

Modelling of Biotechnological Processes – An Approach Based on Artificial Neural Networks

Eduardo Valente^{1,2}, Miguel Rocha¹, Eugénio C. Ferreira³, and Isabel Rocha³

¹ Department of Informatics / CCTC – University of Minho
Campus de Gualtar, 4710-057 Braga, Portugal
mrocha@di.uminho.pt

² Department of Informatics Engineering, ESTCB
Avenida do Empresário, 6000-767 Castelo Branco, Portugal
eduardo@est.ipcb.pt

³ IBB – Institute for Biotechnology and Bioengineering
Centre of Biological Engineering – University of Minho
Campus de Gualtar, 4710-057 Braga, Portugal
{ecferreira, irocha}@deb.uminho.pt

Abstract. In this chapter we describe a software tool for modelling fermentation processes, the FerMoANN, which allows researchers in biology and biotechnology areas to access the potential of Artificial Neural Networks (ANNs) for this task. The FerMoANN is tested and validated using two fermentation processes, an *Escherichia coli* recombinant protein production and the production of a secreted protein with *Saccharomyces cerevisiae* in fed-batch reactors. The application to these two case studies, tested for different configurations of feedforward ANNs, illustrate the usefulness of these structures, when trained according to a supervised learning paradigm.

1 Introduction

This chapter discusses a software tool, the FerMoANN, that intends to be a useful instrument in research on the modelling, control and optimization of fermentation processes, whose aim is to allow an increase on process productivity.

Most biological processes are difficult to describe mathematically and, as a consequence, it is difficult to predict their behaviour. One of the most difficult aspects of its analysis, especially when working in real time, is the dynamic behaviour of biochemical reactions, since most of the times the kinetic equations associated have parameters that are difficult to estimate. Moreover, problems with sterility and lack of reliable measuring techniques are pointed as factors for not having measurements of key process variables [22].

Therefore, it becomes necessary to use non-deterministic models that could be implemented based on data and that are not tied to real-time measurements.

The Artificial Neural Networks (ANNs) are one of such modelling tools that have been extensively used for representing complex phenomena, since they are universal non-linear function approximators and their application does not require any knowledge about the structure of the model [20]. ANNs are therefore suitable for modelling fermentation processes.

This work integrates the ANNs in FerMoANN, with an intuitive design and interface, aiming at becoming a powerful instrument for researchers in the field of Biology / Biotechnology to study fermentation processes. A similar tool has not been found available, so its specifications were made from scratch, based on suggestions given by researchers from the biological area.

In this work, in order to test the performance of the developed tool, two case studies are considered, with different degrees of complexity. These cases are fermentation processes of the yeast *Saccharomyces cerevisiae* (*S. cerevisiae*) and of the bacterium *Escherichia coli* (*E. coli*).

2 Fermentation Processes

In general, fermentation can be defined as a bioprocess that generates a product through the mass culture of a microorganism [30] in a bioreactor.

There are several factors that can influence a fermentation process and among them there is the quantity and composition of the substrate that are provided to the cell culture. The substrate is what supports the life of a particular microorganism and could include nutrients like glucose, oxygen, and light or nitrogen source. Other important parameters affecting cellular growth are the physical and chemical conditions of the environment (temperature, pH, pressure, etc.) and protein induction [16].

This work focuses on the influence of the substrate administration on microorganism cultures, making possible to test any feed profile and see their influence on the productivity of the culture.

2.1 Cultivation Techniques

There are three main cultivation techniques, which are chosen considering the type of the desired product and the characteristics of the microorganism. These modes of operation are continuous, batch and fed-batch cultures:

Batch. In this cultivation mode, an initial amount of substrate is provided, and the process runs with the generation of biomass and products without the addition of more nutrients, until all the substrate is consumed.

Continuous. In this cultivation mode, the substrate is continuously provided to the bioreactor. Also, a part of the fermentation broth is removed in quantities that allow the culture continuity without reaching the maximum capacity of the bioreactor volume. If the rate of substrate addition is constant, a steady state can be reached inside the bioreactor with constant values for biomass, substrate and product concentrations.

Fed-batch. This technique is based on the continuous or sequenced batch supply of one or more nutrients, without removal of the bioreactor broth. The process continues until the desired volume is reached or until the maximum capacity of the container is achieved. The fed-batch cultures have some advantages, such as prevention of substrate inhibition, high cell density cultivation, etc [34].

It is in this last type of culture where simulation and optimization tools could be of greater value for researchers, since they produce data that could be used in fermentation process control, influencing their behaviour in run-time. A vast set of studies could be found, that use fed-batch bioreactors for process control and optimization, namely those in [2], [3], [16] or [21].

However, some of the main difficulties in the design of on-line optimization and control for biological processes lie in the insufficiency of cheap and reliable hardware and software sensors capable of providing direct and on-line measurements of the biological variables. Many times an off-line optimization is made for the determination of the operating conditions before the cultivation is carried out and the optimized operating conditions are not modified during the cultivation [34]. Even with recent developments in hardware and software sensors, only some of the state variables involved, that are critical for efficient control, are available for on-line measurement. For example, usually the dissolved oxygen concentration and gaseous flow rates are available for on-line measurement while the biomass, products and substrates concentrations are often available via off-line analysis. Given the low frequency in biomass, substrate and product measurements in most fermentation processes, it is very difficult to estimate on-line important variables like specific growth, consumption and production rates, the so-called kinetic variables [14]. The specific cell growth rate is one of the most important process variables characterizing the state of microorganisms during fermentations mainly because the biosynthesis of many products of interest is often related with the values assumed by this variable.

The developed tool was designed to aid in the effort of estimating kinetic variables in fermentation processes from other available measurements. For evaluating the software performance, data were gathered from two separate fermentation processes. These case studies will be briefly described in the next two sections.

2.2 The *Saccharomyces cerevisiae* Fermentation Process

This process is based in the production of a secreted protein with post-translational modifications, using the yeast *Saccharomyces cerevisiae* (SEY2102) as the host organism and SUC2-s2 as secreted protein. This process was studied by Park and Ramirez (PR) in a fed-batch bioreactor [21]. The analogy between the products secreted by this type of cells and mammalian cells, the lack of pathogenicity with humans and other characteristics makes this culture suitable for the large production of useful human proteins.

The equations that describe the PR process are as follows [19] (the detailed explanation of each equation is beyond the objectives of this work):

$$\frac{dx_1}{dt} = \frac{4.75A(x_2 - x_1)}{0.12 + A} - \frac{ux_1}{x_5} \quad (1)$$

$$\frac{dx_2}{dt} = \frac{x_3x_4 \exp(-5x_4)}{0.1 + x_4} - \frac{ux_2}{x_5} \quad (2)$$

$$\frac{dx_3}{dt} = \left(A - \frac{u}{x_5} \right) x_3 \quad (3)$$

$$\frac{dx_4}{dt} = -7.3Ax_3 - \frac{u(x_4 - 20)}{x_5} \quad (4)$$

$$\frac{dx_5}{dt} = u \quad (5)$$

$$A = \frac{21.87x_4}{(x_4 + 0.4)(x_4 + 62.5)} \quad (6)$$

- x_1 : Concentration of secreted protein (units / L)
 x_2 : Concentration of total protein (units / L)
 x_3 : Concentration of cells (g / L)
 x_4 : Concentration of substrate (g / L)
 x_5 : Bioreactor Volume (L)
 u : Feed rate (L / h)
 A : Specific growth rate (h^{-1})

In this experiment, the aim is to find the best operating conditions that maximize the production of the secreted protein, which is intrinsically related with the yeast specific grow rate (A). In equation (6) we can observe that the state variable x_4 (substrate concentration) is the essential element for determining A . So, this equation will be used to produce data for the learning process of the ANN model.

