined, a dynamic optimization is performed using fuzzy set theory.

The application of DP, NDP and rollout methods for optimization and optimal control for different BTPs is shown in publications [11, 13-16, 20, 21, 23, 36]. The application of the same methods and fuzzy set theory for optimization and optimal control in the production of L-lysine by *Brevibacterium flavum* 22LD has been shown in publications [12, 19, 22, 24, 27-29].

In [52], Chapter 6 "Optimal control of fermentation processes, optimization criteria, constraints, and final conditions and the main methods for optimal control of BTP, such as Pontryagin's maximum principle, green theorem, calculus of variations, dynamic and neuro-dynamic programming, are discussed in details.

As an illustration Fig. 4 and Fig. 5 show an application of the DP for optimal control of a BTP for L-lysine production in a one-dimensional (Fig. 4) and multi-dimensional case (Fig. 5).



A) B)

Fig. 4 Optimal control with DP with single control variable:
A) optimal profile of the feed flow rate, B) L-lysine quantity before and after optimization.

In the multidimensional case, two more control variables are added to the feeding rate *F*: rotation speed (*n*) and gas-flow rate (*Q*). Optimization criteria are the maximum concentration

of L-lysine at the end of the process. For the one-dimensional case, the optimal control variable is the feed flow rate *F*.

Through this profile application, the L-lysine quantity has been increased by 13.25% at the end of the process (Fig. 4B).

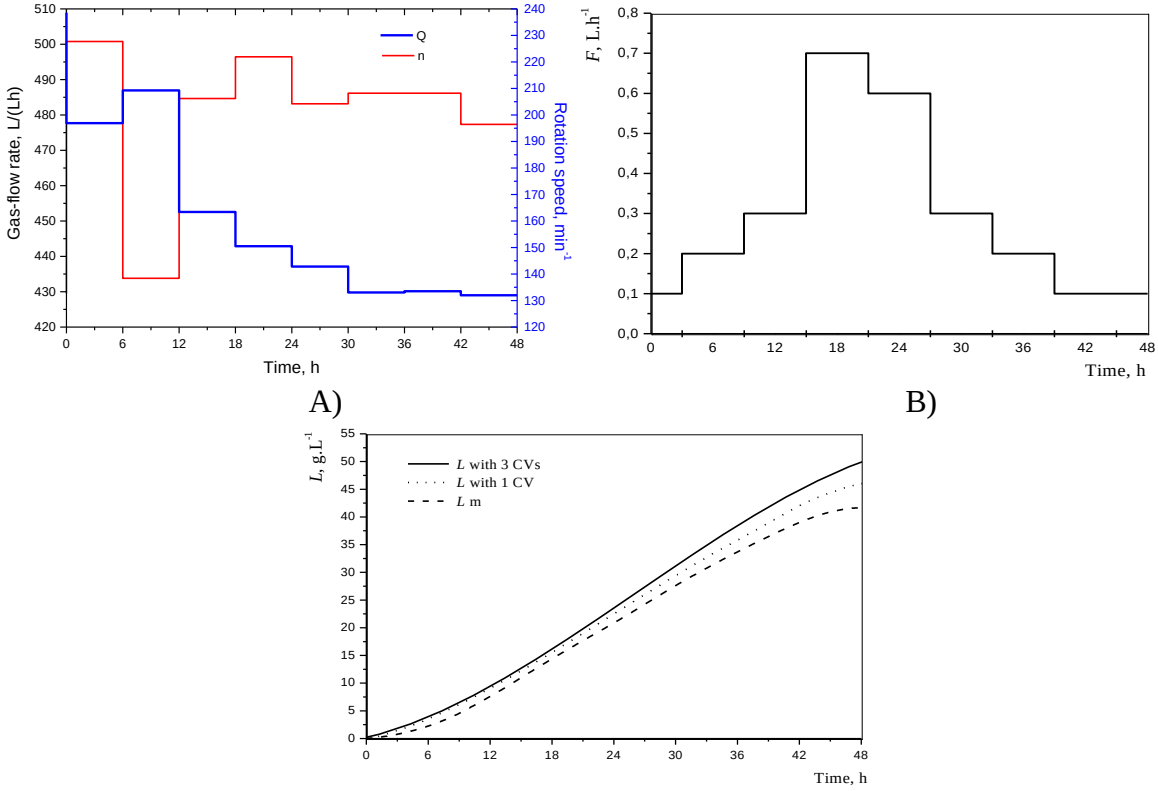
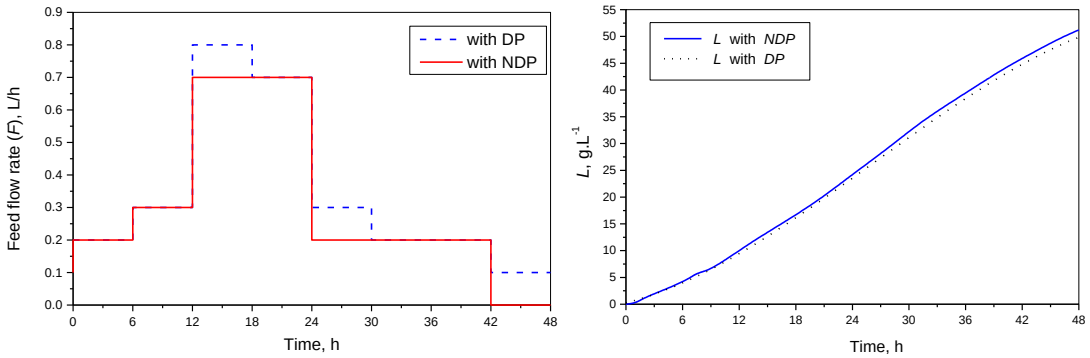


