Freeze-drying modeling and monitoring using a new neuro-evolutive technique

This is an electronic version (author's version) of the paper:

Drăgoi E. N., Curteanu S., Fissore D. (2012). Freeze-drying modeling and monitoring using a new neuro-evolutive technique. Chemical Engineering Science (Elsevier), 72(16), 195-204. DOI: 10.1016/j.ces.2012.01.021

Freeze-drying modeling and monitoring using a new neuro-evolutive technique

Elena-Niculina Drăgoi[1], Silvia Curteanu[1], Davide Fissore[2,*]

[1]*Departiment of Chemical Engineering,*

"Gheorghe Asachi" Technical University of Iasi, Bd. Mageron 71A, 700050 Iasi, Romania

[2]*Dipartimento di Scienza Applicata e Tecnologia,*

Politecnico di Torino, corso Duca degli Abruzzi 24, 10129 Torino (Italy)

[*] Corresponding author
E-mail: davide.fissore@polito.it
Tel: +39-(0)11-0904693
Fax: +39-(0)11-0904699

Abstract

This paper is focused on the design of a black-box model for the process of freeze-drying of pharmaceuticals. A new methodology based on a self-adaptive differential evolution scheme is combined with a back-propagation algorithm, as local search method, for the simultaneous structural and parametric optimization of the model represented by a neural network.

Using the model of the freeze-drying process, both the temperature and the residual ice content in the product vs. time can be determine off-line, given the values of the operating conditions (the temperature of the heating shelf and the pressure in the drying chamber). This makes possible to understand if the maximum temperature allowed by the product is trespassed and when the sublimation drying is complete, thus providing a valuable tool for recipe design and optimization.

Besides, the black box model can be applied to monitor the freeze-drying process: in this case, the measurement of product temperature is used as input variable of the neural network in order to provide in-line estimation of the state of the product (temperature and residual amount of ice).

Various examples are presented and discussed, thus pointing out the strength of the tool.

Key words

Drying; Product processing; Simulation; Bioprocessing; Freeze-drying; Artificial Neural Network.

1. Introduction

Freeze-drying is widely used in pharmaceuticals manufacturing to recover a drug from an aqueous solution as the process is carried out at low temperature, thus preserving heat-sensitive molecules. Firstly, the product is put in vials, placed onto the shelves of the drying chamber. Then, product temperature is decreased, thus freezing most of the water (the "free water"). Afterwards, the pressure is lowered, thus causing ice sublimation (primary drying). In this step heat has to be supplied to the product, e.g. through the shelf. Finally, the amount of water in the product is reduced to the target level by desorbing the "bound water" (secondary drying): this result is achieved by increasing product temperature, and further decreasing chamber pressure.

The operating conditions of a freeze-drying process, i.e. the temperature of the heating shelf, the pressure in the drying chamber, and the duration of the drying steps, have to be carefully selected in order to maintain product temperature below a limit value, that is a characteristic of the product being processed, and to minimize the duration of the drying steps. With this respect, the most critical phase of the whole process is the primary drying, as the limit temperature is lower, and the duration is higher.

Recipe design is generally accomplished by means of a trial-and-error approach: various tests are carried out at lab-scale, and product quality is tested at the end. This method is expensive, time consuming, and it does not guarantee that the optimal recipe is obtained. In this framework, the Guidance for Industry PAT (Process Analytical Technology), issued by US-FDA in 2004, encourages the development of new systems for designing, analyzing and controlling manufacturing, with the goal of ensuring final product quality: quality should be built-in or it should be by design, and not just tested in the products at the end of the process.

Mathematical modeling is at the basis of various tools that have been proposed for off-line and in-line recipe design. By using a mathematical model of the process it is possible to build the Design Space of the formulation, i.e. to determine off-line the range of the operating variables that guarantee to obtain a product with acceptable quality (Sundaram et al., 2010; Giordano et al., 2011; Koganti et al., 2011; Fissore et al., 2011a). Besides, mathematical modeling is at the basis of various control algorithms that can be used for in-line optimization, e.g. LyoDriver^{TM} (Fissore et al., 2010; Pisano et al., 2010) or Model Predictive Control systems (Todorov et al., 2007; Todorov and Tsvetkov, 2008; Daraoui et al., 2010; Pisano et

al., 2011).

In addition, mathematical modeling has been used in various tools designed to monitor the process, i.e. to determine in-line both product temperature and the residual amount of ice. Two different approaches were proposed in the past, namely the pressure rise test and the use of soft-sensor. The pressure rise test consists of closing the valve in the duct between the drying chamber and the condenser for a short time period, thus causing pressure increase in the chamber. A mathematical model is used to calculate the curve of pressure rise, and the unknown parameters (e.g. the ice temperature) are retrieved by looking for the best fit between the measured and the calculated values of chamber pressure (Milton et al., 1997; Liapis and Sadikoglu, 1998; Chouvenc et al., 2004; Velardi et al., 2008; Fissore et al., 2011b). The soft-sensor is an observer that couples the measurement of product temperature and a mathematical model of the process to determine the unknown variables. Both the high gain technique and the Kalman filter were proposed to design the observer to monitor the primary drying of a vial freeze-drying process (Velardi et al., 2009 and 2010; Bosca and Fissore, 2011).

It has to be remarked that the methods based on the pressure rise test assume that the batch of vials is homogeneous, i.e. that the product in all the vials has the same temperature and the same ice content. In any case, it is well known to every freeze-drying practitioner that a batch of vials can he highly non-uniform, mainly due to radiation effect from chamber walls, non-uniform shelf temperature, and other reasons (Barresi et al., 2010; Rasetto et al., 2010). Thus, the use of soft-sensors based on the measurement of product temperature appears particularly suitable to get information about the state of the batch as it is possible to place the thermocouples in various positions, thus monitoring more than one vial, and getting information about the distribution of product temperature in the batch.

Generally, first-principle based models were used both to optimize the process, and to monitor it. However, the time required for the calculations could make unfeasible the in-line optimization, and/or increase significantly the time required for the off-line optimization, unless very simplified models are used. Similarly, various simplifications are required to design a soft-sensor according to high gain or the Kalman filter techniques, and this could jeopardize the reliability of the estimations. In this framework significant advantages could be obtained when using black-box models. Todorov et al. (2007) and Todorov and Tsvetkov (2008) proposed a hybrid Wiener-Hammerstein Fuzzy-Neural model to simulate the dynamics of the process, and implemented it in a Model Predictive Control scheme.

In this paper we will focus on the use of a simple feed-forward neural network, and we will show how it is possible to use it for both modeling and monitoring purposes. Also, an efficient method to develop optimal topology of the neural model is applied.

An artificial neural network (ANN) is a mathematical model which simulates the behavior of the brain, and is composed of neurons organized into layers. In particular, in the feed-forward multilayer perceptron neural network (MLP) each neuron from a layer is connected with all the neurons from the next layer by weighted links. Compared with phenomenological models, the neural networks are much simpler, and the required calculations are faster: these advantages are crucial when calculation speed is an important factor, like in the case of process control applications. Besides, as the neural models are not based on the physical and chemical laws that govern the processes, they are recommended when the phenomenology of the process is too complex, or not perfectly known, as in the case of freeze-drying processes.

The neural networks are very easy to set up and use for specific purposes, but this simplicity is deceptive due to the correlation between the performance of the network and its structure. Consequently, the determination of an optimal topology (number of inputs, outputs, hidden layers and neurons in each hidden layer) and of the optimal internal parameters (weights, biases and activation functions) is an important step that can lead to unacceptable results when it is improperly performed. In each case, different methodologies can be applied to solve the specific optimization. For example, for determining the optimal internal parameters, various training algorithms can be used, depending on the topology of the network. These algorithms can be grouped into supervised (with error correction or reinforcement) and unsupervised algorithms (Rojas, 1996). On the other hand, the optimal topology can be found using techniques that can be grouped into trial-and-error, empirical or statistical methods, constructive or destructive methods, and evolutionary techniques (Curteanu and Cartwright, 2011).

