Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 103748, 8 pages
http://dx.doi.org/10.1155/2013/103748

Research Article

Parameter Identification of Anaerobic Wastewater Treatment Bioprocesses Using Particle Swarm Optimization

Dorin Sendrescu

Department of Automatic Control, University of Craiova, A.I. Cuza 13, 200585 Craiova, Romania

Correspondence should be addressed to Dorin Sendrescu; dorins@automation.ucv.ro

Received 25 January 2013; Accepted 30 June 2013

Academic Editor: Carlo Cattani

This paper deals with the offline parameters identification for a class of wastewater treatment bioprocesses using particle swarm optimization (PSO) techniques. Particle swarm optimization is a relatively new heuristic method that has produced promising results for solving complex optimization problems. In this paper one uses some variants of the PSO algorithm for parameter estimation of an anaerobic wastewater treatment process that is a complex biotechnological system. The identification scheme is based on a multimodal numerical optimization problem with high dimension. The performances of the method are analyzed by numerical simulations.

1. Introduction

It is well known that the biotechnology is one of the fields that over the last decades have a high development. Therefore, due to its advantages, the control of industrial bioprocesses has been an important practical problem attracting wide attention. Biotechnology applications can be found especially in agriculture, in food industry, in medicine and pharmaceutical processes, in waste treatment processes, and so forth. A frequent and important challenge in control of such living processes is finding an accurate model of the system. The bioprocesses are highly nonlinear, and their kinetic parameters are usually badly or inadequately known [1]. This problem becomes of great importance in complex systems where critical instability of the process must be avoided. Parameters characterizing the internal behavior of biotechnological systems are usually not directly accessible to measurement. Their measurement is usually approached indirectly as a parameter estimation problem [2]. In this paper a dynamic model describing the internal structure of the system is formulated, and an algorithm based on PSO for parameter estimation is designed.

In recent years, a progress has been made in the area of continuous-time system identification [3]. Even if the most physical systems are naturally continuous, a much more attention has been paid to parameter estimation of discrete-time systems, mainly because they are better suited for numerical implementations. Continuous-time identification makes possible a more direct link to the physical properties and operation of the underlying systems and the direct estimation of physical parameters which have a clear significance. The most common approach for parameter estimation of linear or nonlinear systems is the use of prediction-error identification methods (PEM) [4]. In this category falls the well-known least squares methods or the maximum likelihood methods. In this approach, identification consists in minimization of a scalar-valued function of the model parameter. In general, this function cannot be minimized by analytical methods so the solution has to be found by iterative, numerical techniques. There is an extensive literature on such numerical problems. In classical approach the most used procedures are the quasi-Newton methods and interior point algorithms. The main drawback of these nonlinear parameter optimization techniques is that they are often unreliable; for example, they give no guarantee of converging to a true minimum. The increasing computational power of personal computers and microcontrollers allowed the implementation of several optimization algorithms inspired from natural phenomena. Examples of these algorithms include the Simulated Annealing [5], Genetics Algorithms (GA) [6], or Ant Colony

Optimization [7] algorithms. Particle Swarm Optimization (PSO) [8] is among these nature inspired algorithms. It is inspired by the ability of birds flocking to find food that they have no previous knowledge of its location. Every member of the swarm is affected by its own experience and its neighbors' experiences. Although the idea behind PSO is simple and can be implemented by two lines of programming code, the emergent behavior is complex and hard to completely understand [9, 10].

The most important approaches for the yield and kinetic coefficients estimation of biotechnological systems make use of the state transformations based on the general structure [11]. In this paper we propose an identification method based on particle swarm optimization techniques for these classes of biotechnological systems considering that the unknown parameters can appear in rational relations with measured variables. The paper is organized in the following way. The nonlinear dynamical model of an anaerobic wastewater treatment bioprocess is given in Section 2. Section 3 presents the identification algorithm using the particle swarm optimization techniques. Some numerical simulations are presented in Section 4 and conclusions in Section 5.

2. Nonlinear Dynamical Model of Anaerobic Wastewater Treatment Bioprocesses

A process that takes place in a bioreactor can be described as a set of m biochemical reactions involving n components (with $n \geq m$). The global dynamics can be represented by the following dynamical state-space model [1]:

$$\frac{d\xi}{dt} = K \cdot \varphi(\xi, t) - D\xi + F - Q, \tag{1}$$

where $\xi \in \Re^{n\times 1}$.

This model describes the behavior of an entire class of biotechnological processes and is referred to as the general dynamical state-space model of this class of bioprocesses [9]. In (1), the term $K \cdot \varphi(\xi, t)$ is the rate of consumption and/or production of the components in the reactor; that is, the reaction kinetics and the term $-D\xi + F - Q$ represents the exchange with the environment. The strongly nonlinear character of the model (1) is given by the reaction kinetics. In many situations, the yield coefficients, the structure, and the parameters of the reaction rates are partially known or unknown. Many of the evolved control methods for these kinds of systems—like model predictive control and robust or adaptive control—are based on good initial estimates of the yield and kinetic parameters.