Fig. 5 Concentration of L-lysine before and after optimal control with DP with single and multi control variables:
A) optimal profiles of the impeller speed and gas flow rate; B) optimal profile of *F*;
C) L-lysine concentration before and after optimization with 3 CVs.

The received optimal profiles of the control variables are shown in Fig. 5A and Fig. 5B. After the application of OC the quantity of L-lysine has been increased by 22.25% at the end of the process (Fig. 5C). The comparison between the method in one-dimensional and multi-dimensional case shows that the L-lysine production at the end of the process is 9.0% higher in multi-dimensional case from the method application (Fig. 4B and Fig. 5C).

The feeding rate obtained by to DP and NDP approach is presented in Fig. 6A. The L-lysine concentration received with DP and NDP is shown in Fig. 6B.

In Fig. 6A can be seen that the used feeding quantity, optimized by the NDP method, is less than the used one optimized by the DP method. Following this way, it could be concluded that better substrate utilization gives rise to the production price decrease. An increase in the concentration of L-lysine at the end of the process by more than 4% was achieved using NDP, compared to the DP method (Fig. 6B).



A) B)

Fig. 6 Concentration of L-lysine after optimal control with DP and NDP:
A) feeding flow rate received with DP and NDP;
B) L-lysine concentration before and after optimization with DP and NDP.

The comparison between the DP and NDP methods, concerning the time consumed for the optimization development, has shown that the necessary time for the optimal control performance due to the NDP approach is less than the necessary one, consumed from the DP method. The NDP method consumes about 130.0 s for the optimization procedure, while the DP optimization one finishes after 1067.0 s. Finally, it could be concluded that the developed method finds a successful application in online optimization procedures of any problems.

Conclusions

The traditional approach for solving Bellman's equation involves gridding of the state space, solving the optimization for each grid point, as well as performing the stage wise optimization until convergence is reached. The comprehensive sampling of state space can be avoided by identifying the relevant regions of the state space through simulation under judiciously chosen suboptimal policies, which are presented using NDP methods. The proposed method is particularly simple to implement and can be applied for on-line optimization.

Realization of such an optimal control approach combined with advanced control techniques (artificial neural networks with classical optimization method) in practice can lead to value and elaboration time reduction in the laboratory fed-batch bioprocesses and not only that but also to an elaboration time reduction technique for optimal control.

NDP is usefully applied for optimal control of a fed-batch process predicting L-lysine by strain *Brevibacterium flavum* 22LD and the result of the optimization is an increase in the product L-lysine at the end of the process.