The combination of neural networks with evolutionary algorithms forms a special class of networks called evolutionary neural networks (Xin, 1999). The advantage of using evolutionary approaches over traditional algorithms consists in the ability to escape local minima, robustness, and ability to adapt in a changing environment (Abbass, 2001). Moreover, the definition of performance criterion is more flexible, and the evolution can be combined with a classic learning algorithm (Xin, 1999; Floreano et al., 2008).

In this work an algorithm belonging to evolutionary techniques, namely Differential Evolution (DE), is used to determine the optimal topology of the neural network. DE is a population based stochastic metaheuristic inspired from the Darwinian principle in which only the best individuals survive to the next generations (Zaharie, 2009). Metaheuristics are universal methods that can be successfully applied to a large area of problems (Gandibleux and Ehrgott, 2005). Their advantages consist in effectiveness and flexibility because they are able to find optimal solutions even in cases of problems with multiple local optima, without requiring special conditions of the objective functions (Olafsson, 2006; Feoktistov, 2006). Besides, among other algorithms from the class of evolutionary algorithms, DE distinguishes as a fast, robust and with fewer control parameters (Bedri Ozer, 2010). In order to improve its performance and to make it more powerful, a series of strategies were developed, namely: i) replacing the manual tuning of control parameters with adaptive and self-adaptive mechanisms; ii) using more mutation strategies; and iii) combining DE with other optimization techniques (Brest, 2009).

In this paper the simultaneous structural and parametric optimization of neural networks is performed with a self-adaptive differential evolution combined with back-propagation algorithm as a local search method. Although better than other algorithms, DE not always performs as expected when trying to solve the large optimization problem described by the neural network and, in order to get better performance, the back propagation algorithm was used to improve the locally best solution obtained.

The paper is organized as follows. The structure and the principal ideas that lie behind the DE algorithm, and the methodology combining DE with neural network, are described in Section 2. The particularities of the proposed algorithm are listed in Section 3. The freeze drying process and its special characteristics are described in Section 4. The results obtained when using the proposed tool for process modeling and monitoring are presented and discussed in Section 5.

2. Differential Evolution

Initially proposed by Storn and Price (1995), DE is an optimization algorithm whose structure is similar to that of genetic algorithms (GA). The main differences are encountered at two levels: encoding and mutation. Classical GAs use a binary encoding, while DE a floating

point encoding. In the DE case, the mutation is performed by adding a scaled differential term to a base vector, while in the GA, mutation is a result of small perturbations in the genes of individuals (Das and Suganthan, 2011).

Like all evolutionary algorithms, DE starts with a pool of potential solutions (the initial population). Until a stop criterion is reached, the solutions are evolved in three stages: mutation, crossover and selection. At the end of the selection step, a new population (the next generation) is obtained. The current generation is then replaced with the next generation and the evolution is repeated. Each solution in the population is a vector containing an encoded structure of the problem being solved. Each cell of the vector contains a specific information denoted characteristic. In this paper, the words 'individual' and 'vector' are alternating and refer to the same term.

The initial population can be randomly generated or taken from the output of another optimization algorithm. In the mutation step, for each individual in the population, three distinct individuals are randomly chosen from the current population in order to create a new, mutated individual by applying Eq. (1):

$$\omega_i = \alpha + F \cdot \beta \tag{1}$$

where α is the base vector, β is the differential term and F is the scaling factor. The differential term is determined by the difference of the other two individuals randomly chosen from the current population:

$$\beta = x_k - x_p \tag{2}$$

F is one of the control parameters of the DE algorithm, and its role is to control the rate at which the population evolves. Various researchers use values for this parameters from the (0,1] interval because, although no upper limit was established, values greater than 1 are seldom effective (Price et al., 2005).

In the crossover step, the characteristics of the mutated individuals are combined with the ones from individuals in the current population with a probability *Cr*, resulting in a new population called trial population. *Cr* is called crossover probability, and its role is to provide extra diversity, and to exploit decomposability (Price et al., 2005). This is another control parameter of the DE algorithm, and it can take values in the interval (0,1].

Beside *F* and *Cr*, the DE algorithm has a third control parameter: the population dimension (*Np*), which represents the number of individuals that form a population in the

current generation. The higher the value of this parameter, the higher the probability to find a good solution but, at the same time, the higher the computational time. As a result, a compromise between efficiency and computational time must be obtained. One rule of thumb indicates that $Np = 10D$ (where D is the problem dimensionality) (Storn, 2008), while other implies that the best values are in the interval 5 - 10D (Das and Suganthan, 2011).

In the selection step, the individuals from the current and trial population compete to each other. The winning individuals are chosen based on their fitness values, and they survive to the next generation.

A general structure of the algorithm is detailed in Figure 1, where g is the current generation, or the number of repetition of mutation, crossover and selection steps, and i is the current individual in the population.

The fitness is represented by a function which has the role of assigning to each individual a numerical description of its performance. This function allows comparing individuals which cannot be compared using other approaches due to their structural differences. In the current work, the fitness of each vector from the DE algorithm is computed by creating the neural network associated with it, and determining its mean square error (MSE) in the training phase. A smaller MSE indicates a better network, but, because the methodology performs a maximization of the fitness function, the following equation was considered:

$$Fitness_i = \frac{1}{MSE_{training} + err\_correction} \tag{3}$$

where $err\_correction$ is a constant value (equal to 10^{-10}) introduced to eliminate the improbable case when $MSE_{training}$ is equal to 0.

By applying different strategies for the mutation and selection, different variants of the DE algorithm were created. In this work, two variants of the algorithm were used: DE/Best/1/Bin and DE/RandToBest/1/Bin. In the DE/Best/1/Bin version, the base vector in the mutation phase is chosen as being the best individual in the population (Best), a single differential term is used (1) and the crossover type is binomial (Bin). In the binomial crossover, for each parameter of the individual, a random number between 0 and 1 is generated. If the value is smaller than Cr, then the trial vector will contain the parameter from the mutation vector. In the contrary case, it will contain the parameter from the current individual. In the DE/RandToBest/1/Bin variant, the differential term from the mutation phase

is slightly modified by incorporating the difference between the best individual found so far and the base vector. As in the previous version, only a single differential term is used combined with binomial crossover.

2.1 Stopping criteria

Depending on the type of problem and the characteristics of the fitness function, various stop conditions can be applied. In case of constrained problems the algorithm stops when all constraints are satisfied, while for multi-objective optimization it is not always clear when the search must stop, due to the nature of conflicting objectives (Price et al., 2005). The most used stop condition is the number of current generations, which should be lower than a predefined maximum value. The disadvantage of this approach consists in the fact that finding a suitable maximum value is a trial and error procedure due to the randomness factor of the algorithm (Zielinski and Laur, 2008).

In this work a combined stop criteria is used. One of the advantages of neural network optimization is that although the structure is not well known, the objective is to obtain high performance that, in most cases, is quantified using mean squared error. Consequently, the first condition is that the mean squared error for the training data must be smaller than 10^{-8}. This value was chosen based on empirical data and complexity of the studied process. The second condition used is related to the maximum number of generations (G_{max}). The algorithm will stop when the MSE for the training phase is smaller than 10^{-8} or when the current number of generations reached G_{max}.

2.2 Control parameters

The robustness and efficiency of the algorithm are sensitive to the setting of control parameters and, consequently, the determination of their optimal values is an important step when using DE for any kind of optimization (Brest, 2009). This sensitivity is determined by the influence of F and Cr on the offspring creation in the mutation and crossover step, and by the direct correlation between Np and the probability to obtain good results. In the same time, each parameter influences the others by poorly known interactions. Consequently, the determination of good values for the control parameters is a hard problem, although the creators of the algorithm suggest that is a straightforward process (Price et al., 2005).

The optimal values of the control parameters are problem dependent (Lu et al., 2010) and the best data for a specific problem are not necessarily good for another one. Some empirical rules were established by various researchers, but sometimes they are confusing, due to the contradictorily directions proposed.

In order to determine the optimal values of the control parameters for specific problems various mechanisms that eliminated the need for empirical rules were determined. In the majority of studies, the self-adaptive methods outperform the others, and various researchers proposing different variants of this approach. In the self-adaptive variants, the parameters are included into the algorithm itself, each individual containing along with the specific characteristics, a group of control parameters. This mechanism can achieve better performance by gradually adapting to the learning experience (Hu and Yan, 2009). By introducing the parameters into the individuals, the need of user intervention regarding the control parameter settings is eliminated. This makes the algorithm more attractive to specialists from different fields of activity such as chemical engineering.