2.3 The *Escherichia coli* Fermentation Process

The bacterium *Escherichia coli* (*Ecoli*) has been one of the most commonly used recombinant protein production systems, because of its ability to grow rapidly and at high densities on inexpensive substrates like glucose. Furthermore, it is one of the best characterized systems regarding molecular genetics, physiology and expression systems. An important problem in this culture lies in the tight control that is necessary during the fermentation process, since at particular conditions (excessive or low glucose concentration) recombinant protein production can be significantly reduced [25], due to the effect of acetic acid or cellular starvation. The main optimization objective in this process is to control the feeding trajectory in a manner that concentration of glucose maintains at optimum levels for cell growth and subsequent high protein production.

The detailed mathematical model describing this process can be found in [26]. The substrate specific consumption rate is given by the following equation:

$$q_s = q_{s,\max} \frac{S}{S + K_S} \frac{K_{i,S}}{K_{i,S} + A} \quad (7)$$

- q_s : Specific substrate uptake rate (h^{-1})
 $q_{s,\max}$: Maximum specific uptake rate (h^{-1})
 K_S : Affinity constant for glucose uptake (g / kg)
 $K_{i,S}$: Inhibition constant of acetate on glucose uptake (g / kg)
 A : Acetate concentration (g/kg)

The specific substrate uptake rate is related with the specific growth rate by:

$$q_s = \frac{\mu}{k_{XS}} \quad (8)$$

μ : Specific growth rate (h^{-1})
 k_{XS} : Global yield coefficient (g/g)

The model differential equations obtained from mass balances to the main state variables are the following, in a matrix format:

$$\frac{d}{dt} \begin{bmatrix} X \\ S \\ A \\ O \\ C_T \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ -k_1 & -k_2 & 0 \\ 0 & k_3 & -k_4 \\ -k_5 & -k_6 & -k_7 \\ k_8 & k_9 & k_{10} \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} X - D \begin{bmatrix} X \\ S \\ A \\ O \\ C_T \end{bmatrix} + \frac{F_{in,S}}{W} \begin{bmatrix} 0 \\ S_{in} \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ OTR \\ -CTR \end{bmatrix} \quad (9)$$

X : Biomass concentration (g/kg)
 S : Substrate concentration (g/kg)
 O : Dissolved oxygen concentration (g/kg)
 C_T : Total dissolved carbon dioxide concentration (g/kg)
 k_x : Yield parameters
 D : Dilution rate (h^{-1})
 $F_{in,S}$: Liquid mass flow of the stream containing glucose fed into the bioreactor (kg h^{-1})
 W : Weight of liquid inside the bioreactor (kg)
 S_{in} : Substrate concentration in the inflow (g kg^{-1})
 OTR : Oxygen transfer rate from the gaseous to the liquid phase (g/kg/h)
 CTR : Carbon dioxide transfer rate from the liquid to the gaseous phase (g/kg/h)

In the *Ecoli* process, the parameters of interest are the specific growth rates (μ_1, μ_2, μ_3) and the specific substrate uptake rate (q_s), which represent the dynamic behaviour of this process. These are the parameters that are difficult to measure in real-time on the culture and for whom an ANN model could be of great interest. For the ANN validation, the mathematic equations 7 and 8 will be used, together with the heuristics from [26], since they determine the kinetic coefficients from other measurable parameters like X, S, A, O , and C_T .

2.4 Levels of Complexity

The main differences in the complexity of the two case studies arise from:

- Number of the input and output variables;
- Continuity and complexity of the mathematical functions to approximate by the ANN.

The PR kinetic behaviour could be described by equation (6), which means that with only one input variable (x_d : substrate concentration) it is possible to achieve the values of the only kinetic coefficient relevant in this process (A : Specific growth rate).

The *Ecoli* process kinetic behaviour, expressed by equation (9) shows that five variables (X, S, A, O, C_T) can have influence in the values of the three kinetic variables (μ_1, μ_2, μ_3).

The number of input and output variables defines the number of neurons at the input layer and at the output layer, respectively, in an ANN. More inputs and more outputs mean more processing units and more connections for the Artificial Neural Network, which increases significantly the processing burden.

The PR equation does not have discontinuity points, and the variations on the kinetic variables are relatively smooth.

The *Ecoli* process has considerable changes in state variables, from a low initial to a very high biomass and product concentration [24]. Also, the switches from different metabolic states cause discontinuities in the values that are very difficult for an ANN to learn.

3 Artificial Neural Networks

Artificial Neural Networks (ANNs) are based on an analogy with biological brains. Their way of operation provides them with features that are not present in traditional computing machines, such as massive parallelism, distributed computing and representation, adaptability, among many others [12].

The building blocks of ANNs are the artificial neurons (Fig. 1). The weights define the interaction of each neuron with its neighbours. For each neuron, any signal that arrives is multiplied by the weight of the corresponding connection. These values are added with an offset, or bias, to reach the activation of the neuron. The activation function applied over this value grants the non linearity of ANN's behaviour.

ANNs are mainly characterized by the topology of the connections between neurons, also known as architecture, the method of adjusting its connection weights (training algorithm) and the activation function [5], [7]. The neurons in an ANN are typically disposed in layers, with a layer for input neurons that receives the signal from the environment and has no processing associated. There is also an output layer, containing

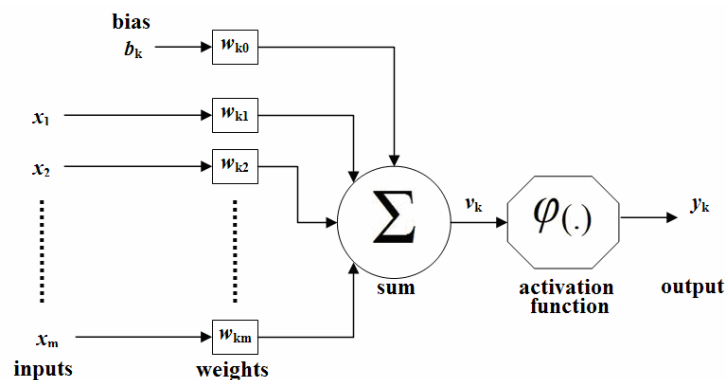


Fig. 1. Artificial neuron structure

neurons with a normal operation, whose outputs are sent to the environment. Internally, between these two layers, other layers of neurons can exist, which are called intermediate or hidden layers. In the same layer, usually all neurons have the same activation function and the same structure of links to other layers of neurons [6].

According to its architecture, an ANN can be classified as a single layer network (it actually has two layers, the input layer and output layer) or multilayer (input layer, one or more hidden layers and the output layer). ANNs can also be classified according to the direction of the internal information flow. If the information flows layer by layer, without cycles, this is a unidirectional or feed-forward network. Otherwise, it is classified as a recursive network. The choice of which model of information flow should be used to represent a process depends on the dynamics and complexity of the problem under analysis [17], [18].

A multilayer ANN has, in principle, a higher ability to solve more complex problems. The greater the number of neurons in each layer and the number of layers, the higher is the degree of complexity of the problems that the network can solve. However, the excessive complexity of a network is often unnecessary and carries an extra processing burden, probably causing overfitting problems. The optimal number of hidden neurons is the smallest number that yields an acceptable prediction error [10]. Therefore, there are mechanisms for constructing and pruning networks, with the aim of building them with only the necessary size for a good performance in learning and generalization of the problem.

In this work, ANN training is done according to the paradigm of supervised learning. There is a data set of known input and output vectors with interrelated data, which the ANN will have to assimilate, through the successive application of these training cases. The training process is repeated while the response of the network does not coincide or not satisfactorily approaches the desired outputs [8].