Anaerobic digestion is a multistage biological wastewater treatment process whereby bacteria, in the absence of oxygen, decompose organic matter to carbon dioxide CO_2, methane CH_4, and water [12]. Four metabolic paths can be identified in this process: two for acidogenesis and two for methanation (see Figure 1). In the first acidogenic path, glucose (or another complex substrate) from the wastewater is decomposed into volatile fatty acids (acetates, propionic acid), hydrogen H_2, and inorganic carbon by acidogenic bacteria.

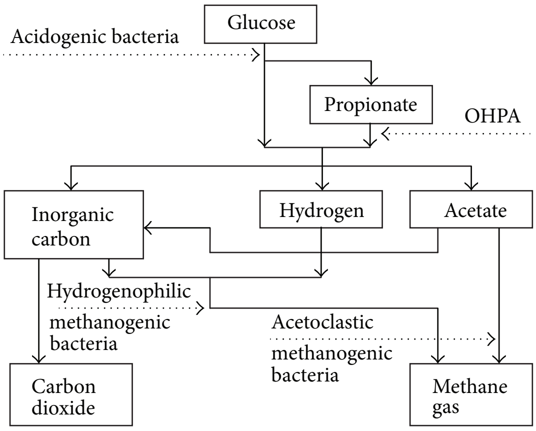


FIGURE 1: A schematic view of a typical anaerobic digestion process.

In the second acidogenic path, Obligate Hydrogen Producing Acetogens (OHPA) decompose propionate into acetate, H_2, and inorganic carbon. In the first methanation path, acetate is transformed into CH_4 and CO_2 by acetoclastic methanogenic bacteria, while, in the second methanation path, H_2 combines inorganic carbon to produce CH_4 under the action of hydrogenophilic methanogenic bacteria. The reaction scheme of this complex bioprocess involves 4 reactions and 10 components.

Since the anaerobic digestion is a complex bioprocess, this dynamical model being described by ten differential equations, for control purpose an appropriately reduced order model can be used. Using the singular perturbation method, the following reduced order dynamical model can be obtained:

$$\dot{X}_1 = \varphi_1 - DX_1, \tag{2}$$

$$\dot{S}_1 = -k_1\varphi_1 - DS_1 + DS_{\text{in}}, \tag{3}$$

$$\dot{X}_2 = \varphi_2 - DX_2, \tag{4}$$

$$\dot{S}_2 = k_3\varphi_1 - k_2\varphi_2 - DS_2, \tag{5}$$

$$\dot{S}_5 = k_4\varphi_1 + k_5\varphi_2 - DS_5 - Q_{CO_2}, \tag{6}$$

$$\dot{P} = k_6\varphi_2 + k_7\varphi_1 - DP - Q_P, \tag{7}$$

where X_1, X_2 are acidogenic and acetoclastic methanogenic bacteria, respectively, and S_1, S_2, S_5 are glucose, acetate and inorganic carbon, respectively, and P is methane; φ_1, φ_2 are the rates of first acidogenic reaction and methanation reaction respectively, $Q_P = c_p P$ with $c_p > 0$ and Q_{CO_2} represent gaseous outflow rates of CH_4 and CO_2, respectively, S_{in} is the influent substrate concentration, D the dilution rate, and k_i $(i = 0, \ldots, 7)$ are yield coefficients. Each reaction rate is a growth rate and may be written as $\varphi_i = \mu_i X_i$, $i = 1, 2$, where μ_i, $i = 1, 2$, is the specific growth rate of reaction i.

Defining the state vector as $\xi = \begin{bmatrix} X_1 & S_1 & X_2 & S_2 & S_5 & P \end{bmatrix}^T$, the model (2)–(7) can be written in matrix form as

$$\begin{aligned}\begin{bmatrix}\dot{X}_1\\ \dot{S}_1\\ \dot{X}_2\\ \dot{S}_2\\ \dot{S}_5\\ \dot{P}\end{bmatrix} &= \begin{bmatrix}1 & 0\\ -k_1 & 0\\ 0 & 1\\ k_2 & -k_3\\ k_4 & k_5\\ k_6 & k_7\end{bmatrix}\begin{bmatrix}\varphi_1\\ \varphi_2\end{bmatrix}\\ &\quad - D\begin{bmatrix}X_1\\ S_1\\ X_2\\ S_2\\ S_5\\ P\end{bmatrix} + \begin{bmatrix}0\\ DS_{\text{in}}\\ 0\\ 0\\ 0\\ 0\end{bmatrix} - \begin{bmatrix}0\\ 0\\ 0\\ 0\\ Q_{\text{CO}_2}\\ Q_P\end{bmatrix},\end{aligned} \tag{8}$$

or in compact form as

$$\dot{\xi} = K\varphi(\xi) - D\xi + F - Q, \tag{9}$$

where $F = \begin{bmatrix}0 & DS_{\text{in}} & 0 & 0 & 0 & 0\end{bmatrix}^T$ is the vector of inflow rates, $Q = \begin{bmatrix}0 & 0 & 0 & 0 & Q_{\text{CO}_2} & Q_P\end{bmatrix}^T$ is the vector of gaseous outflow rates, $\varphi = \begin{bmatrix}\varphi_1 & \varphi_2\end{bmatrix}^T$ is the vector of reaction rates, which can be written as $\varphi(\xi) = G(\xi)\alpha(\xi)$, with $G(\xi)$ being a diagonal matrix whose entries are products of the component concentrations involved in each reaction and $\alpha = \begin{bmatrix}\alpha_1 & \alpha_2\end{bmatrix}^T$ the vector of specific reaction rates, and K is the yield coefficient's matrix. The matrices K and G have the following structure:

$$\begin{gathered}K = \begin{bmatrix}1 & -k_1 & 0 & k_2 & k_4 & k_6\\ 0 & 0 & 1 & -k_3 & k_5 & k_7\end{bmatrix}^T,\\ G = \begin{bmatrix}X_1 S_1 & 0\\ 0 & X_2 S_2\end{bmatrix}.\end{gathered} \tag{10}$$