Among the first studies of self-adaptability for the DE algorithms is the study of Abass (2002). In his work, F and Cr are introduced into the individuals, their values changing using the same mechanisms of mutation and crossover as the other characteristics. Other researchers introduced new mutation schemes, the control parameters evolving by applying different rules to the one used for the other information contained by the individuals (Brest et al., 2007; Zhang et al., 2010).

The main disadvantage of using different formulas for evolving the control parameters is that the complexity and computational time of the algorithm rises. If for problems with low dimensionality this rise is about few seconds or milliseconds, in case of high dimensionality problems the running time is higher. Depending on the type of problem being solved, its application and the computational environment, this increase can be acceptable or not.

In order to keep the complexity and computational time to the minimum, the self-adaptive mechanism used in this paper is as simple as possible, the control parameters evolving in the same manner as the other parameters of the individuals.

2.3 Differential Evolution and artificial neural networks

In order to combine the evolutionary algorithms with neural networks, an encoding procedure must be applied. This procedure has the role of creating a compact representation of the

neural network structure, representation that can be used and understood by the evolutionary algorithm, in general and by DE, in particular. Three types of encoding are encountered in literature: direct, implicit and developmental (Floreano et al., 2008). In the direct encoding a direct mapping between the genotype (evolutionary algorithms work with population of genotype which is usually a string or a vector) and phenotype (representation of neural network) is created. The implicit encoding has a compact representation based on the principle of gene reusability in the biological development (Mouret and Doncieux, 2008). The developmental encoding uses the genome to direct the process towards the construction of the network (Durr et al., 2006).

Evolution has been introduced into neural networks at three levels: connection weights (the procedure is also known as training), architecture (*i.e.* the structure of the neural network) and learning rules (Abbass, 2001; Xin, 1999). Depending on the period in which these levels are evolved, three cases are encountered: evolution of weights, evolution of architectures, and evolution of both weights and architecture.

In this work, the DE based methodology is used for simultaneous optimization of weights and architecture. Moreover, a direct encoding is used, the genotype being representing by a vector containing all the information from the network. For more details about the structure and information contained in the genotype see Dragoi et al. (2011).

3. SADE_NN_2 algorithm

The methodology used for determining the best neural network model is based on a simple self adaptive Differential Evolution algorithm combined with back-propagation. As a result, we called the algorithm SADE_NN_2. Two training procedures are applied to the neural network: first by DE which acts as a global search algorithm and, second, by the back-propagation which performs a local search, improving only the best solution obtained at the end of each generation. Back-propagation was chosen as it is a simple algorithm, widely used, especially for the feed forward neural networks. In order to have a low computation time, the training procedure is applied only once per each generation, for the best solution obtained. This approach is greedy because the local search is used only on the best individual. Although the probability of finding a better solution in a small number of individuals is lower compared with the case when local search is applied for all the population, the overall

computational time is smaller. From our experience, when computational resources are not scarce, the variants in which the local search is applied on a limited number of individuals are better than the ones using a local search on high percentage of population.

Another aspect that needs to be taken into consideration is the dimensionality of the problem. In the present work, part of the problem is represented by the simultaneous optimization of both topology and internal parameters of the neural network. Because a direct encoding with a finite number of elements in the genotype is used, a limit on the topology of the network must be imposed. Consequently, the number of hidden layers and neurons in each hidden layer has the upper limits based on different considerations. First, the number of bits of information (dimension of the vector in the population), that are necessary to encode a network, increases exponentially with the network dimensions and, as a result, the search space of the algorithm increases exponentially (Nolfi and Parisi, 1995). Second, in order to approximate with good results the dynamics of any process, the network must respect some empirical laws which depend on the type and number of data used, neural network inputs and outputs. In this paper, the simplest allowed network contains only the input and output layers, and the most complex one has along with input and output layers, two hidden layers with 25 and 15 neurons, respectively. These values were chosen based on the considerations found in Furtuna et al. (2010). The maximum number of hidden layers and neurons in each hidden layer are set on the basis of empirical laws and, during the evolutionary process, their values can be modified to any possible value within the accepted interval. Distinctively, the number of inputs and outputs is fixed and is closely correlated with the modeled chemical process properties and characteristics. The number and type of input and output parameters will be presented in Section 4 as they are strictly related to the considered process.

Concerning the DE algorithm, a normal distribution initialization combined with a modified mutation and a self-adaptive version was used. The normal distribution was chosen because, in a previous work, its use leads to the best results (Dragoi et al., 2011). The modification added to the mutation consists in ordering the individuals based on their fitness. This approach is a greedy one because it is considered that the probability of creating better individuals is higher when better vector is used. As a result, in the two versions used in the current work, only the individuals participating on the differential terms are ordered. For the DE/RandToBest/1/Bin variant, the ordering is not applied to all vectors from the mutation phase because the best individual participates in the differential term. By applying the reordering procedure to all the vectors, the variant will be transform into a different form of

DE/Best/1/Bin, which is not desired.

Although the evolutionary process and the neural networks are general and can be applied to any kind of process, the characteristics of the freeze-drying process imposed the introduction of a specific module for the training phase, depending on whether the neural network is used for modeling or for monitoring purposes. If in case of monitoring the network takes the inputs and computes the outputs, while in the modeling case a set of inputs, describing the dynamic of the system in the first moments, is provided to the neural model. Using only this information, the network predicts the dynamic of the system for the following time interval, reorganizes these predictions and uses them as new inputs, in order to predict the desired values. Because the network does not know at which moment of time the process ends, information about it must be introduced in the methodology, so that the desired performance parameters of the freeze drying process are correctly identified. In order to use the outputs of the neural model determined at a specific moment t as inputs at t+1, a series of recurrent connections are needed to be introduced in the neural network. The introduction of these connections would have an impact not only on the structure of the network but also on its determination, training and performance. These connections were simulated by using data processing functions. All this modifications were performed in the inner workings of the DE algorithm, the general structure of the SADE_NN_2 methodology being presented in Figure 2.

4. Case study

The ANN can be used for both modeling and monitoring purposes (Figure 3). Neural networks with k+n+m inputs and with z outputs are considered. In the first case, the n variables taken at time t, $t-\Delta t$, ..., $t-n\Delta t$ (inputs k to k+n, where k is a value that indicates the number of inputs without time delays) and the m variables taken at time t, $t-\Delta t$, ..., $t-m\Delta t$ (inputs k+n to k+n+m) are provided to the ANN to get the values of the z output variables at time $t+\Delta t$. In this manner, a recurrent neural network with different time delayed inputs is created and used for modeling. In the case when the ANN is applied for monitoring purposes, it is possible to use the measured values of one (or more) input variables; it is possible to use the ANN (feed forward structure without any recurrences and time delayed inputs) as a soft-sensor that estimates one (or more) variables that cannot be

measured using the values of one (or more) variables that are easily measured.

In case of a freeze-drying process the ANN has to be able to provide information about the state of the product as a function of the operating conditions, namely the temperature of the heating fluid (T_{fluid}) and the pressure in the drying chamber (P_c). The state of the product is identified by the temperature at the interface of sublimation (T_i), as this value should remain below the limit value of the product, and by the thickness of the dried layer (L_{dried}), as this variable gives information about the progress of primary drying (the thickness of the frozen layer, or the sublimation flux, could be used as well to this purpose). The structure of the ANN obtained using the SADE_NN_2 algorithm is that shown in Figure 4. It is characterized by one input layer, with 9 nodes, one for each input variables. It has to be remarked that beside time, T_{fluid} and P_c at current time t, the input variables comprise the values of T_i and L_{dried} in the previous three steps of time. In fact, the effect of the operating conditions T_{fluid} and P_c on the state of the product (T_{fluid} and P_c) depends on the state itself, *i.e.* on what happened with the product in the past. The use of three steps of time allowed to get accurate results, as it will be shown in the following. The ANN comprises 1 hidden layer with 6 nodes and 1 output layer with 2 nodes, one for each output variable.