To ensure the effectiveness of the model, ANNs with three layers (input, hidden and output) were used, since they are able to represent any continuous functional mapping [1], [10]. The number of nodes at the hidden layer could be achieved by practical methods, testing several hypotheses until finding the number that demonstrates best results.

The training algorithm used was *Backpropagation* [10], which is based on the known delta rule, with roots in the optimization of filters [32]. This algorithm is composed of two phases. In the first phase, a stimulus is applied to the network and the error difference between the obtained and the expected output value is registered. In the second phase, that error is propagated backwards through the network, causing an adjustment of the connection weights, thus implementing the learning of the network.

This algorithm has proven to be very effective in most cases but it may, in certain circumstances, show some drawbacks. For example, in learning, where the lowest possible error of training is sought, there may be stagnating points when a local minimum is found. This value is the lowest error in a certain neighbourhood but not the overall minimum. This and other problems that lead the network learning to virtually stagnate can be solved, or at least mitigated, through the use of some heuristics.

One of these heuristics is used by the Resilient Back Propagation (RProp) [23], which is considered the best learning algorithm for networks with various topologies. The basic idea of this algorithm is to update the weights of the network using only the rate of learning and the sign of the partial derivative of the error function of each

weight. This process provides convergence speed, robustness and is easy to implement. Experiences made comparing the use of *Rprop* with the use of the *Backpropagation* algorithm show that the first method performs better in 98% of the cases. If, however, an appropriate pre-processing of the data is made, the performance is almost equal [11], [28].

The learning process could lead to several models presenting all a good learning performance. To choose the best of these models, some selection methods could be used to estimate their generalization capabilities, like cross validation [13]. By training an ANN with a particular data set, there is a risk that the network specializes and adapts its behaviour specifically to the solution of the cases used for training. This leads to loss of generalization and, hence, the loss of capacity to solve new problems not considered in the training data set. To identify the most specialized models, the ANN needs to be validated by applying new cases that are not included in the training set and the responses to those cases should be determined.

4 State of the Art

Nowadays, it is already possible to find a vast set of studies in the area of bioprocesses modelling. For describing fermentation processes, usually three types of models are used: white box, grey box and black box models. These designations classify the models according to the degree of visibility that an observer has on the internal structure of the respective systems. The choice of a particular type generally depends on the existing amount of *a priori* knowledge about the system to be modelled [17,18], the nature of the problem to be solved and on the process variables that could be measured in real-time of the biological experiments.

Traditionally, the whole process is described in a mechanistic way, being the physic behaviour of the process externally visible. Because of this visibility, these models are called white box models. These models are mainly composed by differential equations with kinetics based on the Monod or Monod-derived equations, which describe the cell growth limited by substrate concentrations and often contemplate inhibition by different compounds [5]. These models are empirical and highly non-linear, requiring the determination of both model structure and parameters for each particular case [6, 9, 33]. One of the major applications of these models is process optimization.

The white box approach is used in the work of Veloso *et al.* [31], where the equations of mass balance, a complex kinetic model and an Extended Kalman Observer were used for the on-line determination of biomass concentration. Park and Ramirez [21] proposed an optimal control policy for the optimization of a secreted heterologous protein production process, described by a white-box model. They based their work the Pontryagin's Minimum Principle for determining the necessary conditions for system optimization. Lee and Ramirez [16] established optimal control strategies for maximizing the production of induced foreign protein by recombinant bacteria, by deterministic algorithms that calculate exactly the periods where the cell specific growth rate and protein production are maximized. Some process optimizations require more advanced techniques. For example, Genetic and Evolutionary Algorithms had been used in several works [24, 25, 26, 27]. Mendes *et al.* [19] presented an experimental comparison between Particle Swarm Optimization, Differential Evolution

and real-valued Evolutionary Algorithms for the optimization of fed-batch fermentation processes described by white-box models.

On the other hand, Artificial Neural Networks have proven to be of great value, due to their ability to model nonlinear systems through input-output mapping. ANNs can be constructed from process experimental data, being suitable to use as black box models of biological processes. Lee and Park [15] used an ANN model as a software sensor for monitoring and controlling wastewater treatment process. The problem is that existing measures do not give on-line measurements of nutrient dynamics.

The use of a single ANN to optimize processes does not always bring the best results, even at the level of process function approximation, or at the level of processing burden when the system to approach is too complex and has a great number of inputs and outputs. A possible solution is to divide the problem into several parts, which could be implemented by a set of ANNs or by ANNs combined with other solutions (hybrid model). Chen *et al.* [3] presented an optimization of a fed-batch bioreactor using two cascade recurrent neural networks for process modelling which predict the biomass production based on the feed rate profile. A Genetic Algorithm is then combined with the extended recurrent neural networks model to determine the feed rate profile that maximizes the final quantity of biomass for the fed-batch fermentation process. Another example is the work of Koprinkova *et al.* [14] that focuses on the time delay between the changes in substrate level and the adjustment of the growth rate at the appropriate level in chemostat cultivations. They proposed different feed-forward ANNs with delay elements for both specific growth rate and specific consumption rate estimation as a solution for modelling this behaviour.

When white box and black box components are joined together, a grey box model is obtained. For example, Zuo and Wu [34] proposed a method to semi real-time optimization and control of a fed-batch fermentation system using hybrid neural networks for process modelling and genetic algorithms for determining optimal feeding rates. The hybrid model joins the knowledge-based part (differential equations, from mass balance of the system) and the black box part (ANN, for dynamic behaviour of the specified system). Peres *et al.* [22] presented a method of expressing and combining different types of knowledge, consisting in a gating system that joins knowledge modules that could be white, black or gray box models. By this way, it is possible to combine the capacity of the three types of models to achieve a better prediction. A similar method is applied by Simutis *et al.* [29], which also highlighted the importance of starting with a clear definition of the problem and a corresponding quantitative objective function.

Given all this work in the area, it is important to establish what is intended for the current work stating what is the problem to solve and what is the difference between this and other solutions for related problems:

1. The objective is to optimize a fed-batch bioreactor, increasing product formation. A similar objective could be found in the work of Chaudhuri and Modak [2], but this method should be applicable to any strain of microorganisms;
2. The control variable is the feed profile;
3. Real feed profiles exist, considered like near optimal solutions for maximum cell growth and product formation, obtained by optimization methods;
4. Given 1, 2 and 3, the aim of this chapter is to present a computational tool that allows the use of ANNs in the modelling of these processes.

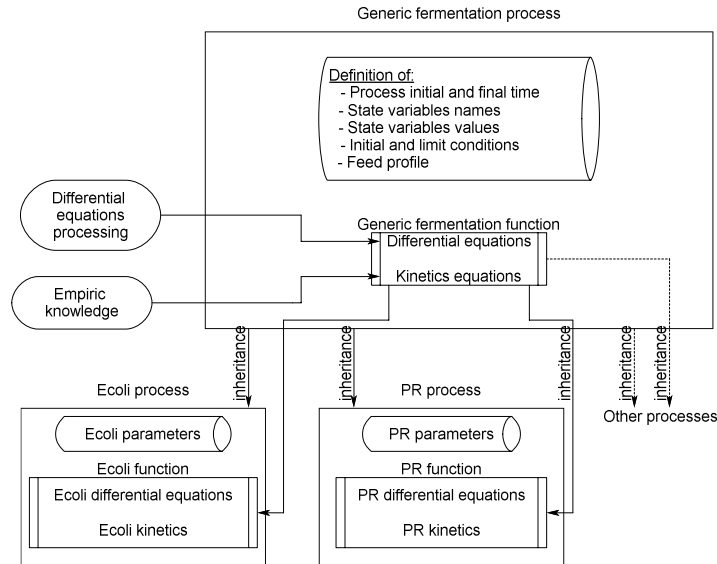


Fig. 2. Schema of the deterministic (white box) model for fermentation processes

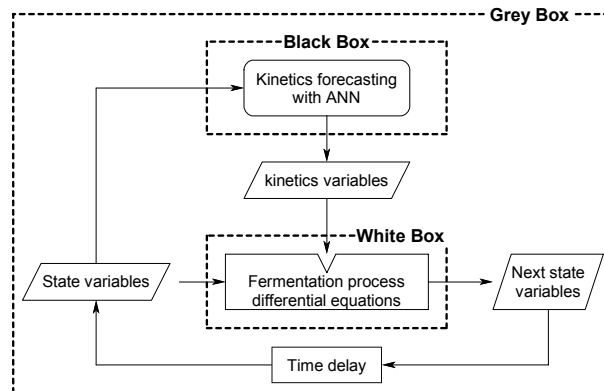


Fig. 3. A schematic representation of the generic grey box model for the simulation of fermentation processes