The most difficult task for the construction of the dynamical model is the modeling of the reaction kinetics [13]. The form of kinetics is complex, nonlinear, and in many cases unknown. In our study one considers that reaction rates are given by the Monod law

$$\varphi_1(\xi) = \mu_1^* \frac{S_1 \cdot X_1}{K_{M_1} + S_1}, \tag{11}$$

and the Haldane kinetic model

$$\varphi_2(\xi) = \mu_2^* \frac{S_2 \cdot X_2}{K_{M_2} + S_2 + S_2^2/K_i}, \tag{12}$$

where K_{M_1}, K_{M_2} are Michaelis-Menten constants, μ_1^*, μ_2^* represent specific growth rate coefficients, and K_i is the inhibition constant.

For simplicity, we will denote the unknown plant parameters by the vector

$$\theta = \begin{bmatrix}\theta_1 \ \theta_2 \cdots \ \theta_{12}\end{bmatrix}^T, \tag{13}$$

where

$$\begin{gathered}\theta_1 = k_1; \qquad \theta_2 = k_2; \qquad \theta_3 = k_3;\\ \theta_4 = k_4; \qquad \theta_5 = k_5 \qquad \theta_6 = k_6 \qquad \theta_7 = k_7\\ \theta_8 = \mu_1^*; \qquad \theta_9 = \mu_2^*; \qquad \theta_{10} = K_{M_1};\\ \theta_{11} = K_{M_2}; \qquad \theta_{12} = K_i.\end{gathered} \tag{14}$$

3. Parameter Estimation Using PSO

At the beginning of parameter estimation, the input and output data are known and the real system parameters are assumed as unknown. The identification problem is formulated in terms of an optimization problem in which the error between an actual physical measured response of the system and the simulated response of a parameterized model is minimized. The estimation of the system parameters is achieved as a result of minimizing the error function by the PSO algorithm.

3.1. Problem Statement. Consider the following n-dimensional nonlinear system:

$$\frac{d\xi(t)}{dt} = f(\xi, t; \theta), \tag{15}$$

where $\xi \in R^n$ is the state vector, $\theta \in R^m$ is the unknown parameters vector, and f is a given nonlinear vector function.

To estimate the unknown parameters in (15), a parameter identification system is defined as follows:

$$\frac{d\hat{\xi}(t)}{dt} = f\left(\hat{\xi}, t; \hat{\theta}\right), \tag{16}$$

where $\hat{\xi} \in R^n$ is the estimated state vector and $\hat{\theta} \in R^m$ is the estimated parameters vector.

The objective function defined as the mean squared errors between real and estimated responses for a number N of given samples is considered as fitness of estimated model parameters:

$$V = \frac{1}{N + D} \sum_{j=1}^{D} \sum_{k=1}^{N} \left(\xi_j^k - \hat{\xi}_j^k\right)^2, \tag{17}$$

where D is the number of measurable states, N is the data length used for parameter identification, whereas ξ_j^k and $\hat{\xi}_j^k$ are the real and estimated values of state j at time k, respectively.

This objective function is a difficult function to minimize because there are many local minima, and the global minimum has a very narrow domain of attraction. Our goal is to determine the system parameters, using particle swarm optimization algorithms in such a way that the value of V is minimized, approaching zero as much as possible.

3.2. Overview of Basic PSO Algorithms. During the last decade, PSO algorithms have gained much attention and wide applications in different fields due to their effectiveness in performing difficult optimization issues, as well as simplicity of implementation and ability to fast converge to a reasonably good solution [14–16]. PSO is a population-based heuristic global optimization technique, first introduced by Kennedy and Eberhart [8] and referred to as a swarm-intelligence technique. It is motivated from the simulation of social behavior of animals such as bird flocking, fish schooling, and swarm. In this algorithm, the population is called a swarm, and the trajectory of each particle in the search space is controlled through the medium of a term called "velocity," according to its own flying experience and swarm experience in the search space. Mathematical description of basic PSO and some important variants is presented in the following.