The time interval Δt has to be carefully selected in the identification procedure as it can significantly affect the accuracy of the results obtained with the neural model. The guideline that can be found in literature is to use a fraction of the time constant of the process: in this case a value of 600 s has been used.

When the proposed network is used for process modeling, given the operating conditions, it is possible to calculate the evolution of the state of the product and, in particular, the maximum temperature of the product ($T_{i,max}$) and the drying time. By this way it is possible to determine if the operating conditions belong (or not) to the design space of the formulation and to optimize the process, *i.e.* to determine those operating conditions that minimize the drying time.

In the case when ANN is used for process monitoring, it is possible to exploit the measurement of product temperature at the bottom of the vial (T_B) and use it as input variable of the network. With this respect, it is possible to assume that $T_i \approx T_B$ (this hypothesis is justified by the fact that the thickness of the frozen layer is of few millimeters and, thus, the temperature gradient is very small), or it is possible to determine T_i as a function of T_B and L_{dried} using a mathematical model of the process (Velardi and Barresi, 2008). We did not use

the measurement of L_{dried} that could be available in case the pressure rise test is used to monitor the process. In fact, our goal is to design a sensor that could be used to monitor the real state of the product in all the vials of the batch and, thus, only the temperature measurement could be effective.

In the first part of the work, the ANN designed using the SADE_NN_2 algorithm was validated using mathematical simulation. In this way it was possible to compare the variables estimated by the ANN with the correct values that could be hardly accessed by means of experiments. The freeze-drying of a 10% w/w sucrose aqueous solution was selected as case study. In order to train and test the optimal neural network, a series of data sets were generated using the detailed model of Velardi and Barresi (2008), with the heat and mass transfer coefficients provided by Giordano et al. (2011). The learning pattern was composed of data describing the overall process in four distinct conditions which, in two dimensional space, delimit the search space in a rectangle shape, containing all the acceptable values for the two most important parameters of the process: T_{fluid} and P_c. For each test, the pattern is represented by data describing the evolution of the process when the parameters have values chosen from the rectangle shape.

Finally, the ANN was used to monitor a real freeze-drying cycle. The temperature measurement is provided by a T-type miniature thermocouple (Tersid S.p.A., Italy) placed at the bottom of the vial. The lyophilization cycle is carried out in a LyoBeta 25™ freeze-dryer (Telstar, Spain). It consists of a vacuum-tight chamber having a volume of 0.2 m^3 and equipped with four shelves with a total area of 0.5 m^2. The product is processed using 3 ml tubing vials (internal diameter of 12 mm), loaded on trays, and filled with 0.3 ml of a 10% w/w aqueous solution of an active pharmaceutical ingredient, corresponding to 2.8 mm of product thickness.

5. Results and Discussion

Due to the stochastic nature of the DE algorithm, at each run of the methodology different results, which are in the vicinity of the global optimum, are obtained. Depending on the different settings of the methodology, the distance to the global optimum can be smaller or bigger. Consequently, a series of five runs for each group of settings was performed. In this way, the best network is chosen from the group with the best results and is not based on

random good outputs.

Two versions of the DE algorithm were used to determine the best neural model for the primary drying process: DE/Best/1/Bin and DE/RandToBest/1/Bin. In Table 1, a set of five results for each version is listed. Also, an average fitness value was computed for each variant and, after that, it was used to determine which is better suited to the problem at hand. An encoding "Inputs: Number_of_hidden_neurons_in_the first_hidden_layer: Number_of_hidden_neurons_in_the second_hidden_layer: Outputs" is used in order to indicate each network topology. For example 3:6:3:1 indicates a network with 3 inputs, 6 neurons in the first hidden layer, 3 neurons in the second hidden layer and 1 output.

The average computed fitness function is higher for the networks determined with RandToBest/1/Bin version than the ones determines with Best/1/Bin. This indicates that the former version is better suited to the current process.

In Table 1, the best network is shown in bold, its topology being 9:6:2. It was determined with the RandToBest/1/Bin version, and was chosen based on the consideration that the highest fitness function represents the model with the lowest error index and, consequently, can be considered the optimal or near optimal model for the primary drying.

Using this network, a set of tests was performed to determine whether the model is suitable for its integration in a control system. Thus, this network is tested in various conditions (not included in the training data set) to determine its flexibility to different changes of the process characteristics.

Figure 5 shows an example of the results obtained when the ANN is used to monitor a process. In this case, mathematical modelling has been used to calculate the evolution of the system and, thus, it is possible to calculate exactly the estimation error. Figure 5 shows the evolution in time of the errors affecting the process temperature (e_T), and the thickness of the dried layer (e_{Ldried}) the two outputs of the neural model.

With respect to product temperature, the accuracy of the ANN is very high, as the error is lower than 0.02 K. With respect to the thickness of the dried layer, the maximum estimation error is about 4.5 %, and it decreases to values lower than 0.5 % in the second half of the primary drying.

Various tests were carried out to confirm the adequacy of the proposed ANN to monitor the freeze-drying process. While in Figure 5 the results obtained for constant operating conditions were shown, Figure 6 shows the results obtained in case the operating conditions

are modified during primary drying, evidencing the good agreement between the values estimated by the ANN and the desired ones.

Two different modeling methodologies were used, applying the neural network to model and monitor the process. In the first situation, the ANN results are represented by long term predictions with a recurrent component, while, for the process monitoring, predictions on short time interval are made only in a feed-forward way. Accurate results are obtained in both cases, with errors within accepted limits.

Figure 7 compares the maximum temperature of the product estimated by the ANN with the desired values. At a specific moment of time, the network is feed with the current state of the system and, based on this information, predicts the maximum value of T_i. In Figure 7, the maximum values of the measured T_i are changing only when the operating conditions are modified. In the case of the predictions performed by the neural network, accuracy of the maximum T_i values are dependent of the specific moment of time in which the information from the system is gathered and provided to the network. Consequently, if the chosen moment is closer to the drying time, the accuracy of the prediction is better, because the network estimates the internal parameters of the system for a smaller period.

Finally, Figure 8 shows an example of validation of the soft-sensor based on the ANN for a real freeze-drying cycle. The accuracy of the product temperature estimation is very high, and also the drying time is correctly estimated. In fact, according to the dynamics of the thickness of the dried layer, we can argue that primary drying is completed in four hours, and this is confirmed by the temperature measurement. In fact, when primary drying in the monitored vial is completed, the temperature measured by the thermocouple sharply increases, as the heat arriving from the shelf is no longer used for ice sublimation.

6. Conclusions

Artificial neural networks are a very powerful and flexible tool. Nevertheless, the accuracy of the calculations can be affected by the topology of the network. In this paper we have proposed a new methodology based on a self-adaptive differential evolution scheme, combined with a back-propagation algorithm as local search method. This algorithm allowed for the simultaneous structural and parametric optimization of the network and was used to design a neural network model for a freeze-drying process. The tool was used both for process

modeling and monitoring. In the first case, it is able to accurately predict the evolution of the product, given the operating conditions. Thus, it can be used to determine off-line the "best" recipe (to minimize the drying time), as well as to optimize in-line the process, *e.g.* in a Model Predictive Control algorithm. In the second case, the neural network was utilized as a soft-sensor to estimate the variables that cannot be measured in-line (thickness of the dried layer), based on the available measurement (product temperature). In both cases, the neural network can be an effective tool to get product quality in a freeze-drying process.

Acknowledgements

This work was partially done by financial support provided by EURODOC "Doctoral Scholarships for research performance at European level" project, financed by the European Social Found and Romanian Government. Valuable suggestions of prof. A. Barresi (Politecnico di Torino) are gratefully acknowledged.