The starting point was the existing knowledge about the processes, in the form of white-box models composed of ordinary differential equations that describe the fermentation processes behaviour (Fig. 2).

To establish the optimum feed profile it is necessary to have a reliable model for the prediction of process state and kinetic variables. These last ones are not possible to measure directly in the culture experiment, and so they are obtained through other variables. Moreover, the structure of the kinetic equations is often obtained with empirical methods that are dependent of the culture strain in study. To avoid this limitation, an

ANN is used to simulate the kinetic behaviour for each process, being the number of inputs and outputs defined by existing process data.

Establishing a structure that could be applied to any fermentation process, the objectives for process optimization becomes too general to be resumed in an objective function like the one proposed by Simutis *et al.* [29]. Instead of achieving an optimal feed profile for a particular process, the objective was to give biologists a tool with which it is possible to construct a reliable model for any fermentation process. In this tool, it should be possible to introduce feed profiles and to test their contribute in process yield maximization.

In this way, the existing mechanistic knowledge is combined with a component implemented with a black box. The result is a hybrid grey box (Fig. 3).

5 The FerMoANN Tool

The FerMoANN computational tool proposed in this work should allow:

- To produce process data off-line, that can be used to test the grey box model without having to collect data from the real experiment, decreasing developing time;
- To select one of the available processes and respective white-box models. It should also be possible to automatically add new modules for different processes;
- To evaluate several ANN configurations, selecting the one that better approximates the process kinetic behavior, i.e the one that shows lower Root Mean Square Error in their learning and validation. In this evaluation, it should be possible to choose the training parameters (e.g. training algorithm), inputs and outputs, validation method, etc.;
- To use trained ANNs to model a given process;
- To visually check the modeling results, and allow testing diverse feed profiles with the aim of finding the one that maximizes the process yield.

The general guidelines established for the code design were:

- Modularity, that was achieved with the division of the software in functional blocks, which can operate independently one of each other, ensuring also better scalability and easing the process of incorporating new software modules;
- Generalization, that was ensured giving the application the ability to model any fermentation process, not merely a set of pre-defined processes;
- Functionality, that was obtained from the previous two principles and ensuring a programme with a simple and user friendly interface that can easily be understood.

The use of ANNs reliefs the researcher from defining mathematically the whole process, leaving the kinetic component as a black box, which is obtained through the model inference from data coming from the microorganism's culture.

The FerMoANN was built using mainly three software packages written in Java by the authors (*ann*, *bio* and *annFerm*):

- The *ann* package provides the necessary features for a generic application using feedforward ANNs, following the paradigm of supervised learning [4, 27]. In this

module, it is possible to set up the implementation of the network at various levels, such as its structure (characterization of the layers), the adjustment of weights (definition of the initial values and adjustment during training), the definition of the data to be presented to train the ANN, the selection of the best training algorithm for the case study, the type of model validation to use, etc.

- The *bio* package refers to fermentation processes modeling by white box where kinetic models are based on Monod-type equations [26]. With this module the behaviour of various pre-established cultures can be simulated offline. Every culture is associated with a fermentation white box model with its own characteristics where, in addition to other parameters, a set of differential equations and the kinetic equations are included [23, 25]. The fermentation processes here considered are the production of recombinant proteins with *S. cerevisiae* and *E. coli* (see section 2) and the overall schema is given in Fig. 2.
- The *annFerm* package provides the bridge between *ann* and *bio* packages. It is in this package that the fermentation processes models are adapted to allow estimation of kinetic coefficients by artificial neural networks. This adaptation results in a grey box model (Fig. 3).

5.1 Development

The FerMoANN was split into two modules with different interfaces. One module is for white box modelling, that is used to produce data for the other module, the grey box model, thus accelerating the development of the application. Each interface contains a set of sections that provide specific functionalities. In the panels, text areas have been included to display information to the user when necessary.

At the programming level, there was a separation between the part concerning the visual elements of the interface and the part responsible for carrying out the main functions of the program (core functionality). All classes for the interface elements have "Intrfc" as a suffix for better distinction of classes that belong to the core.

The white box interface includes several sections for the generation of feed profiles, for the generation of training data, for combination of examples, for the introduction of noise and also for drawing relevant graphs (Fig. 4):

Feed profiles generation. The feeding profiles, i.e. the amount of substrate to feed to the culture at each time, represent the inputs of the simulation. Through this panel multiple profiles can be created with constant, linear, random, saw, wave or sinus functions. Using different feeds to create data allows a broader spectrum of training examples.

White box process simulation. This simulation uses Monod-type models for generating the kinetics variables, which are then introduced into the ODE models. For each feeding profile, a distinct simulation can be conducted resulting in the values for state and kinetic variables over time. The data from this panel is used for ANN training and a comparison between data that is obtained by the model and with ANNs can be performed.

Mixing examples. A number of files with simulation results can be handled, and used to create new files through composition, cutting, interpolation or sampling data. These

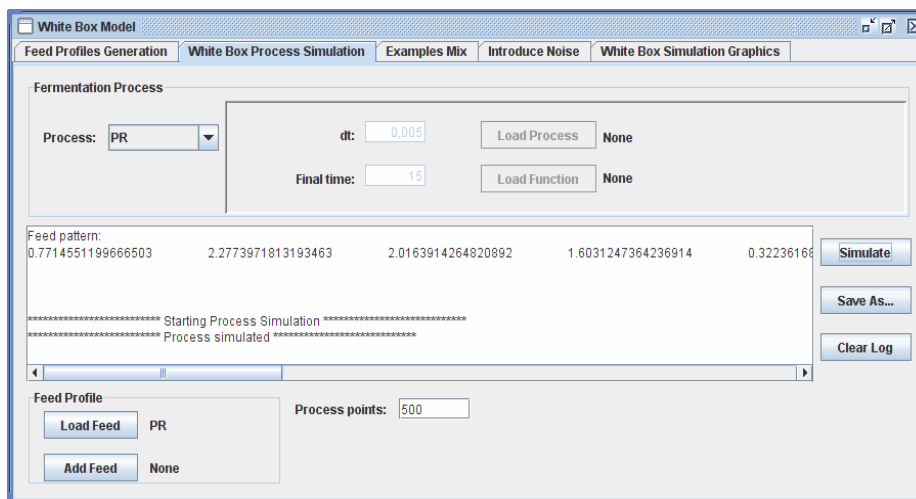


Fig. 4. White box interface functionalities

can create more comprehensive sets of examples or decrease the size of some collections without losing generality.