Candidate solutions of a population called particles coexist and evolve simultaneously based on knowledge sharing with neighboring particles. Each particle represents a potential solution to the optimization problem, and it has a fitness value decided by optimal function. Supposing that search space is D-dimensional, each individual is treated as a particle in the D-dimensional search space. The position and rate of position change for ith particle can be represented by D-dimensional vector, $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ and $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, respectively. The best position previously visited by the ith particle is recorded and represented as $p_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$, called *pbest*. The swarm best position previously visited by all the particles in the population is represented as $p_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$, called *gbest*. Then particles search their best position, which are guided by swarm information p_g and their own information p_i. Each particle modifies its velocity to find a better solution (position) by applying its own flying experience (i.e., memory of the best position found in earlier flights) and the experience of neighboring particles (i.e., the best solution found by the population). Each particle position is evaluated by using fitness function and updates its position and velocity according to the following equations:

$$\begin{gathered} v_i^{k+1} = \omega \cdot v_i^k + c_1 r_1 \left(pbest_i^k - x_i^t \right) + c_2 r_2 \left(gbest_i^k - x_i^k \right), \\ x_i^{k+1} = x_i^k + v_i^{k+1}, \end{gathered} \tag{18}$$

where k is iteration number, ω is inertia weight, c_1 and c_2 are two acceleration coefficients regulating the relative velocity toward local and global best positions, and r_1 and r_2 are two random numbers from interval $[0, 1]$. Many effects have been made over the last decade to determinate the inertia weight. Various studies has shown that under certain conditions convergence is guaranteed to a stable equilibrium point. These conditions include $\omega > (c_1+c_2)/2-1$ and $0 < \omega < 1$. The technique originally proposed was to bound velocities so that each component of v_i is kept within the range $[-V_{\max}, +V_{\max}]$.

Unfortunately, this simple form of PSO suffers from the premature convergence problem, which is particularly true in complex problems since the interacted information among particles in PSO is too simple to encourage a global search. Many efforts have been made to avoid the premature convergence. One solution is the use of a constriction factor to insure convergence of the PSO, introduced in [17]. Thus, the expression for velocity has been modified as

$$\begin{gathered} v_i^{k+1} = h \cdot \left[v_i^k + c_1 r_1 \left(pbest_i^k - x_i^t \right) + c_2 r_2 \left(gbest_i^k - x_i^k \right) \right], \\ x_i^{k+1} = x_i^k + v_i^{k+1}, \end{gathered} \tag{19}$$

where h represents the constriction factor and is defined as

$$h = \frac{2}{\left| 2 - \alpha - \sqrt{\alpha^2 - 4\alpha} \right|}, \tag{20}$$

$$\alpha = c_1 + c_2 > 4. \tag{21}$$

In this variant of the PSO algorithm, h controls the magnitude of the particle velocity and can be seen as a dampening factor. It provides the algorithm with two important features. First, it usually leads to faster convergence than standard PSO. Second, the swarm maintains the ability to perform wide movements in the search space, even if convergence is already advanced, but a new optimum is found. Therefore, the constriction PSO has the potential to avoid being trapped in local optima while possessing a fast convergence. It was shown to have superior performance compared to a standard PSO.

It is shown that a larger inertia weight tends to facilitate the global exploration and a smaller inertia weight achieves the local exploration to fine-tune the current search area [18]. The best performance could be obtained by initially setting ω to some relatively high value (e.g., 0.9), which corresponds to a system where particles perform extensive exploration, and gradually reducing ω to a much lower value (e.g., 0.4), where the system would be more dissipative and exploitative and would be better at homing into local optima. In [19], a linearly decreased inertia weight ω over time is proposed, where ω is given by the following equation:

$$\omega = \left(\omega_i - \omega_f \right) \cdot \frac{k_{\max} - k}{k_{\max}} + \omega_f, \tag{22}$$

where ω_i, ω_f are starting and final values of inertia weight, respectively; $k_{\max}$ is the maximum number of the iteration, and k is the current iteration number. It is generally taken that starting value $\omega_i = 0.9$ and final value $\omega_f = 0.4$ [20].

On the other hand, in [21] was introduced PSO with time-varying acceleration coefficients. The improvement has the same motivation and the similar techniques as the adaptation of inertia weight. In this case, the cognitive coefficient c_1 is decreased linearly and the social coefficient c_2 is increased linearly over time as follows:

$$\begin{gathered} c_1 = \left(c_{1f} - c_{1i} \right) \cdot \frac{k_{\max} - k}{k_{\max}} + c_{1i}, \\ c_2 = \left(c_{2f} - c_{2i} \right) \cdot \frac{k_{\max} - k}{k_{\max}} + c_{2i}, \end{gathered} \tag{23}$$

where c_{1i} and c_{2i} are the initial values of the acceleration coefficients c_1 and c_2; c_{1f} and c_{2f} are the final values of the acceleration coefficients c_1 and c_2, respectively. Usually, $c_{1i} = 2.5$, $c_{2i} = 0.5$, $c_{1f} = 0.5$, and $c_{2f} = 2.5$.

3.3. Identification Algorithm. Considering all the states of the nonlinear system (8) at the sampling moments $k * Ts$ (Ts = sampling period) known, the identification algorithm has the following steps.

Step 1. Initialize a population of particles with random positions and velocities on D dimensions in search space.

Step 2. For each particle, evaluate the desired optimization fitness function (17) in D variables.

Step 3. Compare particle's fitness evaluation with its *pbest*. If current value is better than *pbest*, then set *pbest* equal to the current value x_i in D-dimensional space.

Step 4. Identify the particle in swarm with the best success so far, and assign its index to the variable *gbest*.