List of Symbols

Cr	crossover probability
D	dimensionality of the search space
F	scale factor
G_{max}	number of maximum allowed generations for the DE algorithm
g	current generation in the DE algorithm
L_{dried}	dried layer thickness, m
L_0	product thickness after freezing, m
N	Number of training data
Np	*n*umber of individuals in the DE population
P_c	drying chamber pressure, Pa
T_{fluid}	temperature of the heating fluid, K
T_i	product temperature at the interface of sublimation, K
$T_{i,max}$	maximum value of the product temperature at the interface of sublimation, K
T_B	product temperature at the bottom of the vial, K
u_i	i^{th} individual from the trial population obtained after applying the crossover step
X	list of vectors representing the current population of potential solutions
x_i	i^{th} potential solution from the current populations of vectors

Greeks

α	base term used in the mutation step
β	differential term used in the mutation step
ω_i	mutation solution obtained in the DE algorithm by applying the mutation step to the i vector

References

Abbass, H A., 2001. A memetic Pareto evolutionary approach to artificial neural networks. Proceedings of the 14th Australian Joint Conference on Artificial Intelligence, 2256, 1-12.

Barresi, A.A., Pisano, R., Rasetto, V., Fissore, D, Marchisio, D.L., 2010. Model-based monitoring and control of industrial freeze-drying processes: effect of batch non-uniformity. Drying Technology 28, 577-590.

Bedri Ozer, A., 2010. CIDE: Chaotically Initialized Differential Evolution. Expert System with Applications 37, 4632-4641.

Bosca S., Fissore D., 2011. Design and validation of an innovative soft-sensor for pharmaceuticals freeze-drying monitoring. Chemical Engineering Science 66, 5127-5136.

Brest, J., 2009. Constrained real-parameter optimization with e-self-adaptive differential evolution. In: Mezura-Montes, E. (Ed.). Constraint-handling in evolutionary optimization, Springer, Berlin, pp. 73-93.

Chouvenc, P., Vessot, S., Andrieu, J., Vacus, P., 2004. Optimization of the freeze-drying cycle: a new model for pressure rise analysis. Drying Technology 22, 1577-1601.

Curteanu, S., Cartwright, H., 2011. Neural networks applied in chemistry. I. Determination of the optimal topology of multilayer perceptron neural networks. Journal of Chemometrics 25, 527-574.

Daraoui, N., Dufour, P., Hammouri, H., Hottot A., 2010. Model predictive control during the primary drying stage of lyophilisation. Control Engineering Practice 18, 483-494.

Das, S., Suganthan, P.N.,2011. Differential Evolution: A Survey of the State-of-the-Art. IEEE Transactions on Evolutionary Computation 15, 4-31.

Dragoi, E.N., Curteanu, S., Leon, F., Galaction, A.I., Cascaval, D., 2011. Modeling of oxygen mass transfer in the presence of oxygen-vectors using neural networks developed by differential evolution algorithm. Engineering Applications of Artificial Intelligence 24, 1214-1226.

Durr, P., Mattiussi, C., Floreano, D., 2006. Neuroevolution with Analog Genetic Encoding.

Proceedings of the 9th International Conference on Parallel Problem Solving from Nature. Berlin: Springer-Verlag, Berlin, 671-680.

Feoktistov, V., 2006. Differential Evolution: in search of solutions. Springer-Verlag, Berlin.

Fissore, D., Pisano, R., Velardi, S.A., Barresi, A.A., Galan M., 2010. PAT tools for the optimization of the freeze-drying process. Pharmaceutical Engineering 29, 58-70.

Fissore, D., Pisano, R., Barresi, A.A., 2011a. Advanced approach to build the design space for the primary drying of a pharmaceutical freeze-drying process. Journal of Pharmaceutical Sciences 100, 4922-4933.

Fissore, D., Pisano, R., Barresi, A.A., 2011b. On the methods based on the Pressure Rise Test for monitoring a freeze-drying process. Drying Technology 29, 73-90.

Floreano, D., Durr, P., Mattiussi, C., 2008. Neuroevolution: from architecture to learning. Evolutionary Intelligence 1, 47-62.

Furtuna, R., Curteanu, S., Cazacu, M., 2010. Optimization methodology applied to feed-forward artificial neural network parameter. International Journal of Quatum Chemistry 111, 539–553.

Gandibleux, X., Ehrgott, M., 2005. 1984-2004 - 20 years of multiobjective metaheuristics. But what about the solution of combinatorial problems with multiple objectives? in: Coello Coello, C., Aguirre, A. H., Zitzler, E. (Eds.), Evolutionary multi-criterion optimization, Springer, Berlin, pp. 33-46.

Giordano, A., Barresi, A.A., Fissore, D., 2011. On the use of mathematical models to build the design space for the primary drying phase of a pharmaceutical lyophilization process. Journal of Pharmaceutical Sciences 100, 311-324.

Koganti, V.R., Shalaev, E.Y., Berry, M.R., Osterberg, T., Youssef, M, Hiebert, D.N., Kanka, F.A., Nolan, M., Barrett, R., Scalzo, G., Fitzpatrick, G., Fitzgibbon, N., Luthra, S., Zhang, L., 2011. Investigation of design space for freeze-drying: use of modeling for primary drying segment of a freeze-drying cycle. AAPS PharmSciTech 12, 854-861.

Liapis, A. I., Sadikoglu, H., 1998. Dynamic pressure rise in the drying chamber as a remote sensing method for monitoring the temperature of the product during the primary drying stage of freeze-drying. Drying Technology 16, 1153-1171.

Lu, Y., Zhou, J., Qin, H., Li, Y., Zhang, Y., 2010. An adaptive hybrid differential evolution

algorithm for dynamic economic dispatch with valve-point effects. Expert Systems with Applications 37, 4842-4849.

Milton, N.; Pikal, M.J.; Roy, M.L.; Nail, S.L., 1997. Evaluation of manometric temperature measurement as a method of monitoring product temperature during lyophilisation. PDA Journal of Pharmaceutical Science and Technology 5, 7-16.

Mouret, J.B., Doncieux, S., 2008. MENNAG: a modular, regular and hierarchical encoding for neural-networks based on attribute grammars. Evolutionary Intelligence 1, 187-207.

Nolfi, S., Parisi, D., 1995. Genotypes for neural networks. in. Arbib, M. (Ed.), The handbook of brain theory and neural networks, MIT Press, Cambridge, pp. 431–434.

Olafsson, S., 2006. Metaheuristics. in: Henderson, S.G., Nelson, B.L. (Eds.). Simulation. Handbooks in operations research and management science. Elsevier, Amsterdam, pp. 633-654.

Pisano, R., Fissore, D., Velardi, S.A., Barresi A.A., 2010. In-line optimization and control of an industrial freeze-drying process for pharmaceuticals. Journal of Pharmaceutical Sciences 99, 4691-4709.

Pisano, R., Fissore, D., Barresi, A.A., 2011. Freeze-drying cycle optimization using Model Predictive Control techniques. Industrial Engineering Chemistry Research 50, 7363-7379.

Price, K.V., Storn, R.M., Lampien, J.A., 2005. Differential Evolution. A practical approach to global optimization. Springer, Berlin.

Rasetto, V., Marchisio, D.L., Fissore, D., Barresi, A.A. 2010. On the use of a dual-scale model to improve understanding of a pharmaceutical freeze-drying process. Journal of Pharmaceutical Sciences 99, 4337-4350.

Rojas, R., 1996. Neural Networks. A systematic introduction. Springer, Berlin.

Storn, R., Price, K., 1995. Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, Berkley.

Sundaram, J., Hsu, C.C., Shay, Y.M., Sane, S.U., 2010. Design space development for lyophilization using DOE and process modeling. BioPharm International 23, 26-36.

Todorov, Y. V., Terziyska, M., Petrov, M., 2007. Nonlinear model predictive controller using a Fuzzy-Neural Wiener-Hammerstein model, In: Proceedings of 16th International

Conference on Process Control, Stribske Pleso, Slovakia, Article No 216, 1-7.

Todorov, Y. V., Tsvetkov, T. D., 2008. Volterra Model Predictive Control of a lyophilization plant, In: Proceedings of 4th International IEEE Conference – Intelligent Systems, 3, 2013-2018.

Velardi, S.A., Barresi, A.A., 2008. Development of simplified models for the freeze-drying process and investigation of the optimal operating conditions. Chemical Engineering Research and Design 87, 9-22.

Velardi, S.A., Rasetto, V., Barresi, A.A., 2008. Dynamic Parameters Estimation Method: advanced Manometric Temperature Measurement approach for freeze-drying monitoring of pharmaceutical. Industrial Engineering Chemistry Research 47, 8445-8457.