References

1. Bellman R. (1954). Some Problems in the Theory of Dynamic Programming, Econometrica: Journal of the Econometric Society, 22(1), 37-48.
2. Bellman R. (1957). Dynamic Programming, Princeton University Press, New Jersey, USA.
3. Bellman R., M. Giertz (1973). On the Analytic Formalism of the Theory of Fuzzy Sets, Information Sciences, 5, 149-156.
4. Bertsekas D. (2000). Dynamic Programming and Optimal Control, 2nd Ed., Athena Scientific, Belmont, MA, USA.
5. Bertsekas D., J. N. Tsitsiklis (1996). Neuro-dynamic Programming, Athena Scientific, Belmont, MA, USA.

6. Carrillo-Ureta G. E., P. D. Roberts, V. M. Becera (2003). Optimal Control of a Fermentation Process, Proceedings of the European Control Conference (ECC), Cambridge, UK, 1947-1952.
7. Eeranna B. C., S. Rao, G. P. Reddy (2014). Artificial Neural Network Based Modeling and Control of Bioreactor, International Journal of Engineering Research & Technology, 3(8), 117-122.
8. ElMezain A., M. Saleh, J. Zhang, A. Soliman, et al. (2019). Optimal Control of Batch Processes Via a Deterministic Q-learning Method, arXiv, Computer Science (Systems and Control), arXiv:1904.03654.
9. Filev D., P. Angelov (1992). Fuzzy Optimal Control, Fuzzy Sets and Systems, 47(2), 151-156.
10. Goldberg D. E. (1989). Genetic Algorithms in Search, Optimisation, and Machine Learning, Addison Wesley, Reading, MA.
11. Hristova T., S. Tzonkov (1998). Modeling and Optimization of Fed-batch Fermentation Process, Chemical and Biochemical Engineering Quarterly, 12(4), 213-215.
12. Ilkova, T. (2005). Using Dynamic and Rollout Neuro-dynamic Programming for Static and Dynamic Optimization of a Fed-batch Fermentation Process, International Journal Bioautomation, 3, 29-35.
13. Ilkova T. (2006). Neuro-dynamic Optimal Control: An Application of a Fed-batch Whey Fermentation by Strain *Kluyveromyces lactic* MC 5 in a Stirred Tank Bioreactor, Proceedings of the International Conference Automatics and Informatics, Sofia, Bulgaria, 107-110.
14. Ilkova T. (2007). Applications of the Neuro-dynamic Programming for Optimal Control of Fed-batch Fermentation Processes, Proceedings of the International Conference Automatics and Informatics, Sofia, Bulgaria, Vol. II, III-47-III-50.
15. Ilkova T. (2008). Application of Dynamic and Neuro-dynamic Programming for Optimal Control of Fermentation Processes, Avangard Prima, Sofia, Bulgaria. (In Bulgarian)
16. Ilkova T. (2008). Rollout Optimal Control of Fed-batch Fermentation Process, Proceedings of the International Symposium Bioprocess Systems, I.53-I.62.
17. Ilkova T., M. Petrov, S. Tzonkov, U. Viesturs (2004). Fuzzy Optimization of Initial Conditions of the Biosynthesis L-lysine, Proceedings of the International Symposium Bioprocess Systems, Sofia, Bulgaria, I.29-I.31.
18. Ilkova T., M. Petrov, D. Bertsekas (2006). Neuro-dynamic Optimal Control of a L-lysine Fed-batch Fermentation, Biotechnology & Biotechnological Equipment, 20(3), 204-207.
19. Ilkova T., M. Petrov (2007). Multi-objective Optimization of the L-Lysine Production, Proceedings of the International Conference Automatics and Informatics, Sofia, Bulgaria, Vol. II, III-69-III-72.
20. Ilkova T., M. Petrov (2008). Dynamic and Neuro-dynamic Optimization of a Fed-batch Fermentation Process, Lecture Notes in Artificial Intelligence, 5253, 365-369.
21. Ilkova T., M. Petrov (2008). Multiobjective Neuro-dynamic Optimization and Model Predictive Control for Receiving of Ecological Clean and Wasteless Technology at the L-lysine Production, Journal of International Scientific Publication: Ecology & Safety, 2, 480-494.
22. Ilkova T., M. Petrov (2008). Neuro-Dynamic Control Algorithm for a Fed-Batch Fermentation Process, Proceedings of the International Bulgarian-Greek Conference Computer Science, Kavala, Greece, Part I, 127-132.
23. Ilkova T., M. Petrov (2009). Rollout Optimization of Lactose Oxidation for Receiving of Ecological Clean and Wasteless Technology at Cheese Product, Journal of International Scientific Publication: Ecology & Safety, 3, Part 1, 466-475.