Adding noise. The introduction of noise on data serves to create data sets that increase the generalization abilities of the ANN model. This can be useful when the set of available data is not rich enough to provide a good training process for the ANN.

White box simulation graphs. The graphs allow for a better visualization of the data generated by the white-box mathematical model. The preview is done in two separate graphs for the state and kinetics variables. It also allows visual comparisons between results of different simulations.

The grey box interface is intended to determine the best ANN configuration that fits the problem under study and the application of those ANNs to simulate the process. Here a number of functionalities can be found, namely the ANN's evaluation, the ANN's training, the process simulation using a trained ANN and the evaluation and training graphs (Fig. 5):

ANN's evaluation. This tool can assess various configurations of ANNs, varying the input and output variables, the number of nodes in the hidden layer (all ANNs are feed-forward with one hidden layer, completely connected), the training algorithm and the number of training epochs. Each model is evaluated by its training error and generalization error in a validation set. It is thus possible to determine which parameter combinations will lead to more accurate models.

ANN's training. An ANN with a given configuration (typically the one that resulted from the above study) can be trained. It then can be used to predict the kinetic variables, integrated within the grey box model.

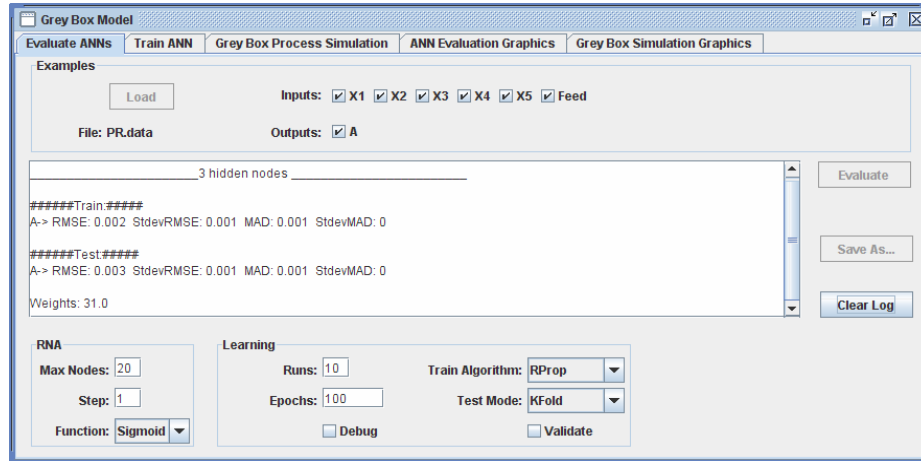


Fig. 5. Grey box interface functionalities

Grey box process simulation. This section enables the simulation of the fermentation process. Trained ANNs calculate the kinetic variables from the values of the state variables in a particular time point and differential equations represent the mass-balance equations that determine the time evolution of the values of the state variables.

ANN evaluation graphs. These graphs show the errors of the evaluation of ANN configurations. They provide information on the settings, data and variables that were used in the training. The graphs allow a better interpretation of the error curves (both in training and validation sets).

Grey box simulation graphs. These graphs show data resulting from grey box model simulations. They show the time evolution for each state and kinetic variable. It is also possible to compare two simulations and thus examine, for example, the difference between the curves obtained with a grey box model and the ones obtained with white box models.

5.2 Experiments and Results

For a demonstration of the software capabilities, two case studies were used: PR and *Ecoli* processes, already described in section 2.

The first step of the modelling process was the feeding profiles generation that allow a wide spectrum of examples to be achieved. The profiles range from optimized ones (using Evolutionary Algorithms [24]), random profiles, constant profiles, etc. These files were then used to simulate the process with white box models, thereby obtaining one data set for each profile. Fig. 6 shows the results for the PR case study with two distinct profiles.

The state variables ($x_1, x_2 \dots$) are those involved somehow in the process, and for these variables, it is possible to obtain culture measures at different time points. The

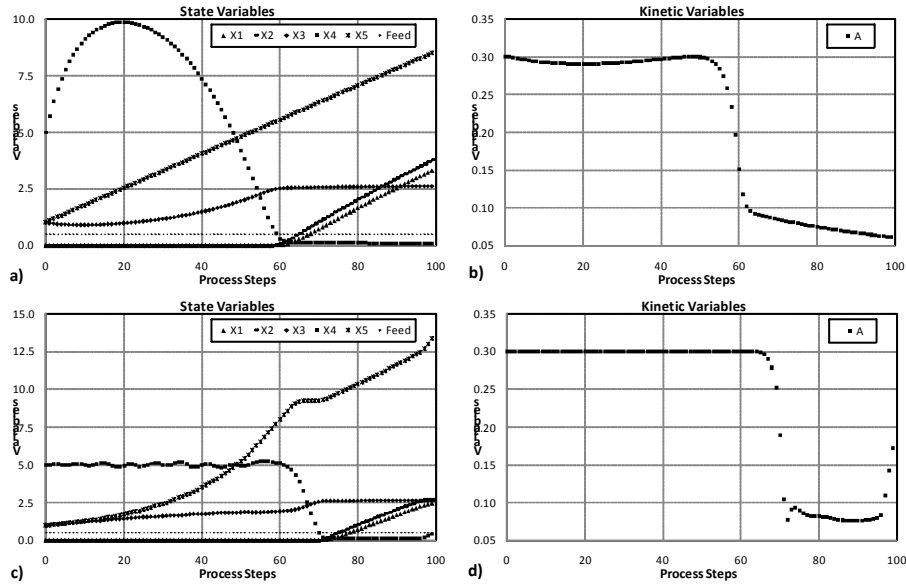


Fig. 6. Example of white box modelling simulation for PR process. a) and b) Constant feed profile; c) and d) Feed profile optimized using Evolutionary Algorithms.

blocks on the left show the graphs for each state variable. In the blocks on the right, the kinetic variable (A) curve is shown, which represents the dynamics of the process. In Fig. 6, only the cases of PR process simulation with optimized and constant feed profiles are shown, but it is also possible to use other feed profiles (e.g. saw, sinusoidal, random and ramp functions). The process steps are the simulation points used in the interpolation (100 for PR). In Fig. 7, the graphs for the *Ecoli* case study are shown.

22 different feed profiles were used for the PR process with values between 0 and 3 L/h (given that this was assumed as the maximum capacity of the feeding device), considering 4 random profiles, 4 linear, 3 constant, and 11 optimized. With each of those 22 profiles the simulation of the PR process was performed.

Using the data composition interface it was possible to group all simulations forming a single global training set for the ANN. In principle, the PR process will not be too difficult to simulate, because of its simplicity. So, from the global data set a sample of 150 values was retrieved, which make the ANN training data set.

For the *Ecoli* case study, 21 different feed profiles were used: 5 random, 5 linear, 4 constant and 7 optimized. Being a more complex process, 500 examples were extracted from the entire data set for ANN evaluation.

If all the configuration parameters are considered, the universe of possibilities for different network configurations is enormous. To limit this universe it was stipulated that certain parameters remain unchanged, such as training algorithm (RProp), the activation function (sigmoid), the number of epochs (1000) and the validation mode (KFold). This configuration grants a sufficient base for a good ANN training.