Velardi, S.A., Hammouri, H., Barresi, A.A., 2009. In line monitoring of the primary drying phase of the freeze-drying process in vial by means of a Kalman filter based observer. Chemical Engineering Research and Design 87, 1409-1419.

Velardi, S.A., Hammouri, H., Barresi, A.A., 2010. Development of a high gain observer for in-line monitoring of sublimation in vial freeze-drying. Drying Technology 28, 256-268.

Xin, Y., 1999. Evolving artificial neural networks. Proceedings of the IEEE 87, 1423-1447.

Zaharie, D., 2009. Influence of crossover on the behavior of Differential Evolution Algorithms. Applied Soft Computing 9, 1126-1138.

Zielinski, K., Laur, R., 2008. Stopping criteria for differential evolution in constrained single-objective optimization. in: Chakraborty U. (Ed.). Advances in differential evolution. Springer, Berlin, pp. 111-138.

Zielinski, K., Peters, D., Laur, R., 2005. Stopping criteria for single-objective optimization. Proceedings of the Third International Conference on Computational Intelligence, Robotics and Autonomous Systems, Singapore.

List of Tables

List of Figures

Lower graph: values of dried layer thickness estimated by the ANN.

(L_0 = 2.8 mm, T_{fluid} = -15°C, P_c = 8 Pa).

Table 1

DE version	Simulation	Network Topology	Fitness
Best/1/Bin	1.	9:9:2	878 878.471
	2.	9:7:2	288 318.847
	3.	9:9:2	899 495.271
	4.	9:10:4:2	660 923.537
	5.	9:8:2	985 436.259
	Average	-	742 610.482
RandToBest/1/Bin	1.	9:9:2	736 362.096
	2.	9:5:2:2	294 705.385
	3.	**9:6:2**	**1 070 414.421**
	4.	9:7:2	941 897.479
	5.	9:8:2	793 596.163
	Average	-	761 995.108

Figure 1

1: Set the parameters of the algorithm

2: $g = 0$

3: Generate initial population $X(g) = \{x_1(g), x_2(g), \ldots\ldots, x_n(g)\}$

4: Compute fitness of the initial population

5: While stop criteria is not reached

6: For each individual in the population

7: ω_i = generate mutant (g)

8: u_i = perform crossover (ω_i , $x_i(g)$)

9: If (fitness(u_i) > fitness ($x_i(g)$)) then

10: $x_i(g + 1) = u_i$

11: Else $x_i(g + 1) = x_i(g)$

12: End If

13: End For

14: $g = g + 1$

15: Compute fitness of the new generation

16: End While

Figure 2

1: Set the *Np* parameter of DE

2: Set the maximum allowed topology

3: $g = 0$

4: Generate initial population by using a normal distribution

$$X(g) = \{x_1(g), x_2(g), \ldots, x_n(g)\}$$

5: Temporarily create the networks encoded into the individuals

6: Compute the fitness of the initial population by using the network's MSE

7: While stop criteria is not reached

8: For each individual in the population

9: $F = x_{i,u}(g)$

10: $Cr = x_{i,u+1}(g)$

11: ω_i = generate mutant (g)

12: u_i = perform crossover (ω_i , $x_i(g)$)

13: Temporarily create the network corresponding to the u_i individual and compute its fitness

14: If (fitness(u_i) > fitness ($x_i(g)$)) then

15: $x_i(g + 1) = u_i$

16: Else $x_i(g + 1) = x_i(g)$

17: End If

18: End For

19: $g = g + 1$

20: Compute fitness of the new generation by temporarily transforming each individual in the corresponding neural network

21: Identify the best individual $x_b(g)$

22: Temporarily create the network associated with $x_b(g)$

23: Apply BackPropagation to the network

24: Encode into $x_b(g)$ the new network obtained after BackPropagation

25: End While

Figure 3

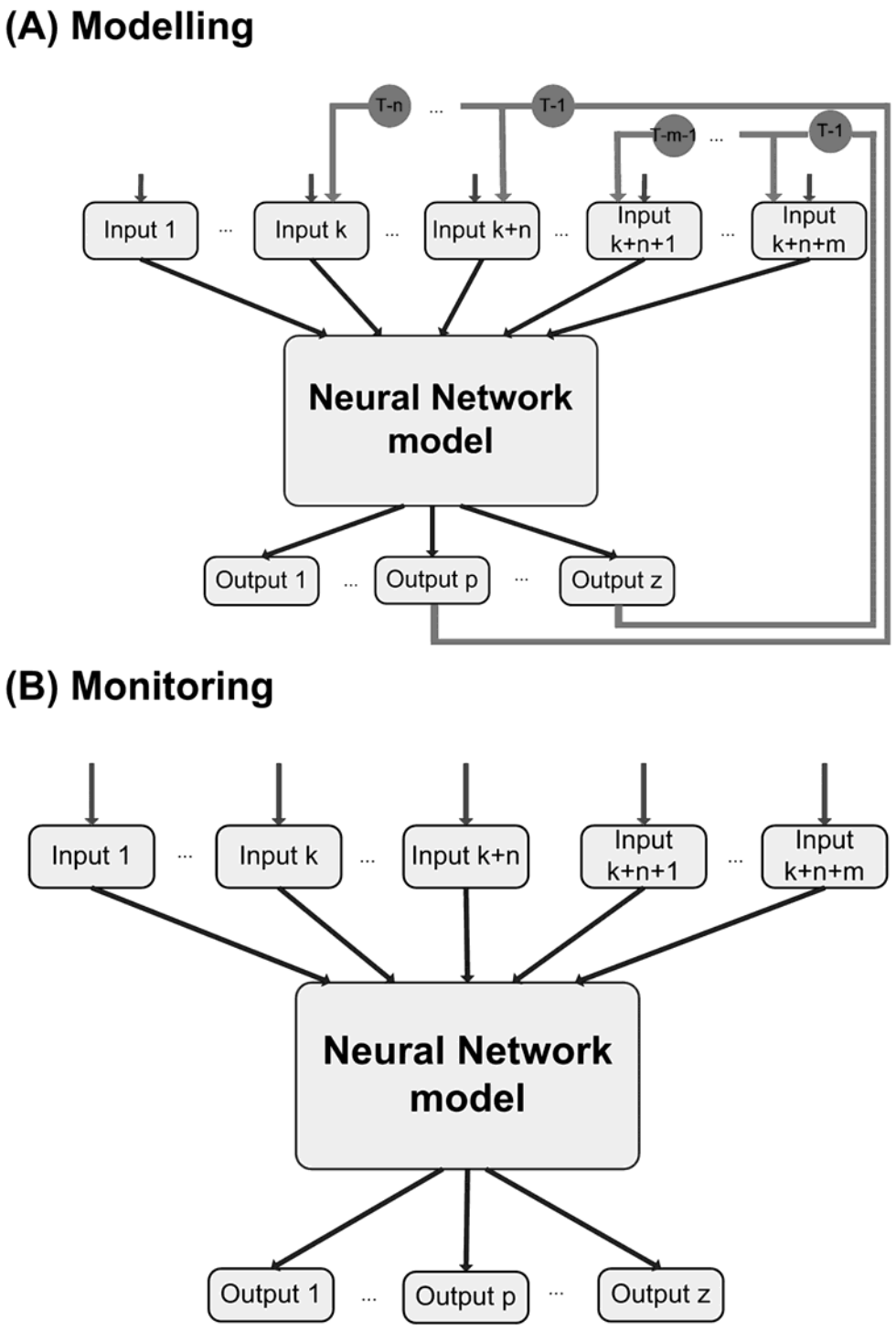

(A) Modelling
T-n
...
T-1
T-m-1
...
T-1
Input 1
...
Input k
...
Input k+n
...
Input k+n+1
...
Input k+n+m
Neural Network model
Output 1
...
Output p
...
Output z
(B) Monitoring
Input 1
...
Input k
...
Input k+n
Input k+n+1
...
Input k+n+m
Neural Network model
Output 1
...
Output p
...
Output z

Figure 4

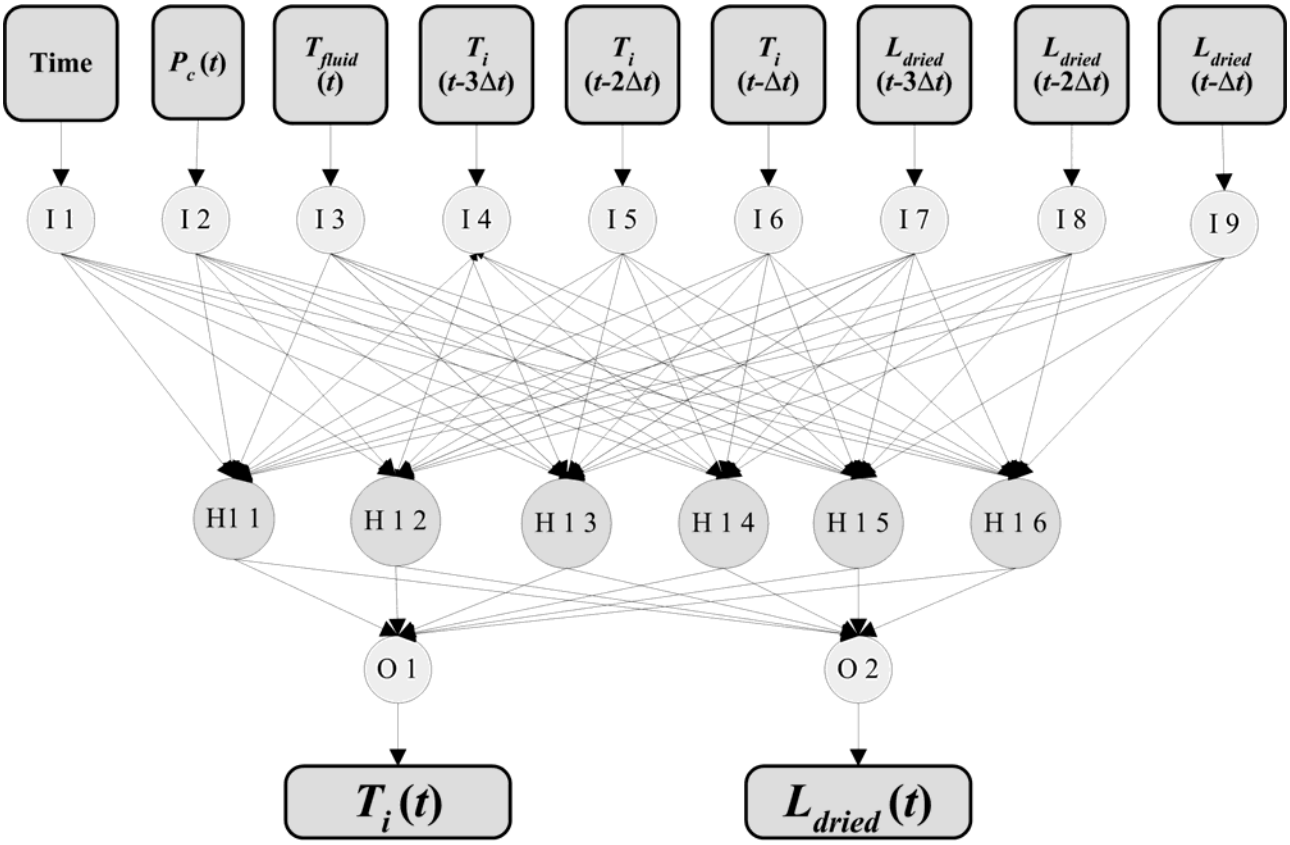

Time
$P_c(t)$
T_{fluid} (t)
T_i $(t\text{-}3\Delta t)$
T_i $(t\text{-}2\Delta t)$
T_i $(t\text{-}\Delta t)$
L_{dried} $(t\text{-}3\Delta t)$
L_{dried} $(t\text{-}2\Delta t)$
L_{dried} $(t\text{-}\Delta t)$
I 1
I 2
I 3
I 4
I 5
I 6
I 7
I 8
I 9
H1 1
H 1 2
H 1 3
H 1 4
H 1 5
H 1 6
O 1
O 2
$T_i(t)$
$L_{dried}(t)$