24. Ilkova T., M. Petrov (2011). L-lysine Neuro-dynamic Optimal Control, Global Journal of Medical Research, 11(4), 63-68.
25. Ilkova T., M. Petrov (2024). Neuro-dynamic, Rollout and Model Predictive Control for Optimal Control of Fermentation Processes, Contemporary Mathematics, 5(3), 3790-3803.
26. Ilkova T., M. Petrov, S. Tzonkov (2014). An Approach for Optimal Control of Fed-batch Fermentation Process with Multi Control Variables, Biotechnology & Biotechnological Equipment, 18(2), 201-205.
27. Ilkova T., S. Tzonkov (2002). Optimization of Fed-batch Fermentation Process by Dynamic Programming, Proceedings of the International Symposium Bioprocess Systems, Sofia, Bulgaria, I.22-I.25.
28. Ilkova T., S. Tzonkov (2003). Neuro-dynamic Programming: Application to a Fermentation Process, Proceedings of the International Symposium Bioprocess Systems, Sofia, Bulgaria, I.29-I.32.
29. Ilkova T., S. Tzonkov (2004). Optimal Control of a Fed-batch Fermentation Process by Neuro-dynamic Programming, International Journal Bioautomation, 1, 57-66.
30. Kaisare N., J. M. Lee, J. H. Lee (2002). Simulation-based Optimization for Nonlinear Optimal Control, Proceedings of the IFAC World Congress, Barcelona, Spain, 234-368.
31. Kim K. D., A. J. Ho, K. S. Lee, D. R. Yang (2005). The Application of Control Using Neuro-dynamic Programming with a Feature Map, IFAC 16th Triennial World Congress, Prague, Czech Republic, 38(1), 995-1000.
32. Lee J. H., B. Cooley (1997). Recent Advances in Model Predictive Control, Proceedings of the 5th International Conference on Chemical Process Control – V: Assessment and New Directions for Research, California, AIChE Symposium Series, 201b-216b.
33. Luus R. (2000). Iterative Dynamic Programming, Taylor and Francis Group, Chapman & Hall: New York, USA.
34. Ohno H., E. Nakanishi, T. Takamatsu (1976). Optimal Control of a Semibatch Fermentation, Biotechnology and Bioengineering, 18(6), 847-864.
35. Pedrozo H. A., A. M. Dallagnol, C. E. Schvezov (2021). Genetic Algorithm Applied to Simultaneous Parameter Estimation in Bacterial Growth, Journal of Bioinformatics and Computational Biology, 19(1), 2050045.
36. Pencheva T., M. Petrov, T. Ilkova, O. Roeva, et al. (2006). Bioprocess Engineering, Avangard Prima, Sofia, Bulgaria.
37. Petrov M., T. Ilkova (2005). Fuzzy Optimization of the Biosynthesis of L-lysine, Chemical and Biochemical Engineering Quarterly, 19(3), 283-289.
38. Petsagkourakis P., I. O. Sandoval, E. Bradford, D. Zhang, et al. (2020). Reinforcement Learning for Batch Bioprocess Optimization, Computers & Chemical Engineering, 133, 106649.
39. Roeva O. (2012). Optimization of *E. coli* Cultivation Model Parameters Using Firefly Algorithm, International Journal Bioautomation, 16(1), 23-32.
40. Roeva O., S. Fidanova (2012). Application of Genetic Algorithms and Ant Colony Optimization for Modelling of *E. coli* Cultivation Process, Real-World Applications of Genetic Algorithms, InTech Open, 261-282.
41. Roeva O., S. Tzonkov (2005). Optimal Feed Rate Control of *Escherichia coli* Fed-batch Fermentation, International Journal Bioautomation, 2, 30-36.
42. Sagan H. (1971). Calculus of Variations and Optimal Control Theory, Journal of the Franklin Institute, 291(4), 305-313.
43. Slavov T., O. Roeva (2012). Application of Genetic Algorithm to Tuning a PID Controller for Glucose Concentration Control, WSEAS Transactions on Systems, 11(7), 223-233.