Step 5. Change the velocity and position of the particle according to (19).

```
Step 6. if a criterion is met (usually a sufficiently good fitness
or a maximum number of iterations)
    then Stop;
    else
    go to Step 2.
```

3.4. "Classical" Identification Procedure. The "classical" approach for identification of yield coefficients is a two-step procedure under the assumption that full state measurements are available [1]. This method is based on a state transformation that allows reformulating the dynamical model into separate submodels. The first submodel depends only on the reaction structure and is independent of the kinetics. It can be linearly reparametrized and used for the identification of the yield coefficients by means of linear regressions, provided suitable identifiability conditions are satisfied. We present briefly this method for estimating the yield coefficients, and we use it for comparison in the next section.

The general dynamical model given in (2) represents a particular class of nonlinear state-space models. The nonlinearity lies in the reaction rates $\varphi_i(\xi)$ that are (nonlinear) functions of the state variables. These functions enter the model in the form $K\varphi_i(\xi)$ (where K is a constant matrix), which is a set of linear combinations of the same nonlinear functions $\varphi_i(\xi)$. This particular feature can be exploited to separate the nonlinear part from the linear part of the model by an adequate linear state transformation. More precisely, one chooses a nonsingular partition

$$\begin{pmatrix} K_a \\ K_b \end{pmatrix} = TK, \tag{24}$$

with $K_a \in R^{p\times m}$ of full row rank matrix (i.e., $p = \operatorname{rank}(K)$), $K_b \in R^{(n-p)\times m}$, and T a permutation matrix. The induced partitions of the vectors ξ and u are

$$\begin{pmatrix} \xi_a \\ \xi_b \end{pmatrix} = T\xi, \qquad \begin{pmatrix} u_a \\ u_b \end{pmatrix} = Tu. \tag{25}$$

Model (2) is then partitioned into two submodels

$$\begin{aligned} \frac{d\xi_a}{dt} &= K \cdot \varphi(\xi_a, t) - D\xi_a + u_a, \\ \frac{d\xi_b}{dt} &= K \cdot \varphi(\xi_b, t) - D\xi_b + u_b. \end{aligned} \tag{26}$$

Then with the state transformation

$$\begin{aligned} \xi_a &= \xi_a, \\ z &= C\xi_a + \xi_b, \end{aligned} \tag{27}$$

one transforms the initial model into

$$\begin{aligned} \dot{\xi}_a &= K_a\varphi(\xi_a, z - C\xi_a) - D\xi_a + u_a, \\ \dot{z} &= -Dz + Cu_a + u_b, \end{aligned} \tag{28}$$

where the $(n-p)\times p$ matrix C is the unique solution of

$$CK_a + K_b = 0. \tag{29}$$

That is,

$$C = -K_b K_a^*, \tag{30}$$

where K_a^* is a generalized inverse or pseudoinverse of K_a. The subsystem (29) can be augmented with an equation derived from (8) as follows:

$$\begin{aligned} \dot{z} &= -Dz + Cu_a + u_b, \\ \xi_b &= z - C\xi_a. \end{aligned} \tag{31}$$

It can be considered as a linear time-varying (if d varies in the course of time) model with state z, input (ξ_a, u_a, and u_b), and output ξ_b. It is nonlinearly parametrized by the yield coefficients but linearly reparametrized by the nonzero entries of C. When data of the signals ξ_a, u_a, u_b, and ξ_b are available, the auxiliary model (31) can be used to identify the yield coefficients independently of the knowledge of the reaction rates. The model (31) can be used to perform the identification of the nonzero entries of C by a linear regression technique, with the yield coefficients k_i recovered afterwards from (29).

For the estimation of kinetic parameters the main approach is the use of a parameter observers (high gain observers, regressive parameter estimator, sliding mode observers, etc.). All these techniques have numerous tuning parameters and are difficult to implement.

4. Simulation Results and Discussion

The efficacy of our approach is shown by numerical simulations on an interval of 30 hours. The model given by relation (9) was integrated using a fourth-order Runge-Kutta routine with a sampling period of 1 minute and with initial conditions: IC = $\begin{bmatrix} 2.5 & 6 & 0.2 & 10 & 0.5 & 0 \end{bmatrix}$.

The influence of sampling period and type of the optimization algorithm and of noisy measurements are analyzed. To compare statistical performances of the different

Table 1: Influence of the sampling period.

Estimated parameter	"Real" value	Sampling period		
		1 min	10 min	30 min
θ_1	5.5	5.5009	5.4811	5.4798
θ_2	1	0.9982	1.0111	1.0637
θ_3	16	15.9110	15.9328	14.5629
θ_4	10	10.0016	9.9707	9.9571
θ_5	1.5	1.4938	1.4626	1.3078
θ_6	0.2	0.2003	0.2011	0.1873
θ_7	3	2.9823	2.9581	2.6947
θ_8	0.2	0.2002	0.1974	0.1949
θ_9	0.6	0.5994	0.6010	0.6188
θ_{10}	0.75	0.7539	0.6754	0.6483
θ_{11}	4	3.9731	3.9007	3.0265
θ_{12}	20	20.3973	19.4738	20.3947
NMSE	**0**	**3.0717*e* − 05**	**5.6623*e* − 04**	**1.2143*e* − 01**

Table 2: Influence of the algorithm type.