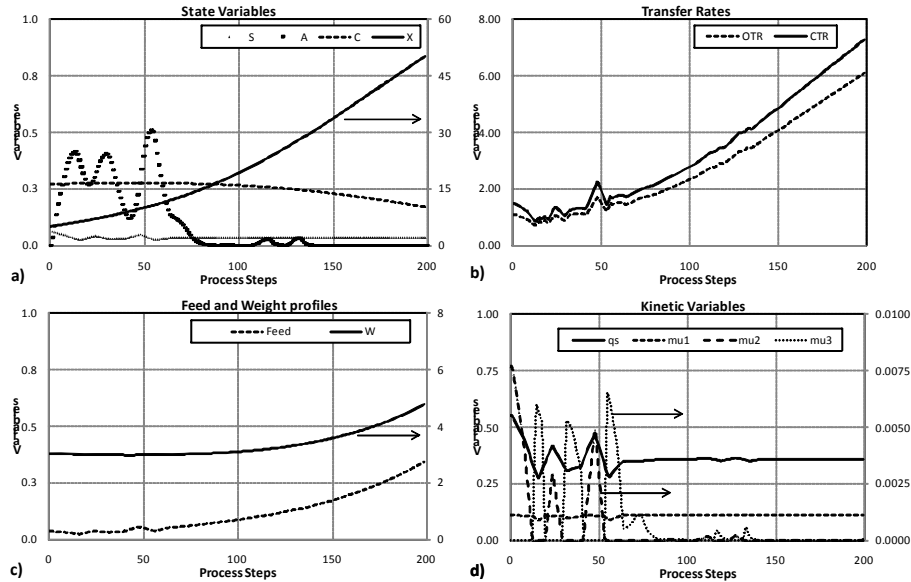


Fig. 7. Example of white box simulation for *E. coli* process with optimized feed profile

For each evaluation, the parameters changed were the state variables used as ANN inputs and the number of nodes in the ANN hidden layer. An ANN with 20 nodes in hidden layer should be able to simulate very complex cases, so every value between 0 and 20 nodes was tested. To achieve the best combination of input variables (state variables) several tests took place ranging from the inclusion of all variables in the model to the inclusion of only one of them. Supposedly, the more input variables are included, the more the network is capable to determine the outputs (kinetic variables), even for processes with complex characteristics.

For the PR process, the first test was made with each state variable separately. In Fig. 8 we show two examples of evaluation results. This figure shows the error over the training and test sets during the evaluation. The measures of error used were the Root Mean Square Error (RMSE) and Mean Absolute Deviation (MAD). The results for both measures have been proportionately identical, which leads to assume that conclusions can be drawn using only RMSE results.

In the horizontal axis the number of hidden nodes for each ANN structure evaluated is shown. The training error shows how the ANN assimilated the process in question. The test set error shows the adaptation of each network, after trained, to new examples from the same biological process. The best solution is the one that shows lower values of error in both training and test sets. For example, in Fig. 8 it is possible to see that process simulation with the input variable x_4 presents lower values of error than when the input variable x_5 is used. It may also be noted that, using x_4 , a network with only one node in the inner layer is sufficient to simulate the process, since the increase in the number of nodes does not bring significant gains in reducing the errors.

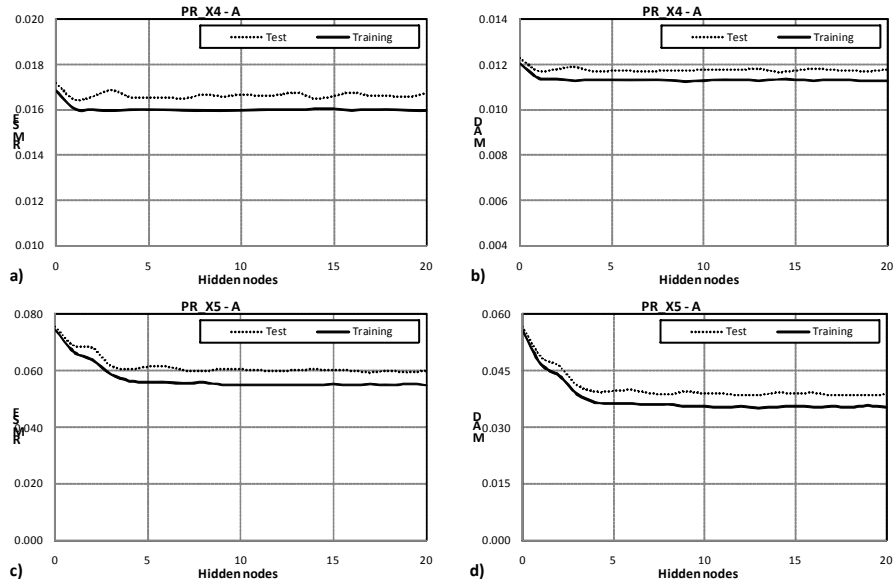


Fig. 8. Result of evaluation for the PR process only with x_4 (a)(b) and only with x_5 (c)(d) state variables as inputs

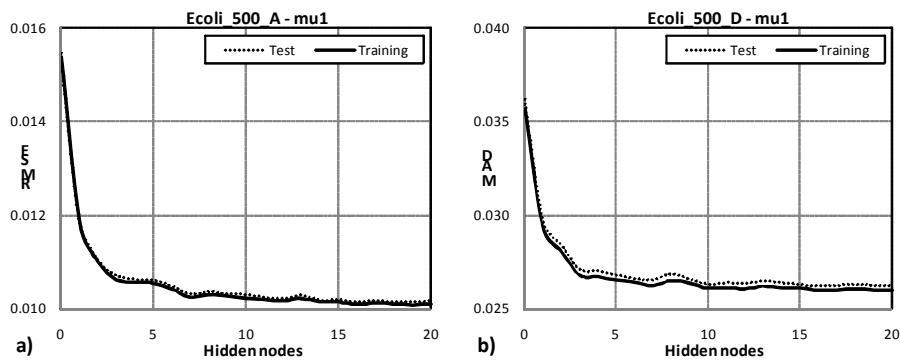


Fig. 9. Evaluation of $mu1$ forecasting with the state variable A (a) and with state variable D (b) as inputs

In the *Ecoli* process, the amount of kinetic variables increases the complexity of the selection method. All possibilities considered for the PR case are now multiplied by 4 (the number of *Ecoli* kinetic variables). For an easier interface implementation, kinetic variables were named $mu1$, $mu2$ and $mu3$. For example, regarding $mu1$, if we want to test the influence of each state variable in its prediction we would obtain 10 graphs (the number of *Ecoli* state variables). Consider, for instance, the RMSE graphs for prediction of $mu1$, when only the A or the D state variables are used as inputs, as shown in Fig. 9.

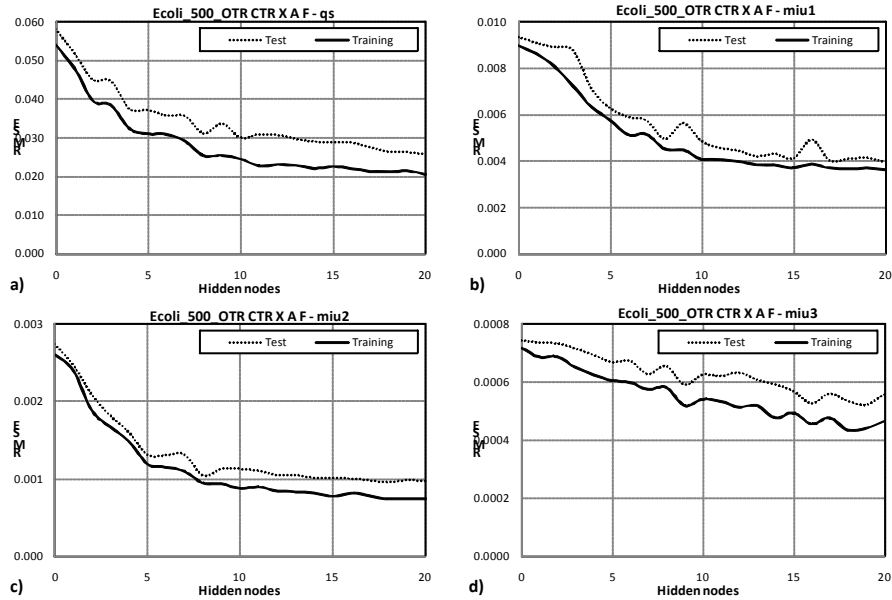


Fig. 10. Evaluation of *E. coli* kinetic variables prediction with the state variables OTR, CTR, X, A and F as inputs

```

Inputs: [X4]
Outputs: [A]

WeightMatrix:
0.5227209185050394 1.2984096761629806 XXXXXXXX
-0.4203806279909669 -1.2952322001992079 XXXXXXXX
0.8013370829170636 1.0525967185615312 XXXXXXXX
-2.305726654130054 -2.8398798639075307 XXXXXXXX
1.8710979486974038 2.697814107979465 XXXXXXXX
1.206495516731544 2.374242632271765 XXXXXXXX
1.4481105476809775 18.892957693779234 2.0763012218131625 -0.4949892490908072 0.952993085287861 -4.44412264392531 49.66497026492257 8.19371037568122 X]

Activation function for each node:
sigmoid,1.0][sigmoid,1.0][sigmoid,1.0][sigmoid,1.0][sigmoid,1.0][sigmoid,1.0][sigmoid,1.0]

Layers:
[0]
[1 2 3 4 5 6]
[7].
    
```

Fig. 11. Trained ANN. Inputs: state variables; Outputs: kinetic variables; Weight Matrix: nodes connections weights; Activation function: activation function for each non-input node. Layers: number of layers and number of nodes by layer.

It is possible to create any combination of input variables. For example, in the *E. coli* case study, the variables OTR, CTR, X, A and F can be chosen to estimate the 4 kinetic variables, and the result is shown in Fig. 10.

With the possibility of changing the state variables in the simulation, the researcher can confirm whether the variables that initially were thought to be the most suitable for the prediction are in fact those with better results. This analysis may even lead to the proposal of new hypotheses and a different view of the processes.

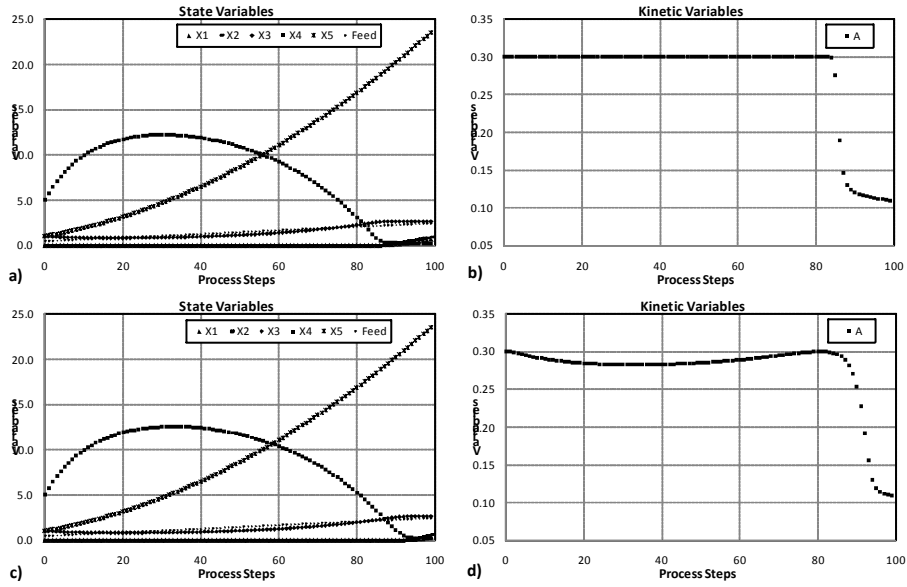


Fig. 12. PR process simulation using a linear feed profile: with grey box (above); with white box (below)

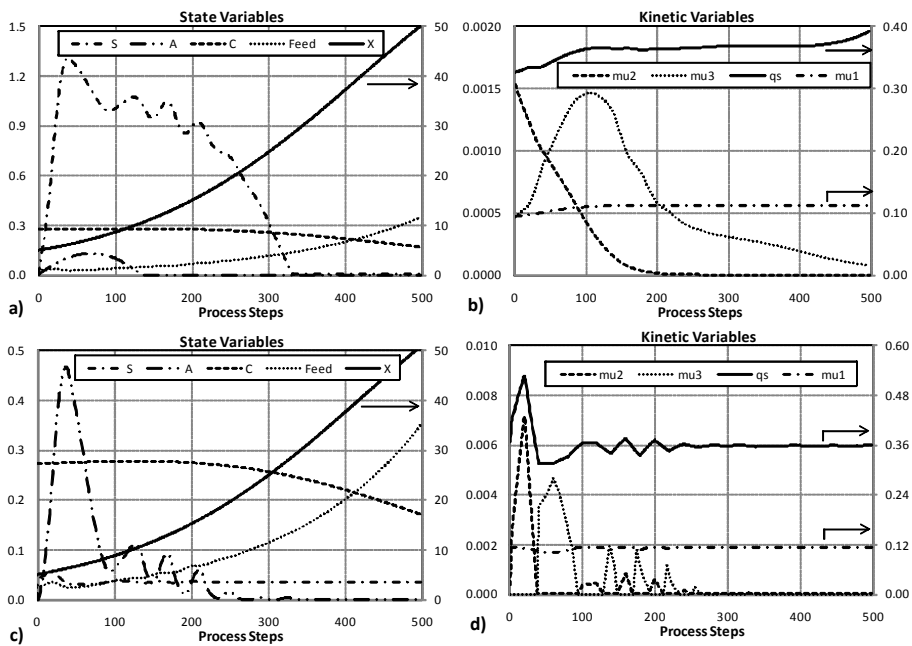


Fig. 13. *E. coli* process simulation using an optimized feed profile. With grey box (above); with white box (below).

The more different combinations are tested the better, since the combination of input variables does not present linear results. For example, the junction of two state variables may result in a worst prediction than the estimation made with only one of them separately. Similarly, a variable that does not produce great results when used alone, or in certain combinations, could produce excellent results when used in other combinations. Once the best combination of input variables and the best number of hidden nodes is determined, an ANN is trained to be further used for the process simulation. For example, in the PR case study, using only the state variable x_4 and 6 hidden nodes, the ANN would be the one given in Fig. 11.

The ANNs are then trained and used to simulate the process, now with a grey box model. Using the ANN in Fig. 11, and a linear feed profile, the results for the PR process simulation are shown in Fig. 12.

In Fig. 12 it is shown that the performance of the model was quite satisfactory, since the curves of the kinetics and state variables of the two models are very similar, even using only 150 training cases. The simulation for the *Ecoli* case study was also satisfactory (Fig. 13), although it is evident that the ANNs have some difficulties to model functions with abrupt variations, as for example the variable q_S at the beginning of the process. For this simulation, the ANN used has 10 hidden nodes, it was trained with the input variables OTR, CTR, X, A and F and using an optimized feed profile.

6 Conclusions

In the work described in this chapter a software tool, the FerMoANN, was developed to aid in the modelling of fed-batch fermentation processes that can be used to create ANN based models from data coming from real experiments. It is possible to shape a process without the need to know the mathematical description of its mechanism, since learning is made directly from examples created from experimental data, showing the potential of the application of ANNs in this type of problems.

The application makes possible to introduce new fermentation processes at runtime, enabling the application for a multitude of cases. It not only seeks to provide a solution to a specific case, but serves as a platform for a layman in ANNs to evaluate and use multiple models for each culture.

This is possible since user interfaces are implemented in a simple and intuitive way. The modularity of the FerMoANN makes it scalable; the main modules can easily be used for future work. The core implementation of the functionalities is detached from the user interface, to make code understanding and re-use easier.

The implementation of several graph displays allows making visual comparisons between results, providing easier analysis of the factors influencing the course of the process. Thus, it is possible to identify potential optimizations that can be made, such as to change the feed profile. These optimizations can be tested within the tool, thereby avoiding some real experiments that bring, necessarily, increased costs for research.

In the future, a major aim is to improve the capabilities of FerMoANN with new functionalities. An important concern is to create an interface to make the introduction of new processes easier, without having to write Java code. A generic objective function that could use the model to automatically define an optimization curve for some parameters (like feed rate profile) could also be implemented.

References

1. Bishop, C.M.: *Neural Networks for Pattern Recognition*. Oxford University Press, USA (1995)
2. Chaudhuri, B., Modak, J.M.: Optimization of Fed-Batch Bioreactor Using Neural Network Model. *Bioprocess Engineering* 19, 71–79 (1998)
3. Chen, L., Nguang, S.K., Chen, X.D., Li, X.M.: Modelling and Optimization of Fed-Batch Fermentation Processes Using Dynamic Neural Networks and Genetic Algorithms. *Biochemical Engineering Journal* 22, 51–61 (2004)
4. Cortez, P.A.: *Nature Inspired Models for Time Series Prediction*. PhD Thesis. University of Minho (2002)
5. Coulman, G.A., Stieber, R.W., Gerhardt, P.: Dialysis Continuous Process for Ammonium-Lactate Fermentation of Whey: Mathematical Model and Computer Simulation. *Appl. Env. Microbiol.* 34(6), 725–732 (1977)
6. Fan, G., Wolkowicz, G.S.K.: Analysis of a Model of Nutrient Driven Self-Cycling Fermentation Allowing Unimodal Response Functions. *Discrete and Continuous Dynamical Systems* 8(4), 801–831 (2007)
7. Fausett, L.: *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice-Hall, US (1994)
8. Fyfe, C.: *Artificial Neural Networks and Information Theory*. University of Paisley (2000)
9. Georgiev, T., Ratkov, A., Kristeva, J., Ivanova, V., Dimitrova, A., Manchev, A., Ratkov, B.: Improved Mathematical Model of L-Lysine Fermentation Process Based on Manure. In: *24th Int. Conf. Information Technology Interfaces IT/ 2002*, Cavtat, Croatia, pp. 24–27 (2002)
10. Haykin, S.: *Neural Networks – A Comprehensive Foundation*, 2nd edn. Prentice-Hall, New Jersey (1999)
11. Igel, C., Toussaint, M., Weishui, W.: Rprop Using the Natural Gradient. In: *Trends and Applications in Constructive Approximation*. International Series of Numerical Mathematics, vol. 151, pp. 259–272. Birkhäuser Verlag (2005)
12. Jain, A.K., Mao, J., Mohiuddin, K.M.: Artificial Neural Networks: A Tutorial. *IEEE Computer* 29(3), 56–63 (1996)
13. Kohavi, R.: A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. In: *International Joint Conference on Artificial Intelligence*, pp. 1137–1143 (1995)
14. Koprinkova, P., Petrova, M., Patarinska, T., Bliznakova, M.: Neural Network Modelling of Fermentation Processes: Specific Kinetic Rate Models. *Cybernetics and Systems: An International Journal* 29, 303–317 (1998)
15. Lee, D.S., Park, J.M.: Neural Network Modelling for On-line Estimation of Nutrient Dynamics in a Sequentially-operated Batch Reactor. *Journal of Biotechnology* 75, 229–239 (1999)
16. Lee, J., Ramirez, W.F.: Optimal Fed-Batch Control of Induced Foreign Protein Production by Recombinant Bacteria. *AIChE Journal* 40(5), 899–907 (1994)
17. Levišauskas, D., Tekorius, T.: Model-Based Optimization of Fed-Batch Fermentation Processes Using Predetermined Type Feed-Rate Time Profiles. A Comparative Study. *Information Technology and Control, Kaunas, Technologija* 34(3), 231–236 (2005)
18. Mandic, D.P., Chambers, J.A.: *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*. John Wiley & Sons, Chichester (2001)

19. Mendes, R., Rocha, M., Rocha, I., Ferreira, E.C.: A Comparison of Algorithms for the Optimization of Fermentation Processes. In: Proceedings of the 2006 IEEE Conference on Evolutionary Computation, vol. 7, pp. 7371–7378. IEEE Press, Los Alamitos (2006)
20. Oliveira, R.: Combining First Principles Modelling and Artificial Neural Networks: A General Framework. *Computers and Chemical Engineering* 28, 755–766 (2004)
21. Park, S., Ramirez, W.F.: Optimal Production of Secreted Protein in Fed-Batch Reactors. *AIChE Journal* 34(9), 1550–1558 (1988)
22. Peres, J., Oliveira, R., Azevedo, S.F.: Knowledge Based Modular Networks for Process Modelling and Control. *Computers and Chemical Engineering* 25, 783–791 (2001)
23. Riedmiller, M.: Rprop – Description and Implementation Details. Technical Report. Cite-seer (1994)
24. Rocha, M., Neves, J., Rocha, I., Ferreira, E.C.: Evolutionary Algorithms for Optimal Control in Fed-Batch Fermentation Processes. In: Raidl, G.R., Cagnoni, S., Branke, J., Corne, D.W., Drechsler, R., Jin, Y., Johnson, C.G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G. (eds.) *EvoWorkshops 2004*. LNCS, vol. 3005, pp. 84–93. Springer, Heidelberg (2004)
25. Rocha, M., Neves, J., Veloso, A., Ferreira, E.C., Rocha, I.: Evolutionary Algorithms for Static and Dynamic Optimization of Fed-Batch Fermentation Processes. In: Ribeiro, B., Albrecht, R.F., Dobnikar, A., Pearson, D.W., Steele, N.C. (eds.) *Adaptive and Natural Computing Algorithms*, pp. 288–291. Springer, Heidelberg (2005)
26. Rocha, I.: Model-Based Strategies for Computer-Aided Operation of a Recombinant *E. coli* Fermentation. Phd Thesis, University of Minho (2003)
27. Rocha, M.: Optimization of Connectionists Learning Models With Genetic and Evolutionary Computation. Phd Thesis, University of Minho (2004)
28. Rojas, R.: *Neural Networks – A Systematic Introduction*. Springer, Heidelberg (1996)
29. Simutis, R., Oliveira, R., Manikowski, M., Azevedo, S., Lübbert, A.: How to Increase the Performance of Models for Process Optimization and Control. *Journal of Biotechnology* 59, 73–89 (1997)
30. Stanbury, P.F., Whitaker, A.: *Principles of Fermentation Technology*. Pergamon Press, Oxford (1984)
31. Veloso, A.C., Rocha, I., Ferreira, E.C.: On-Line Estimation of Biomass in an *E. coli* Fed-Batch Fermentation. In: Proceedings of the 2nd Mercosur Congress on Chemical Engineering – ENPROMER and 4th Mercosur Congress on Process Systems Engineering, Rio de Janeiro, Brazil (2005)
32. Widrow, B., Sterns, S.D.: *Adaptive Signal Processing*. Prentice-Hall, New York (1985)
33. Zheng, Y., Gu, T.: Analytical Solutions to a Model for the Startup Period of Fixed-Bed Reactors. *Chemical Engineering Science* 51(15), 3773–3779 (1996)
34. Zuo, K., Wu, W.T.: Semi-realtime Optimization and Control of a Fed-Batch Fermentation System. *Computers and Chemical Engineering* 24, 1105–1109 (2000)