44. Smets I. Y., J. E. Claes, E. J. November, G. P. Bastin, et al. (2004). Optimal Adaptive Control of (Bio)chemical Reactors: Past, Present and Future, Journal Process Control, 14(7), 795-805.
45. Sorensen O. (1995). Neural Networks for Optimal Control, Proceedings of the IFAC System Structure and Control, Nantes, France, 361-366.
46. Staniskis J. (1984). Optimal Control of Biotechnological Processes, Mokslas, Vilnius, Lithuania. (In Russian)
47. Stojkovic M. (2023). Optimal Control in Reactive Processing for Fine Chemicals and Polymers in Stirred Jacketed Batch and Semi-batch Reactors: A Bibliography, Fine Chemical Engineering, 4(2), 203-241.
48. Su Z. C., A. H. F. Chow, N. Zheng, Y. P. Huang, et al. (2020). Neuro-dynamic Programming for Optimal Control of Macroscopic Fundamental Diagram Systems, Transportation Research Part C: Emerging Technologies, 116, 102628.
49. Takamatsu T., I. Hashimoto, S. Shioya, K. Mizuhara, et al. (1975). Theory and Practice of Optimal Control in Continuous Fermentation Process, Automatica, 11(2), 141-148.
50. Trelea I., M. Titica, G. Corrieu (2004). Dynamic Optimisation of the Aroma Production in Brewing Fermentation, Journal Process Control, 14(1), 1-16.
51. Viesturs U., A. Kuznetzov, V. Savenkov (1986). Fermentation Systems, Zinatne: Riga, Latvia, 241-274. (in Russian)
52. Viesturs U., I. Simeonov, T. Pencheva, J. Vanags, et al. (2010). Contemporary Approaches to Modelling, Optimisation and Control of Biotechnological Processes, Prof. M. Drinov Academic Publishing House, Sofia, Bulgaria.
53. Wang Q. J. (1997). Using Genetic Algorithms to Optimise Model Parameters, Environmental Modelling & Software, 12(1), 27-34.
54. Yamaguchi A., C. Atkeson (2016). Neural Networks and Differential Dynamic Programming for Reinforcement Learning Problems, Proceeding of the IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 5434-5441.
55. Zadeh L. A. (1965). Fuzzy Sets, Information and Control, 8, 338-353.
56. Zimmermann H. J. (1983). Fuzzy Mathematical Programming, Computers & Operations Research, 10(4), 291-298.

Assoc. Prof. Tatiana Ilkova, Ph.D.
E-mail: tanjasilkova@gmail.com

Tatiana Ilkova was born in 1970. She received M.Sc. Degree in Engineering of Biotechnology (1995) and Ph.D. Degree (2008) from the Technical University – Sofia. At present, she is an Associate Professor at the Faculty of Engineering, South-West University "Neofit Rilski", Bulgaria. Her scientific interests are in the fields of bioprocess systems, modelling, optimization and optimal control of bioprocesses and modelling of ecological systems. She has about 140 scientific publications with more than 50 known citations.

Assoc. Prof. Mitko Petrov, Ph.D.
E-mail: mpetrov1959@gmail.com

Mitko Petrov (born 1959) graduated from the Technical University – Sofia in 1987 as a Mechanical Engineer. He has worked as an Associate Professor at the Institute of Biophysics and Biomedical Engineering – Bulgarian Academy of Sciences since 1988. He achieved his Ph.D. Degree in 2003. He is an Associate Professor since 2007. His scientific interests are in the fields of bioprocess systems, modelling, optimization and optimal control of bioprocesses, modelling and optimization of apparatus of bioprocess systems, and modelling of ecological systems. He has about 190 scientific publications with more than 160 known citations.