Estimated parameter	"Real" value	Type of optimization algorithm		
		PSO-1	PSO-2	PSO-3
θ_1	5.5	5.5613	5.5849	5.4338
θ_2	1	1.0405	1.0083	0.8406
θ_3	16	15.2050	14.9971	15.6475
θ_4	10	10.0848	10.1252	9.9468
θ_5	1.5	1.5721	1.6422	1.1698
θ_6	0.2	0.2022	0.2040	0.3005
θ_7	3	2.8006	2.7916	2.7569
θ_8	0.2	0.1990	0.1975	0.2286
θ_9	0.6	0.5993	0.6311	0.6594
θ_{10}	0.75	0.9991	1.1153	0.4728
θ_{11}	4	3.5221	3.3470	4.0707
θ_{12}	20	20.9257	20.5048	20.0971
NMSE	**0**	**3.0717*e* − 04**	**0.0283**	**0.3845**

approaches the empirical normalized mean square error (NMSE) was used, that is defined as

$$\mathrm{NMSE} = \frac{1}{N}\sum_{j=1}^{N}\mathrm{NMSE}\left(\widehat{\theta}_j\right), \qquad (32)$$

with $\mathrm{NMSE}(\widehat{\theta}_j) = ((\widehat{\theta}_j - \theta_j^*)/\theta_j^*)^2$, where N is the number of estimated parameters, $\widehat{\theta}_j$ is the jth element of the estimated parameter vector while the "$*$" superscript denotes the true value of the parameter.

In order to study the sensitivity of the estimation method to the sampling period and to the type of PSO algorithm, and to the noise, the following parameters were used.

Sampling period: $T_s \in \{1\ \mathrm{min},\ 10\ \mathrm{min},\ 30\ \mathrm{min}\}$.

Types of the optimization algorithm are as follows:

PSO_1: algorithm based on relation (19) with h defined by relation (20), and $c_1 = c_2 = 2.1$;

Table 3: Influence of the noise level.

Estimated parameter	"Real" value	Noise level		
		50 dB	40 dB	30 dB
θ_1	5.5	5.4001	5.3697	5.3362
θ_2	1	1.0066	0.6628	0.4180
θ_3	16	15.5721	16.6508	16.3867
θ_4	10	9.8502	9.8463	9.7771
θ_5	1.5	1.1817	1.1428	0.9150
θ_6	0.2	0.1867	0.2769	0.2711
θ_7	3	2.9626	2.9178	3.1112
θ_8	0.2	0.2042	0.2350	0.3020
θ_9	0.6	0.6030	0.6254	0.5744
θ_{10}	0.75	0.3306	0.2379	0.0058
θ_{11}	4	3.7860	4.7660	4.5629
θ_{12}	20	20.8802	20.9267	19.7113
NMSE	**0**	**3.6827*e* − 3**	**0.0827**	**0.1625**

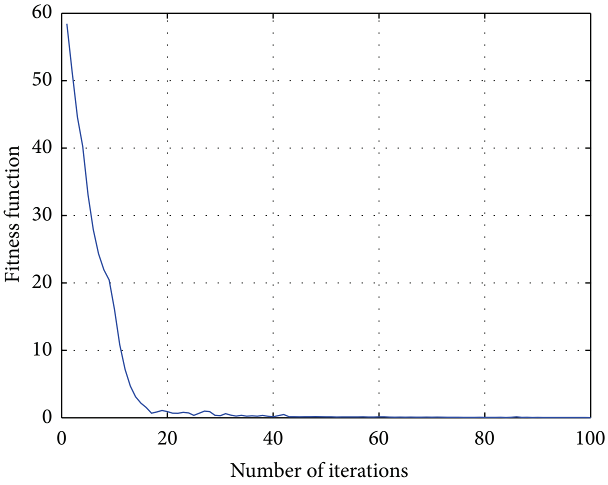


Figure 2: Convergence rate for PSO_1 algorithm.

PSO_2: algorithm based on relation (18) with ω, c_1, and c_2 defined by relations (22) and (23) and $\omega_i = 0.9$, $\omega_f = 0.4$, $c_{1i} = 2.5; c_{2i} = 0.5; c_{1f} = 0.5, c_{2f} = 2.5$;

PSO_3: algorithm based on relation (18) with ω, c_1, andc_2 constants: $\omega = 0.75, c_1 = 1.8, c_2 = 2.2$, and $V_{\max} = 10$.

Noise: {zero-mean white: SNR = 50 dB, 40 dB, 30 dB}.

The results of these simulations are presented in Tables 1, 2, and 3. The simulations were performed with a number of particles between 50 and 120. All the presented results are obtained for a number of 80 particles. For a greater number of particles the accuracy of the estimates was not better. In Figure 2 is presented the convergence rate for PSO_1 algorithm with a sampling period of 1 min.