Figure 5

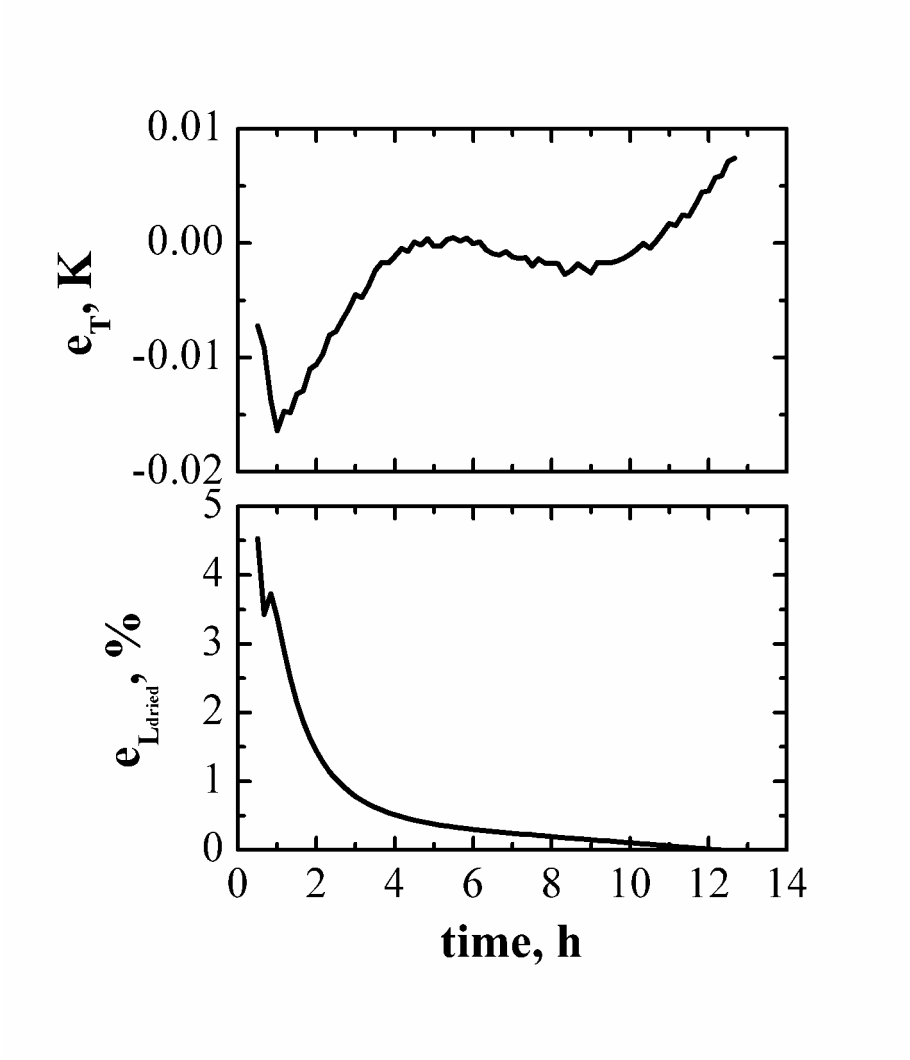


Figure 6

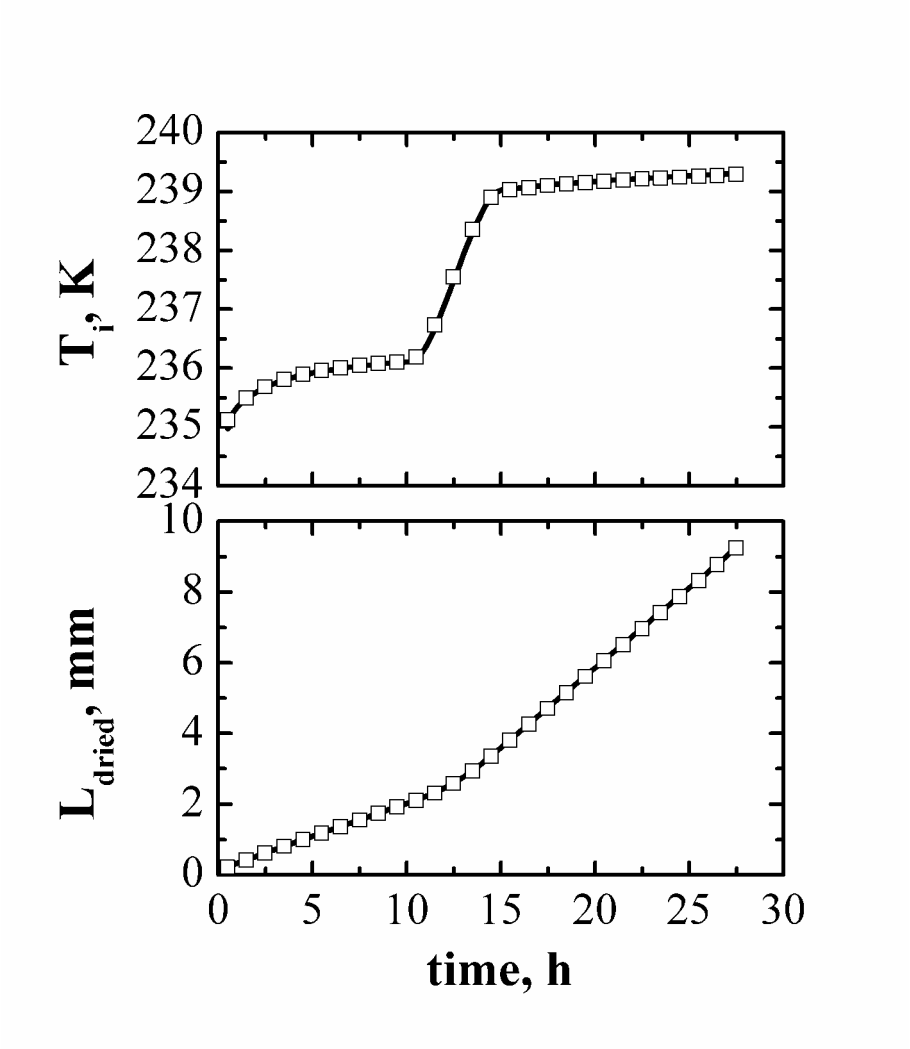

240
239
238
237
236
235
234
T_i, K
10
8
6
4
2
0
L_{dried}, mm
0
5
10
15
20
25
30
time, h

Figure 7

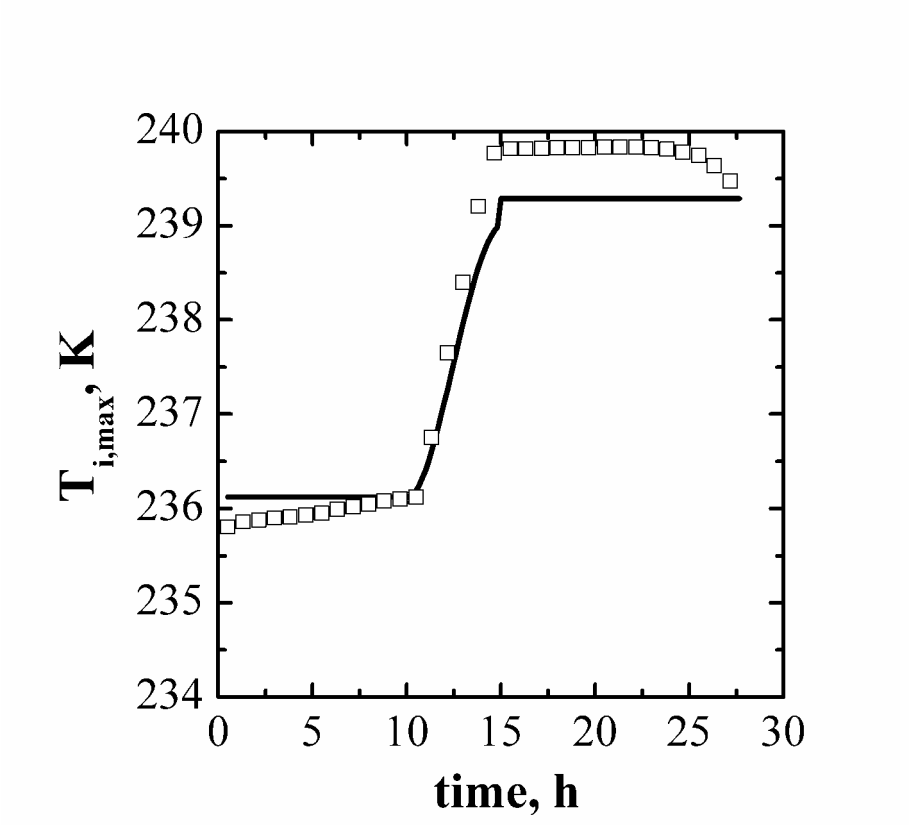
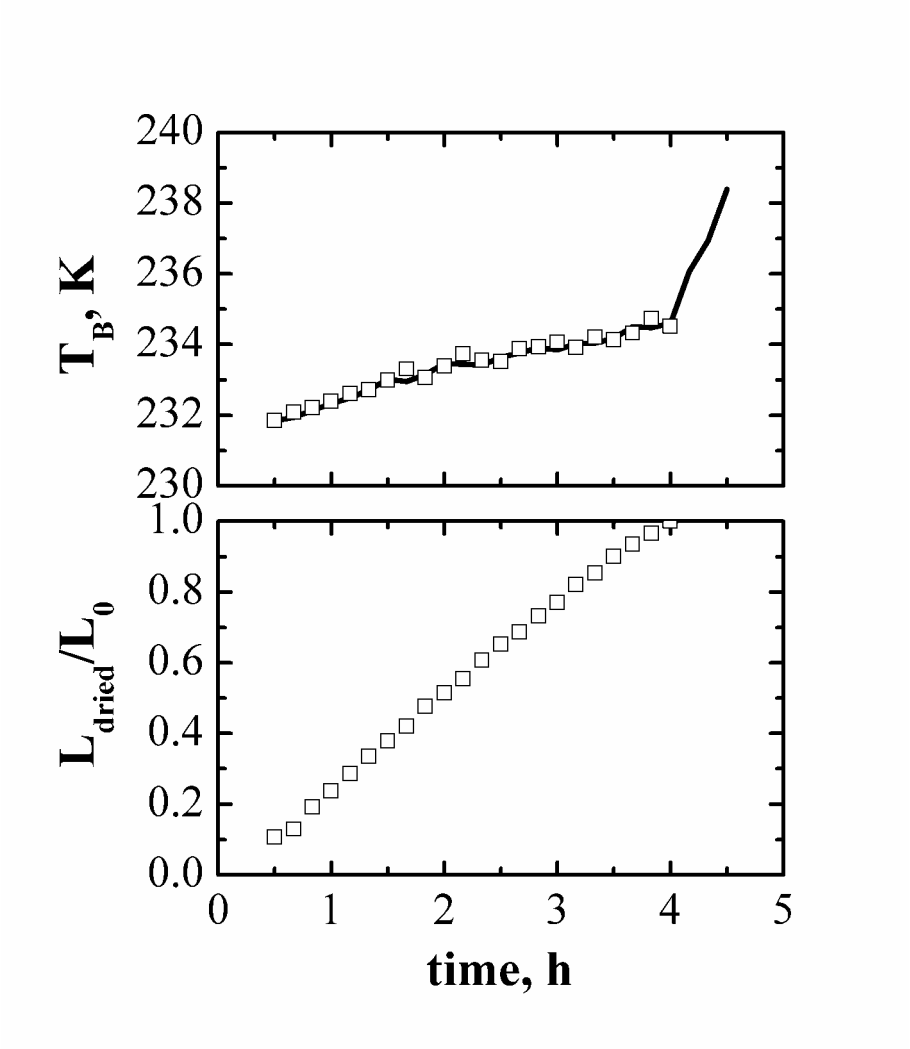


Figure 8


240
238
236
234
232
230
T_B, K
1.0
0.8
0.6
0.4
0.2
0.0
L_{dried}/L_0
0
1
2
3
4
5
time, h