The results presented in Tables 1–3 suggest that our proposed parameter estimation technique yields consistent results. The method is very simple, easily completed and needs fewer parameters, which made it fully developed. However, the research on the PSO is still at the beginning, and a lot of problems are to be resolved. Research on the topology of the new pattern particle swarm which has a better function can be carried out. The neighbouring topology of the different particle swarms is based on the imitation of the different societies. It is meaningful to the use and spread of the algorithm to select the proper topology to enable PSO have the best property and do the research on the suitable ranges of different topologies. Blending PSO with the other intelligent optimization algorithms means combining the advantages of the PSO with the advantages of the other intelligent optimization algorithms to create the compound algorithm that has practical value. For example, the particle swarm optimization algorithm can be improved by the Simulated Annealing approach. Rapid swarm convergence is one of the main advantages of PSO, but this can also be problematic since, if an early solution is suboptimal, the swarm can easily stagnate around it without any pressure to continue exploration. Overall the results indicate that PSO algorithms can be used in the optimization of parameters during model identification.

5. Conclusions

The paper presents a particle swarm optimization based identification procedure for offline estimation of yield and kinetic coefficients in an anaerobic wastewater treatment bioprocess. The identification scheme is formulated in terms of an optimization problem where the error between an actual physical measured response of the system and the simulated response of a parameterized model is minimized. This function is multimodal, and classical iterative methods fail to find the global optimum. The estimation of the system parameters is achieved as a result of minimizing the error function by the PSO algorithm.

The simulations evaluated the effects of sampling period and some basic variants of PSO algorithm and of the noisy measurements. The proposed strategy can still converge to accurate results even in the presence of measurement noise, as illustrated by the numerical study. The PSO algorithm has a simpler procedure and higher computational efficiency than other optimization techniques.

Nomenclature

c_1, c_2:	Acceleration coefficients
D:	Dilution rate (h^{-1})
F:	Vector of feeding rates
F_{in}:	Input feed rate (Lh^{-1})
K:	Matrix of yield coefficients
$k_i, i = \overline{1,7}$:	Yield coefficients
Q:	Vector of rates of removal of the components in gaseous form
S_1:	Glucose concentration (g/L)
S_2:	Acetate concentration (g/L)
S_5:	Inorganic carbon concentration (g/L)
X_1:	Acidogenic concentration (g/L)
X_2:	Acetoclastic methanogenic concentration (g/L)
P:	Methane concentration (g/L)
S_{in}:	Glucose concentration on the feed (g/L)
Φ:	Vector of reaction rates (reaction kinetics)
μ_1^*, μ_2^*:	Maximal specific growth rates (h^{-1})
ξ:	State vector
$\mathfrak{R}$:	The set of real numbers
Ω:	Inertia weight
θ:	The vector of unknown parameters
$\theta_i, i = \overline{1,9}$:	The unknown parameters.

Acknowledgments

This work was supported by CNCSIS-UEFISCSU, Project no. PN II-RU TE 106, 9/04.08.2010.

References

[1] G. Bastin and D. Dochain, *On-Line Estimation and Adaptive Control of Bioreactors*, Elsevier, New York, NY, USA, 1990.

[2] L. Chen, *Modelling, identifiability and control of complex biotechnological systems [Ph.D. thesis]*, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, 1992.

[3] L. M. Li and S. A. Billings, "Continuous time non-linear system identification in the frequency domain," *International Journal of Control*, vol. 74, no. 11, pp. 1052–1061, 2001.

[4] L. Ljung, *System Identification—Theory for the User*, Prentice-Hall, Upper Saddle River, NJ, USA, 2nd edition, 1999.

[5] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[6] J. M. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.

[7] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Proceedings of the European Conference on Artificial Life (ECAL '91)*, Elsevier, 1991.

[8] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.

[9] J. Kennedy, "Particle swarms: optimization based on sociocognition," in *Recent Developments in Biologically Inspired Computing*, The Idea Group, 2004.

[10] H. Li, H. Zhong, Z. Yan, and X. Zhang, "Particle swarm optimization algorithm coupled with finite element limit equilibrium method for geotechnical practices," *Mathematical Problems in Engineering*, vol. 2012, Article ID 498690, 14 pages, 2012.

[11] O. Bernard and G. Bastin, "Identification of reaction networks for bioprocesses: determination of a partially unknown pseudo-stoichiometric matrix," *Bioprocess and Biosystems Engineering*, vol. 27, no. 5, pp. 293–301, 2004.

[12] D. Selişteanu, E. Petre, and V. B. Răsvan, "Sliding mode and adaptive sliding-mode control of a class of nonlinear bioprocesses," *International Journal of Adaptive Control and Signal Processing*, vol. 21, no. 8-9, pp. 795–822, 2007.

[13] S. F. Azevedo, P. Ascencio, and D. Sbarbaro, "An adaptive fuzzy hybrid state observer for bioprocesses," *IEEE Transactions on Fuzzy Systems*, vol. 12, pp. 641–651, 2004.

[14] I. Hassanzadeh and S. Mobayen, "Controller design for rotary inverted pendulum system using evolutionary algorithms," *Mathematical Problems in Engineering*, vol. 2011, Article ID 572424, 17 pages, 2011.

[15] T.-S. Zhan and C.-C. Kao, "Modified PSO method for robust control of 3RPS parallel manipulators," *Mathematical Problems in Engineering*, vol. 2010, Article ID 302430, 25 pages, 2010.

[16] P. Umapathy, C. Venkataseshaiah, and M. S. Arumugam, "Particle swarm optimization with various inertia weight variants for optimal power flow solution," *Discrete Dynamics in Nature and Society*, vol. 2010, Article ID 462145, 15 pages, 2010.

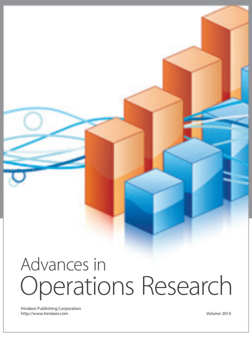
[17] M. Clerc and J. Kennedy, "The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

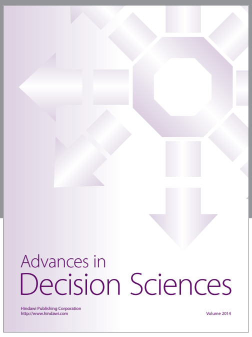
[18] X. Qian, M. Cao, Z. Su, and J. Chen, "A hybrid particle swarm optimization (PSO)-simplex algorithm for damage identification of delaminated beams," *Mathematical Problems in Engineering*, vol. 2012, Article ID 607418, 11 pages, 2012.

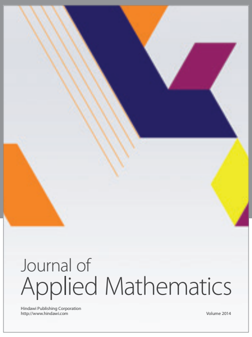
[19] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in Particle Swarm Optimization," in *Proceedings of the Congress on Evolutionary Computating*, pp. 84–88, July 2000.

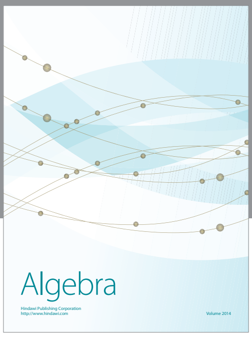
[20] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1945–1950, IEEE Press, 1999.

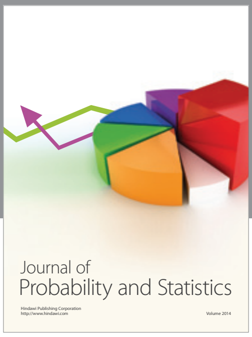
[21] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.


Advances in
Operations Research

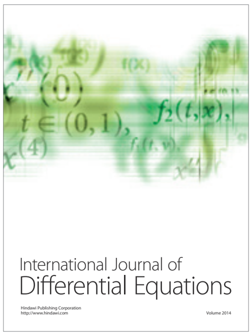

Advances in
Decision Sciences


Journal of
Applied Mathematics

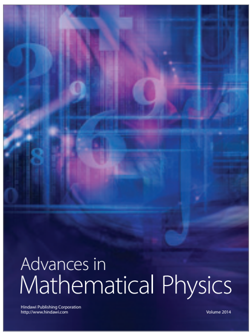

Algebra

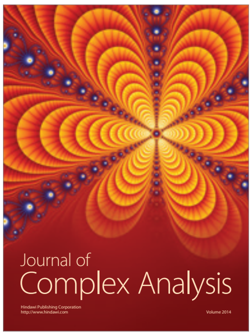

Journal of
Probability and Statistics

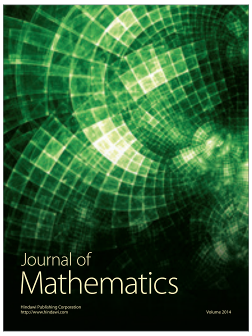
The Scientific
World Journal

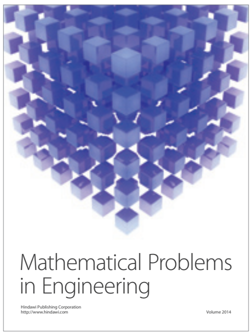

International Journal of
Differential Equations

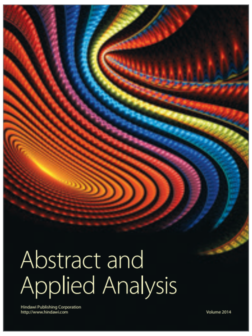
International Journal of
Combinatorics

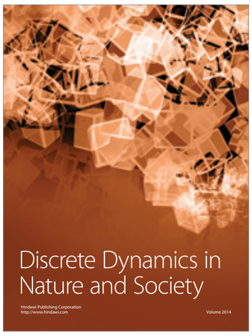

Advances in
Mathematical Physics

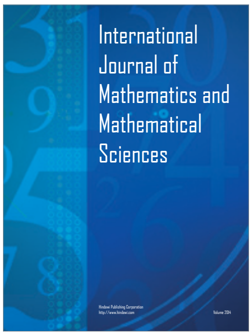

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

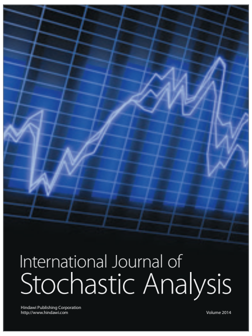

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization