Available online at www.sciencedirect.com
SCIENCE DIRECT®
Control Engineering Practice 12 (2004) 1073–1090
CONTROL ENGINEERING PRACTICE
www.elsevier.com/locate/conengprac

Automatization of a penicillin production process with soft sensors and an adaptive controller based on neuro fuzzy systems

Marcos J. Araúzo-Bravo[a,b,*], José M. Cano-Izquierdo[c], Eduardo Gómez-Sánchez[d], Manuel J. López-Nieto[e], Yannis A. Dimitriadis[d], Juan López-Coronado[c]

[a] *Department of Biochemical Engineering and Science, Faculty of Computer Sciences, Kyushu Institute of Technology, Iizuka 820-8502, Fukuoka, Japan*
[b] *Department of Electromechanical Engineering, Superior Polytechnical School, University of Burgos, Avenida Cantabria S/N, Burgos 09006, Spain*
[c] *Department of Systems Engineering and Automatic Control, School of Industrial Engineering, Polytechnic University of Cartagena, Campus Muralla del Mar, Cartagena 30202, Murcia, Spain*
[d] *Department of Signal Theory, Communications and Telematics Engineering, School of Telecommunications Engineering, University of Valladolid Camino del Cementerio S/N, Valladolid 47011, Spain*
[e] *Process Technology Department, Antibióticos S.A.U., León, Spain*

Received 26 November 2002; accepted 14 November 2003

Abstract

This paper addresses the automatization of a penicillin production process with the development of soft sensors as well as Internal Model Controllers (IMC) for a penicillin fermentation plant using modules based on FasArt and FasBack neuro-fuzzy systems. While soft sensors are intended to aid the human supervision of the process currently being conducted at pilot plants, the proposed controller will make the automatization feasible and eliminate the need for human operator. FasArt and FasBack feature fast stable learning and good MIMO identification, which makes them suitable for the development of adaptive controllers and soft sensors. In this paper, these modules are evaluated by training the neuro-fuzzy systems first on simulated data and then applying the resulting IMC controllers to a simulated plant. Moreover, training the systems on data coming from a real pilot plant, and evaluating the controller performance on the same real plant. Results show that the trend of reference is captured, thus allowing high penicillin production. Moreover, soft sensors derived for biomass, viscosity and penicillin are very accurate.

In addition, on-line adaptive capabilities were implemented and tested with FasBack, since this system presents learning guided by error minimization as new data samples arrive. With these features, adaptive IMC controllers can be implemented and are helpful when dynamics have been poorly learned or the plant parameters vary with time, since the performance of static models and controllers can be improved through adaptation.

Keywords: Intelligent knowledge-based systems; Process control; Adaptive control; Fuzzy hybrid systems; Soft sensing

1. Introduction

Biochemical products, especially penicillin, have an important added value that continues pushing the research in several fields related to their production. For example, enhancements in penicillin producing cultures improved the productivity from milligrams per liter in the original strain of Fleming *Penicillium notatum* to more than 30 g/l of *G* penicillin. More than 400 fungi strains with the capability to produce penicillin have been classified and multiplied by generating random mutants (Paul, Kent, & Thomas, 1993) and using genetic engineering (Agrawal, Deepika, & Joseph, 1999). However, although new strains have higher productivities, fermentations carried out with seeds of these strains must take place under very controlled conditions, hence justifying research in other

*Corresponding author. Department of Electromechanical Engineering, Superior Polytechnical School, University of Burgos, Avenida Cantabria S/N, Burgos 09006, Spain. Tel.: +34-947-25-90-65; fax: +34-947-25-89-10.
E-mail addresses: marara@ubu.es (M.J. Araúzo-Bravo), josem.cano@upct.es (J.M. Cano-Izquierdo), edugom@tel.uva.es (E. Gómez-Sánchez), mlopez@antibioticos.it (M.J. López-Nieto), yannis@tel.uva.es (Y.A. Dimitriadis), jl.coronado@upct.es (J. López-Coronado).

0967-0661/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/j.conengprac.2003.11.002

fields. In particular, this paper will address the development of software sensors to aid the human supervision of the process, as well as the development of automatic controllers that may eventually alleviate the need for human supervision of the process.

Software sensors are important in the penicillin production process for several reasons. The supervision of the fermentation process must maintain certain variables within strict limits since biological systems are highly sensitive to abnormal changes in operation conditions. Supervision is also essential because of the restrictions imposed by regulatory authorities to allow the commercialization of the biochemical products. These authorities, such as FDA in the USA, demand proof that operations have adhered to procedures that guarantee chemical consistency (Lennox, Montague, Hiden, Kornfeld, & Goulding, 2001). To carry out this supervision, important variables must be monitored. In addition, if automatic controllers are to be derived, they may benefit from the estimation of important variables, not measured on-line.

Although hardware sensor technology has improved considerably, many variables are still monitored through laboratory analyses. These are expensive and involve considerable delays. While useful for a posteriori analysis of the process, they are inadequate for on-line supervision. Developments in the area of advanced bioprocess control have demonstrated the applicability of inferential estimation of bioprocess state variables from secondary variables that can be easily monitored on-line, yielding the so-called software sensors. Several approaches have been applied, such as stochastic and statistical methods (Cunha, Glassey, Montague, Alabert, & Mohan, 2002; McAvoy, Su, Wang, & He, 1992) neural networks (Linko, Zhu, & Linko, 1999; Warnes, Glassey, Montague, & Kara, 1998), fuzzy logic (Havlik & Lubbert, 1992) and neuro-fuzzy systems (Araúzo-Bravo, Gómez-Sanchéz, Dimitriadis, Cano-Izquierdo, & López-Coronado, 1999b; Cano-Izquierdo, Dimitriadis, Araúzo-Bravo, Abajo-Manzano, & López-Coronado, 1996).

In addition to monitoring, this paper also addresses the development of an automatic controller for the penicillin production process at Antibióticos, S.A.U. pilot plant (sited at León, Spain). This process is currently supervised by human experts aided by a distributed control system that displays all available measurements and executes control commands. The automatization of the control would facilitate the repeatability of the process, thus reducing the number of anomalous batches, and could also reduce production costs. However, this is a difficult task due to the non-linear dynamics and time varying parameters featured by the penicillin production process. For these reasons, pure mathematical techniques that calculate optimal feeding trajectories by means of a mathematical process model (Jonhson, 1987) were not feasible until recently since it is not easy to model and optimize non-linear systems with slow response unless accurate models and reliable on-line sensors for the state variables are available. A traditional approach in the systems control field is to search for an equivalent linear system. In other words to adjust control parameters according to a linearization of the system transfer function near its equilibrium state, so that the equilibrium state becomes an attractor of the state space of the system (Isidori, 1995). This solution simplifies the problem, but the loss of information may be critical in the penicillin production process.

The great complexity and uncertainty of biological processes require a sophisticated operational logic which cannot easily fit into the mathematical framework of the traditional control approach (Shioya, Shimizu, & Yoshida, 1999). The development of more intelligent methods for practical industrial application is needed. One such valuable set of alternatives is knowledge-based (KB) methodologies that are well suited to model and control biological processes because they allow for the possibility of working with the fragmentary, uncertain, qualitative and blended knowledge typical of such processes. In particular, neural networks (Massimo, Montague, Willis, Tham, & Morris, 1992; Montague, Morris, Wright, Aynsley, & Ward, 1986) fuzzy logic methods (Horiuchi & Kishimoto, 2002) and neuro-fuzzy systems (Araúzo-Bravo, et al., 1999a) have been applied to the penicillin process. Once a KB model of the plant has been derived, it is possible to use model based controllers. Among them, Model Based Predictive Controllers (MBPC) (Richalet, 1993; Camacho & Bordons, 1994) have already been applied to bioprocess problems (Azimzadeh, Galán, & Romagnoli, 2001; Zuo & Wu, 2000), but they require non-linear optimization techniques. Internal Model Control (IMC) structure, on the other hand, permits a rational control design procedure without strong mathematical requirements, allowing for the consideration of control quality and robustness in design decisions (Economou, Morari, & Piasson, 1986). It has also been proved that it can be easily extended to the control of non-linear plants.

In this paper, we are going to use neural networks and fuzzy logic to build KB controllers. Neural networks are well suited for non-linear plant identification without a priori knowledge, and provide a solution to build model and control modules by learning direct and inverse dynamics (Narendra & Parthasarathyu, 1990). This is why they have been widely applied to bioprocess control problems (Boskovic & Narendra, 1995; Montague et al., 1986). Among neural networks, Multilayer Perceptrons (MLPs) (Rosenblatt, 1958) are very popular, but they have several drawbacks, notably that they cannot be used to offer adaptive solutions (Grossberg, 1982). In addition, the knowledge contained in their weights

cannot be expressed in understandable terms (Carpenter & Tan, 1995), while the use of fuzzy logic (Zadeh, 1965) within neural networks allows for the expression of acquired knowledge with rules that resemble those handled by humans.

This paper applies of FasArt and FasBack neuro-fuzzy systems (Cano-Izquierdo, Dimitriadis, Gómez-Sánchez, & López-Coronado, 2001) to build software sensors to estimate important variables in the penicillin production process, such as biomass concentration or viscosity of the broth, and to develop the modules for IMC and adaptive IMC (AIMC) controllers. Both the soft sensors and the controllers are initially validated on a simulated model. Results achieved at a real pilot plant at Antibióticos, S.A.U. will be explained later in the paper. In this sense, the soft sensors are intended to aid the human supervision of the process actually being carried out at this pilot plant, as well as in the production plant. On the contrary, the development of the controller aims at showing the feasibility of the automatization of the control process, that is currently human-operated.

The remainder of the article is organized as follows: Section 2 briefly describes the penicillin production process; Section 3 presents FasArt and FasBack most relevant features; Section 4 describes how soft sensors are derived using these neuro-fuzzy systems; Section 5 describes the IMC strategy pointing out the ways to obtain model and control modules with FasArt and FasBack; Section 6 starts showing experimental results for soft sensors and a controller derived for a simulated plant, under realistic scenarios, and then moves onto results achieved during the usage of these soft sensors and an IMC applied to biomass control at the real pilot plant; Section 7 briefly illustrates the rules extraction feature from the neuro-fuzzy systems used here; finally, Section 8 presents the main conclusions.

2. The penicillin production process and its control

2.1. Process description

The classical penicillin production process is an aerobic fermentation in *fed batch* fermentors made with some *Penicillium* strains, usually *Penicillium chrysogenum* (Nielsen, 1997) that transforms substrates rich in carbohydrates into penicillin. As with other antibiotic production processes (Cunha et al., 2002), the penicillin process operated at Antibióticos involves four stages. The incubation of the culture strains provides the seed that grows in seed fermentors until a stage of maturity is reached. Then, the seed is transferred to a final-stage fermentor. These fermentors are operated in fed-batch mode under standard conditions in order to optimize the synthesis of penicillin. After that, the product is withdrawn by solvent extraction in the downstream.

This paper addresses the control and monitoring of the main fermentation stage. This process is carried out in stirred aerated reactors with air sterilized by filtration. At the beginning, initial substrates (carbon, phosphorus and nitrogen sources) and penicillin-G reaction precursor (phenyl acetic acid) are introduced in the reactor to start the fermentation. The process follows with an exponential growth phase (tropophase), in which the substrate concentration in the fermentor is high. Since penicillin production is a substrate repression process (Revilla, López-Nieto, Luengo, & Martín, 1984) there is almost no production in this phase. When the microorganism has grown enough to reduce the substrate concentration, the penicillin production phase (idiophase) starts. During the entire process, the experts analyze the evolution of variables that are measured either on-line (dissolved oxygen (*DO*), carbon dioxide production rate (*CPR*)) or off-line (viscosity, biomass concentration, phosphorus concentration, penicillin concentration), and decide targets for controlled variables (dissolved oxygen, nitrogen concentration, carbon concentration, phenyl acetic concentration, pH), and therefore manipulate the main control variables (carbon source, nitrogen source, phenyl acetic source). To optimize the penicillin production, they must face the negative effect of non-measurable information (uncertainty in the substrate composition, non-homogeneity in the broth), structure altering phenomena (diauxic growth, changes of metabolic pathways, expression or repression of genes caused by chemical factors or temperature, cell pellets forming), and other unpredictable events such as broth contamination. The most important variables are characterized in Table 1, showing that the three most informative ones (the biochemical state—biomass and penicillin concentration—and the viscosity) cannot be measured on-line, and thus the control of the process has to rely on the information yielded by some physico-chemical variables, such as *CPR* and *DO*, that can be measured on-line. The table also shows that there are constrained variables, some within very strict limits, such as dissolved oxygen.

2.2. Existing monitoring and control implementations

The efficient supervision of the fermentation is hindered by problems in the on-line identification of the process state and by inconsistencies imposed on the system by the complex and poorly understood nature of the media, cultures and raw material, causing inherent process variability. Moreover, the aerobic fermentation associated with penicillin production suffers from problem of limited oxygen transfer capability, that constrains the process control. Thus, the fermentation evolution depends on human operator decisions to

Table 1
Features of the main fermentation variables with respect to: measurement timing (⊙ on-line, • off-line); time evolution (↗ incremental, ↘ decremental → constant, ⇝ oscillant); constraints (⩛ high values, ⩡ low values, ⇒ in a range); and position in the control hierarchy (↓ low level, ↑ high level)

Symbol	Name	Measurement	Evolution	Restriction	Control
Variables associated to the substrate					
S	Carbon source	•	⇝		↑
N	Nitrogen	•	⇝	⩡	↑
PS	Phosphorous	•	↘	⩡	↑
AFA	Phenil acetic acid	•	⇝	⇒	↑
Physico-chemical variables					
RPM	Agitation speed	⊙	↗	⩛	↓
PRE	Pressure	⊙	→	⇒	↓
T	Temperature	⊙	→	⇒	↓
pH	Acidity	⊙	→	⇒	↓
DO	Dissolved oxygen	⊙	↘	⩡	↑
CPR	CO_2 production rate	⊙	↗		
VIS	Viscosity	•	↗	⩛	↑
Biochemical variables					
X	Biomass	•	↗		↑
P	Penicillin	•	↗		↑

adjust the culture growth, minimizing the effects of such limitations. Apart from the mechanical limitations of the fermentor, the operator needs information about the biomass concentration and other physiological parameters of the culture. Several mathematical models (Birol, Undey, Parulekar, & Cinar, 2002; Tiller, Meyerhoff, Sziele, Schügerl, & Bellgardt, 1994; Nielsen & Villadsen, 1992; Nestaas & Wang, 1983; Bajpai & Reuß, 1981; Heijnen, Roels, & Stouthamer, 1979) were developed to handle such information in a systematic way. However, neither of them is adequate to solve the real problems of the penicillin industrial processes carried out at Antibióticos, since the industrial conditions there are much stricter than those considered for the modeling. For all these reasons the process is controlled using the empirical knowledge of the experts as the single strategy. Traditionally, experts choose the cellular growth as the main control variable because is the only one that can be characterized consistently in terms of its effects on production (Mou, 1975). With this approach, the highest production is achieved when the carbon source is controlled after the tropophase in order to maintain a low accumulation of biomass (Montague et al., 1986). Generally, they use a verbal biomass reference, that is tracked through manipulating control variables using empirical knowledge.

2.3. New monitoring and control implementations

It is possible, from a qualitative description offered by experts, (as shown in Table 1), to use empirical knowledge in order to implement an expert system or to tune manually a fuzzy system. Nevertheless, in both cases numerical adjustment of rule parameters is required. Moreover, the success of the resulting systems depends on both rule adjustment to be well done, and the qualitative description to be accurate. On the contrary, the use of neuro-fuzzy systems allows rule parameters to be automatically tuned by a learning process that acquires knowledge from fermentation data. However, to integrate adequately this new control scheme in Antibióticos pilot plant, the data used to derive the models must be generated, while the variables to be controlled must be determined, as well as their reference values. Finally, it must be decided how this new controller interacts with the rest of the control and monitoring systems already existing in the plant.

Input–output data sets are generated with a simulated plant, if the plant to be controlled is also simulated, or collecting data from the real plant otherwise. With these data sets, neural networks are trained to build soft sensors and controllers, as described in Section 4. Once these networks are trained, the knowledge they acquire is stored as a set of fuzzy rules that describe the process modeled. If these rules are expressed in some linguistic form, as shown later in Section 7, they are likely to be similar to the linguistic description of the process offered by the experts.

In addition, in order to automate the control of the plant, a biomass reference trajectory has to be proposed. Note that, since penicillin is a secondary metabolite of the process, controlling it directly is very difficult. This fact has been assumed in many other approaches (Montague et al., 1986; Montague, Morris, & Bush, 1988; Massimo et al., 1992; Willis, Montague, Morris, & Tham, 1991). These conclude that biomass

concentration has to be controlled to optimize penicillin production. The biomass reference trajectory used in our research was suggested by Antibióticos experts, and it can be mathematically expressed (as inspired by Mou (1975)) as follows:

$$X^{ref} = \begin{cases} e^{\mu_{tr} t}, & t \leqslant t_1, \\ e^{\mu_{tr} t} + \Delta X_{id} \frac{1-e^{-\mu_{id}(t-t_1)}}{1+e^{-\mu_{id}(t-t_1)}}, & t > t_1, \end{cases} \quad (1)$$

where ΔX_{id} is the total growth expected in the idiophase, and t_1 is the moment of change from the tropophase to the idiophase. Tracking this reference guarantees high growth rate μ_{tr} during the tropophase stage and low growth rate μ_{id} during the idiophase. This law was used to test controller performance on a simulated plant. Subsequently, to operate with a more realistic reference in the Antibióticos plant, another reference was model inspired by statistical process control (SPC) ideas (Lennox et al., 2001), it was designed a reference that averages the biomass profile of several successful fermentations controlled by experts.

Finally, to implement the new control scheme, the different control tasks (choice of dynamic references and set points, tracking of dynamic references, and regulation of set points) were decoupled through a hierarchical control system, as depicted in Fig. 1. The automation system, which is a distributed control system (DCS) based on OPTO 22, provides an interface to the basic field instrumentation and the gas flow analyzers, and also handles alarms, safety interlocks and logic sequences, as well as basic-level control loops. A FIX DMACS information system is used to collect all measurements from the automation system. It also provides access to results of laboratory analyses, performs the calculations of average values and stores both the current and historical data into the real time database. The Adaptive Internal Model Controller (AIMC) proposed here was implemented with Matlab in a Windows NT workstation placed at the top level of the hierarchy. Thus, AIMC decides the objectives of the low level controllers, which are *stabilizing controllers* based typically on PID strategies. These objectives were previously filtered by the *constraints handling* module that is activated for protecting personnel and equipment and for tackling large deviations from the target conditions.

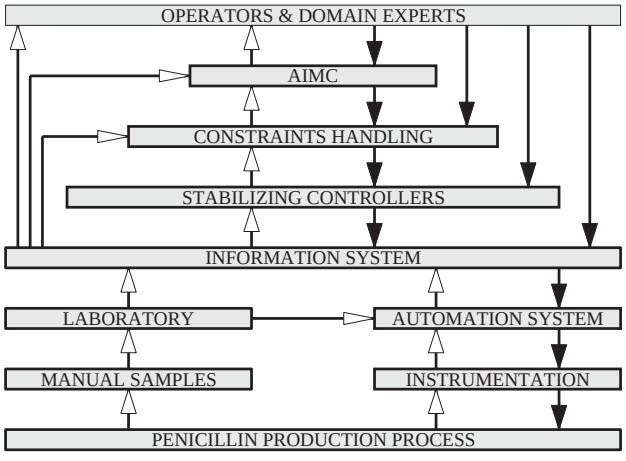

Fig. 1. Overall scheme of the hierarchical control structure for the penicillin production process. Bold arrows are control fluxes while hollow are information fluxes.

3. FasArt and FasBack neuro-fuzzy architectures

As mentioned above, the penicillin production process features high non-linearity and time varying parameters, and thus mathematical models are difficult to derive. In fact, none of the existing models for the penicillin production process describes it accurately. A good approximation for tackling such problems is offered by FasArt and FasBack (Cano-Izquierdo et al., 1996) hybrid systems that combine the adaptability of the Adaptive Resonance Theory (ART) (Grossberg, 1976) family of neural networks, with the capability of fuzzy sets theory (Zadeh, 1965) to express knowledge with rules. Based on Fuzzy ARTMAP (Carpenter, Grossberg, Markuzon, & Reynolds, 1992) architecture (and inheriting its structure, shown in Fig. 2b), FasArt and FasBack were proposed to overcome several ambiguities observed in Fuzzy ARTMAP, introducing fuzzy logic in a formal way, so that learning is equivalent to generating a set of fuzzy rules, and prediction consists of the use of a fuzzy inference engine with such rules. As an emergent result, they are more robust in the presence of noise in the training data (Cano-Izquierdo et al., 2001).

As fuzzy systems, FasArt and FasBack have a number of fuzzy rules with fuzzy antecedents (IF part), stored in the ART^a module, and fuzzy consequents (THEN part), stored in the ART^b module. The inter-ART module links them in a many-to-one mapping, i.e. different antecedents may have the same consequent. Examples of such fuzzy rules are shown in Fig. 12. A rule is true if *all* its antecedent components belong to their respective fuzzy set, as shown for input vector $(F(t), CPR(t), CPR(t-1), CPR(t-2))$ in the figure, where F is the feed and CPR is the carbon dioxide production rate. However, a fuzzy rule is not *either* true *or* not true; rather, it is true to some degree. For rules in Fig. 12, this degree is calculated using the product law, $\eta_{F(t)} \cdot \eta_{CPR(t)} \cdot \eta_{CPR(t-1)} \cdot \eta_{CPR(t-2)}$, where η represents a fuzzy membership function to a fuzzy set associated to the neuron and computed from its weights, as shown pictorially in Fig. 2a. It is noteworthy that several rules can be true to some extent at the same time, and therefore participate in producing the output, which is computed as a weighted average of the THEN part of all active rules.

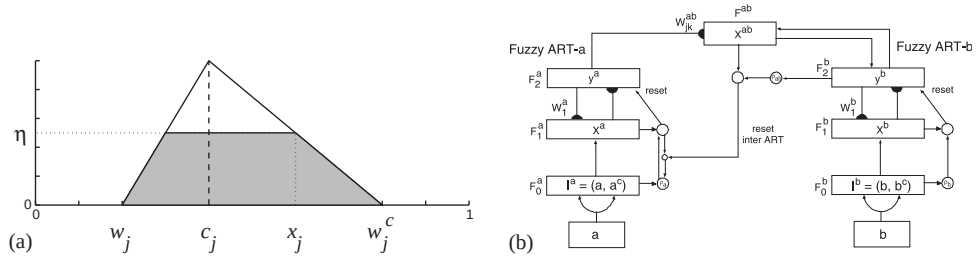

Fig. 2. Elements of the neuro-fuzzy systems FasArt and FasBack: (a) One dimensional fuzzy membership function for fuzzy set j, and the associated weights, w_j, c_j and w_j^c. For an input pattern x, the membership degree to the fuzzy set is given by η. (b) Network architecture, consisting of a Fuzzy ART^a module, where the antecedents of the fuzzy rules are stored, a Fuzzy ART^b module that holds the consequents, and the inter-ART module, that relates them.

As neural networks, FasArt and FasBack can be seen as two modules (ART^a and ART^b) that cluster samples in the input and output spaces, and an inter-ART module that map the input clusters into the output clusters (or equivalently, that relates the IF part of the rules to the THEN part). These modules have some user-tuned parameters that allow for control of how learning takes place: ρ_A, ρ_B (vigilance parameters) determine how fine (i.e. how specific) the clustering of antecedents and consequents should be; γ_A, γ_B (fuzzification rates) indicate how fuzzy or crisp antecedents and consequents clusters are; finally $\beta_A, \beta_B, \beta_A^C, \beta_B^C$ (learning rates) control the degree in which the new knowledge replaces the old one.

During the *learning stage*, these modules build the fuzzy rules as follows. Given a training input vector $\mathbf{a}$ and its associated output $\mathbf{b}$, they must be normalized into $[0,1]^M$ and their *complementary code* representation is calculated as $\mathbf{I}^a = (\mathbf{a}, \mathbf{a}^c) = (\mathbf{a}, \mathbf{1} - \mathbf{a})$ and $\mathbf{I}^b = (\mathbf{b}, \mathbf{b}^c) = (\mathbf{b}, \mathbf{1} - \mathbf{b})$.

Vector $\mathbf{I}^a$ is compared to existing *fuzzy templates*, stored in ART^a. Since the template j is stored in network weights $\mathbf{w}_j$ and $\mathbf{c}_j$ (see Fig. 2a). The similarity of the pattern to template j (i.e. the membership degree of the pattern to fuzzy set j) is given by

$$\eta_{R_j}(\mathbf{I}^a) = \prod_{i=1}^{M} \eta_{ji}(I_i), \tag{2}$$

where

$$\eta_{ji}(I_i) = \begin{cases} \max\left(0, \dfrac{\gamma_a(I_i^a - w_{ji}^a + 1)}{\gamma_a(c_{ji}^a - w_{ji}^a + 1)}\right) & \text{if } I_i^a \leqslant c_{ji}^a, \\ \max\left(0, \dfrac{\gamma_a(1 - I_i^a - w_{ji}^a + 1)}{\gamma_a(1 - c_{ji}^a - w_{ji}^a + 1)}\right) & \text{if } I_i^a > c_{ji}^a. \end{cases} \tag{3}$$

Then a *winner node* J is selected, to be that most similar to the pattern, i.e.

$$J = \arg\max_j \{\eta_{R_j}\}. \tag{4}$$

However, even if this template is the most similar, it may be because it is too general. The template is adequate if

$$\frac{|\mathbf{I}^a \wedge \mathbf{w}_J^a|}{|\mathbf{I}^a|} \geqslant \rho_a. \tag{5}$$

In this case *resonance* is said to occur. Otherwise, unit J inhibited for the rest of this pattern presentation, and a new winner is searched for, or a new one is committed (a new template is created) if none of the existing units meet Eq. (5).

Similarly procedures are carried out in ART^b, to determine the template that best matches pattern $\mathbf{I}^b$, say node K.

In that state, the network is predicting that template J in the input space maps into template K in the output space. If that is correct, learning proceeds in ART^a and ART^b according to Eqs. (7)–(10). If, however, the network had previously learned that template J maps into an ART^b template other than K, *inter-ART reset* occurs, causing the inhibition of unit J in ART^a. In addition, ρ_a is raised temporarily by

$$\rho_a = \frac{|\mathbf{I}^a \wedge \mathbf{w}_J^a|}{|\mathbf{I}^a|} \tag{6}$$

so that a new winner J is selected in ART^a, that is more specific than the inhibited one.

When learning takes place, templates J in ART^a and K in ART^b are updated to reflect the influence of the input vectors, by modifying weights $\mathbf{w}$ and $\mathbf{c}$ as follows:

$$\mathbf{w}_J^{a(new)} = \beta_a(\mathbf{I}^a \wedge \mathbf{w}_J^{a(old)}) + (1 - \beta_a)\mathbf{w}_J^{a(old)}, \tag{7}$$

$$\mathbf{w}_K^{b(new)} = \beta_b(\mathbf{I}^b \wedge \mathbf{w}_K^{b(old)}) + (1 - \beta_b)\mathbf{w}_K^{b(old)}, \tag{8}$$

$$\mathbf{c}_J^{a(new)} = \beta_a^c \mathbf{I}^a + (1 - \beta_a^c)\mathbf{c}_J^{a(old)}, \tag{9}$$

$$\mathbf{c}_K^{b(new)} = \beta_b^c \mathbf{I}^b + (1 - \beta_b^c)\mathbf{c}_K^{b(old)}. \tag{10}$$

If some unit in module is newly committed, *fast learning* is performed (i.e. $\beta = \beta^c = 1$).

Besides, the inter-ART map is modified so that $w_{JK}^{ab} = 1$, and all other $w_{Jk}^{ab} = 0$, to learn that template J in ART^a predicts template K in ART^b.

Through this process, FasArt and FasBack decide both the number of fuzzy rules, and the shape of their

antecedents, using only the available training data, and no a priori knowledge.

During the *prediction stage*, for every input pattern **a** FasArt and FasBack work as fuzzy logic systems predicting the output. They calculate the normalization of vector **a**, complementary code representation $\mathbf{I}^a$, and membership of this pattern to each of the ART^a templates using Eq. (2). Each of these ART^a templates predict one ART^b template, but the predicted output is obtained by defuzzification using the average of the fuzzy centers (see Wang, 1994):

$$\mathbf{y}(\mathbf{I}^a) = \frac{\sum_{k=1}^{N^b}\sum_{j=1}^{N^a}\eta_{R_j}^a(\mathbf{I}^a)w_{jk}^{ab}\mathbf{c}_k^b}{\sum_{k=1}^{N^b}\sum_{j=1}^{N^a}\eta_{R_j}^a(\mathbf{I}^a)w_{jk}^{ab}}, \tag{11}$$

where N^a and N^b are the number of templates in ART^a and ART^b, respectively, and $\mathbf{c}_k^b$ is the point where $\eta_{R_k}^b$ is maximum.

FasBack is a modification of FasArt, that uses the backpropagation algorithm (Rumelhart & McClelland, 1986) to refine learning in order to reduce global error, by locally re-learning the wrong input/output relations. This is carried out by using the descending gradient method, varying parameters (weights) in the direction indicated by the derivative of error with respect to the parameters vector (Wang, 1994). Furthermore, a penalty method is used to reduce the influence of wrong rules, although these rules are not completely forgotten and can be recalled if they become valid again. This penalty method is implemented letting weight w_{jk}^{ab} be smaller than 1 (in FasArt, $w_{jk}^{ab}=1$ means that template j in ART^a predicts template k in ART^b).

Therefore, applying the descending gradient method, new learning rules can be deduced for weights c_J^a, c_K^b and w_{jk}^{ab} (Cano-Izquierdo et al., 2001), as follows:

$$c_{in}^{a(new)} = c_{in}^{a(old)} - \varepsilon(y_m - d_m)\frac{\eta_{R_j}(\mathbf{I})}{\eta_{in}(I_n)}\frac{\partial\eta_{in}(I_n)}{\partial c_{in}^a} \times \frac{\sum_{l=1}^{N^b}w_{il}^{ab}(y_m - c_{lm})}{\sum_{l}^{N^b}\sum_{k=1}^{N^a}w_{kl}^{ab}\eta_{R_k}(\mathbf{I})}, \tag{12}$$

$$c_{jm}^{b(new)} = c_{jm}^{b(old)} - \varepsilon(y_m - d_m)\frac{\sum_{k=1}^{N^a}w_{kj}^{ab}\eta_{R_k}(\mathbf{I})}{\sum_{l}^{N^b}\sum_{k=1}^{N^a}w_{kl}^{ab}\eta_{R_k}(\mathbf{I})}, \tag{13}$$

$$w_{ij}^{ab(new)} = w_{ij}^{ab(old)} - \varepsilon(y_m - d_m)\frac{(y_m - c_{jm}^b)\eta_{R_i}(\mathbf{I})}{\sum_{l}^{N^b}\sum_{k=1}^{N^a}w_{kl}^{ab}\eta_{R_k}(\mathbf{I})}, \tag{14}$$

where Eqs. (12) and (14) are repeated for each element m of the error vector.

Due to the fact that this refining is local, and old rules are not completely forgotten, adaptation is stable and can be carried out on-line, without the need for storing previously learned patterns, as it would be the case with multilayer perceptrons using the backpropagation learning algorithm. With parameters ρ and γ it is possible to balance the compromise between complexity and accuracy. However, because for the same ρ and γ values FasBack algorithm will improve FasArt accuracy (Cano-Izquierdo et al., 2001), these parameters can be more relaxed in FasBack. Therefore, it will achieve similar accuracy, but with less complexity.

A more detailed discussion on FasArt and FasBack algorithm and properties can be found in Cano-Izquierdo et al. (2001).

4. Soft sensor implementation with FasArt and FasBack

On-line measurements of the main process variables are rare in biotechnological processes due to the lack of accurate, cheap on-line sensors that are robust in industrial conditions. Therefore, software sensors become important tools for supervising the fermentation. Some FasArt models were developed to monitor important variables, such as biomass, penicillin production and viscosity. FasArt and FasBack, as other neuro-fuzzy systems, are capable of learning fuzzy rules from examples. They are suitable as soft sensors to estimate unmeasured variables and also to provide some explanation of how this estimation is done.

To build soft sensors is necessary to determine which variables are informative to predict another variable. The variables were chosen after successive estimations using methods inspired in multivariable regression (Farlow, 1984). Then, data must be collected including values of the input variables and that of the variables to be predicted, and a training set is generated. It must be noted that though the training process can be carried out off-line, all input variables should be measurable on-line.

Once soft sensors are trained, they can be used to estimate the values of variables that are not easily measured from the values of those that can be measured on-line. Moreover, due to the incremental learning capability of FasArt and FasBack, the sensors can improve their knowledge during the performance phase if actual values of the predicted variables are obtained with a delay.

5. IMC implementation with FasArt and FasBack

The Internal Model Control concept refers to a group of methods that have been proposed since the 1970s with the objective of developing control strategies that combine the easy interpretation featured by classical controllers such as PID, with capabilities to deal with parameter uncertainty, noise in the signals and process constraints. For these reasons, these methods have been

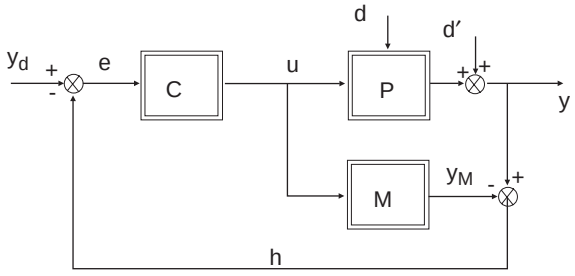

Fig. 3. Non-linear IMC structure. The blocks corresponding with the plant P, model M and controller C are represented in double line to denote the non-linear character of such modules.

widely implemented in the chemical industry (Braatz, 1996).

The basic IMC structure can be adapted to different control problems: control of SISO systems, non-linear control or multivariate control (Economou et al., 1986; Garcia & Morari, 1985). Fig. 3 shows the IMC structure for the control of non-linear systems, in which a model of the plant M is placed in parallel with the actual plant P, so that if the model is perfect, the feedback h is equal to the plant perturbation d. In such a case the system behaves as it in an open loop, thus overcoming the instabilities associated with the feedback (Morari & Zafiriou, 1998).

Though any feedback controller can be structured as an IMC, and conversely an IMC can be transformed into feedback form, the design of the controller associated with an IMC is easier than that associated to a feedback structure. This is due to the fact that IMC structure allows explicit inclusion of robustness as a design objective, with the use of the estimated perturbation as feedback signal. This allows IMC to have dual stability, perfect control and zero offset properties (Garcia & Morari, 1982). Furthermore, it has been proved that these IMC properties can be extended to controllers for non-linear plants (Economou et al., 1986). The IMC stability property requires conditions for the input–output stability of the plant and the controller and the availability of a plant model that is accurate enough. In the penicillin fermentation process the input–output stability of the plant is obviously guaranteed, since the quantity of biomass and products that can be yielded are limited by the feeding. The perfect control and zero offset properties are based on the use of controllers that approximate plant inverse dynamics.

Despite these advantages, the lack of either the plant model or the inverse model is a serious drawback, and often this is the case for bioprocesses since valid analytical models are not available, or are not accurate enough. Moreover, even if a plant model M was available, inverting it to build the control module C is not always possible due to the fact that either the inverse may not exist, or its implementation may not be feasible. Reasons for this could be that M is a non-minimum phase model, has time delays, or even that using its inverse would demand high gain loops.

Neuro-fuzzy methods provide a solution to build model and control modules by learning direct and inverse dynamics, and are well suited for non-linear plant identification (Lee, Jeon, Park, & Chang, 2002; Vlassides, Ferrier, & Block, 2001). Among them, FasArt and FasBack neuro-fuzzy systems (Cano-Izquierdo et al., 2001) feature fast stable learning guided by matching and error minimization, fuzzy representation of the knowledge, which allows the inclusion of expert rules, and good MIMO identification performance. Thus they are appropriate for building IMC strategies. These features allow the controller and the model to adapt to plant variations, permitting the design of an adaptive IMC (AIMC), which is of great interest in the control of a penicillin plant, in which not only parameters vary with time due to production degradation or strain mutations, but production results also vary from fermentation to fermentation even under the same conditions. Moreover, the possibility of interpreting the rules corresponding to C and M modules within the IMC framework allows experts to have an overall understanding of the system behavior at all times. Thus, if the performance is incorrect, it is possible to carry out an independent analysis of the model and the controller to find out the causes of the problem. In addition, FasArt and FasBack limit their output range to that learned during the training, hence ensuring input–output stability of the control module C.

One issue that arises when building IMC strategies with neuro-fuzzy systems is how to build modules M and C. Deriving M consists of training a network to replicate plant dynamics. However, FasArt and FasBack offer two different ways to obtain the control module C.

Inverting the fuzzy rules associated with M yields another set of rules that represent the inverse dynamics, as shown by (Gómez-Sánchez, Cano-Izquierdo, Araúzo-Bravo, Dimitriadis, & López-Coronado, 1998), thus being a fuzzy module C. The building of such a control module requires one model learning (direct) and the availability of an inversion method. On the other hand, a neuro-fuzzy system could be applied to learn inverse dynamics. In this case, module C can be built with a single model learning (inverse), taking as input signals the outputs of the plant, and as supervision signals the inputs to the plant. While the former approach has the advantage of reducing in some cases the influence of noise (Karniel, Meir, & Inbar, 2001), it is more complex, since some rule inverter is required. Here, for simplicity, the second approach was adopted.

Finally, if adaptation is desired, FasArt/FasBack modules can be determined in two stages. The first uses historical data from the plant to endow model and control modules with initial knowledge, using the scheme shown in Fig. 4 (Narendra & Parthasarathyu, 1990). A second phase takes place during normal performance, where fresh data is used to carry out on-line adaptive learning that enhances the knowledge of the existing models, as shown in Fig. 5. This is possible due to specific features of neural networks based on ART, that allow the learning of new data samples without catastrophic forgetting of previous knowledge (Grossberg, 1980). The adaptation of the plant model is carried out by learning an input/output pair every time a plant output value becomes available. The adaptation of the controller, on the contrary, requires knowing how new data samples alter inverse dynamics. Therefore, for the controller, an adaptation law as proposed by Hunt and Sbarbaro (1991) follows:

$$\mathbf{p}(k+1) = \mathbf{p}(k) + \alpha \cdot \mathbf{e} \cdot \mathbf{J}, \tag{15}$$

where $\mathbf{p}$ is the parameter (weights) vector, $\mathbf{e}$ is the tracking error, $\mathbf{J} = \partial y_m / \partial u_n$ is the Jacobian matrix of the plant, calculated numerically using the model, and α is an adaptation rate.

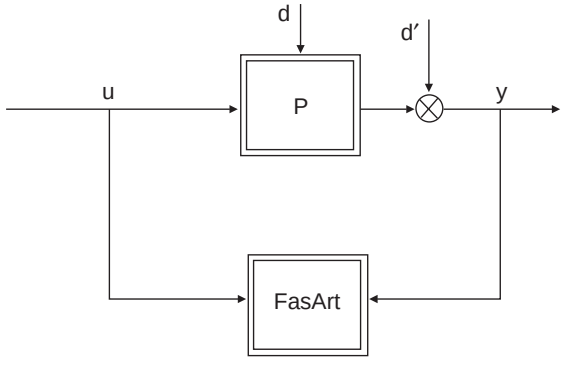

Fig. 4. Direct learning scheme for the off-line learning phase of FasArt and FasBack.

6. Experimental results

The application of the aforementioned ideas was carried out in three steps. Initially, for fast prototyping reasons, the controllers were trained and tested (with data sets different from the training sets) on a simulated plant. Subsequently, for security reasons, they were trained on real data and tested on the simulated plant. Finally, they were validated on the real plant. This section will show results achieved in the first and third steps and the results achieved in the second step are shown elsewhere (Araúzo-Bravo et al., 1999a). In addition, models for variables other than those to be controlled were derived, becoming soft sensors useful for monitoring the real plant, as also reported in this section. Due to confidentiality reasons, the values of some parameters and plots are scaled in the range [0,1], in real experiments.

6.1. Application to a simulated penicillin plant

In order to make fast prototypes and test the controllers in a safe way, a penicillin simulator was implemented, with a two-fold purpose: on one hand, it generated data to train FasBack modules; on the other, it replaces the actual plant within the AIMC structure during the test. Among the several mathematical models of penicillin fermentation revised (Nielsen & Villadsen, 1992; Nestaas & Wang, 1983; Bajpai & Reuß, 1981; Heijnen et al., 1979), Tiller's et al. (1994) model was selected, since it offers a good approximation to some real cases though it is quite simple. Its mathematical description is shown in the appendix. It also helped to check the hypothesis proposed by Mou (1975), stating that finding a good control for biomass facilitates the

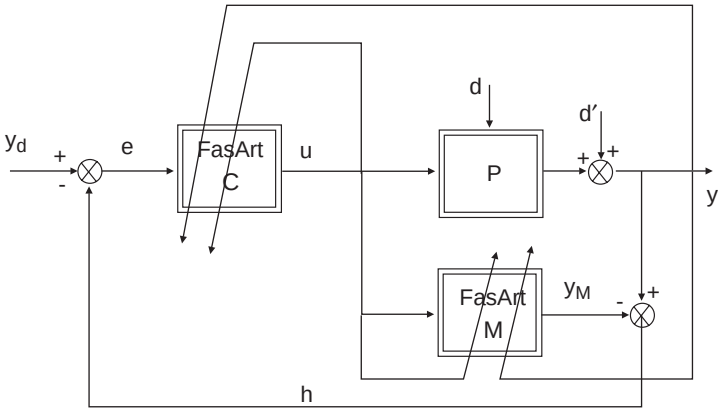

Fig. 5. On-line learning scheme for FasArt and FasBack. Since ART systems use learning guided by matching, they need information that flows from inputs and outputs to the blocks that perform the learning. These blocks are marked in the figure with crossed arrows to make the adaptation explicit.

control of penicillin production, and furthermore, that the main control variable is the carbon source feeding F. Results obtained on the simulated plant, and also those shown in the next subsection for the real plant, lead us to conclude that this hypotheses is acceptable in most cases.

6.1.1. Monitoring

Building soft sensors for the simulated plant is useful as an initial test of the neuro-fuzzy systems capabilities to approximate the dynamics of the problem. Moreover, this study may also help the design of soft sensors for the real plant, with decisions such as the selection of input features or the ranges of variables.

For this evaluation soft sensors were derived for two important variables, biomass and penicillin production, while viscosity was not considered at this point since all the mathematical models reviewed (Tiller et al., 1994; Nielsen & Villadsen, 1992; Nestaas & Wang, 1983; Bajpai & Reuß, 1981; Heijnen et al., 1979) do not model the viscosity of the broth; modifying those mathematical models to reflect the effect of viscosity is out of the scope of our research. However, as shown later in Section 6.2.1, a soft sensor was derived using real data, that is in fact a good implicit model of viscosity that can overcome the lack of mathematical models of this variable.

To derive soft sensors for biomass and penicillin production, several nominal batches were simulated using random feeding laws constrained within certain bounds, as shown in Fig. 6a, while other inputs were kept fixed. These data were used to train the neural models, and several other test unseen batches were used to evaluate performance (Gómez-Sánchez et al., 1998). Fig. 6b shows a typical result of biomass prediction, while average relative root mean square error (RRMSE) indices can be seen in Table 3 for the prediction made by FasBack, illustrating the high accuracy achieved.

6.1.2. Control

To validate the proposed control scheme, a PID controller was tested. An important advantage of PID control is that it does not require a plant model, since its parameters can be tuned by trial and error. When the reference for biomass is set according to Eq. (1), the controller offers poor tracking and generates a feeding law that is not feasible since it is highly oscillating. An increase in the derivative gain would lead to better tracking but rougher feeding law, even less feasible. Instead, decreasing the derivative gain allows finding a feasible feeding, but with a very slow tracking, thus strongly influencing penicillin production (Araúzo-Bravo et al., 1999a). These results are in agreement with (Boskovic & Narendra, 1995) who show (in the case of an alcoholic fermentation with *Saccaromyces cerevisiae*) that when the plant is complex (non-linear, with noise and time variant parameters), PID control is not satisfactory and more elaborated controllers should be used.

In order to build modules M and C for the AIMC, 30 fermentations were generated, using feeding laws that varied randomly but within bounds suggested by experience (one example of such law is shown in Fig. 6a), while other inputs were kept fixed in all the fermentations. To simulate more realistic conditions, Gaussian noise was added to mass measurements, with 0.1% amplitude (these values would be provided by a gas spectrometer in a real scenario) and 5% for laboratory analyses. Although a simulator can provide continuous measurements of any variable, the laboratory variables were down-sampled to realistic rates and then interpolated to have a closer approach to results

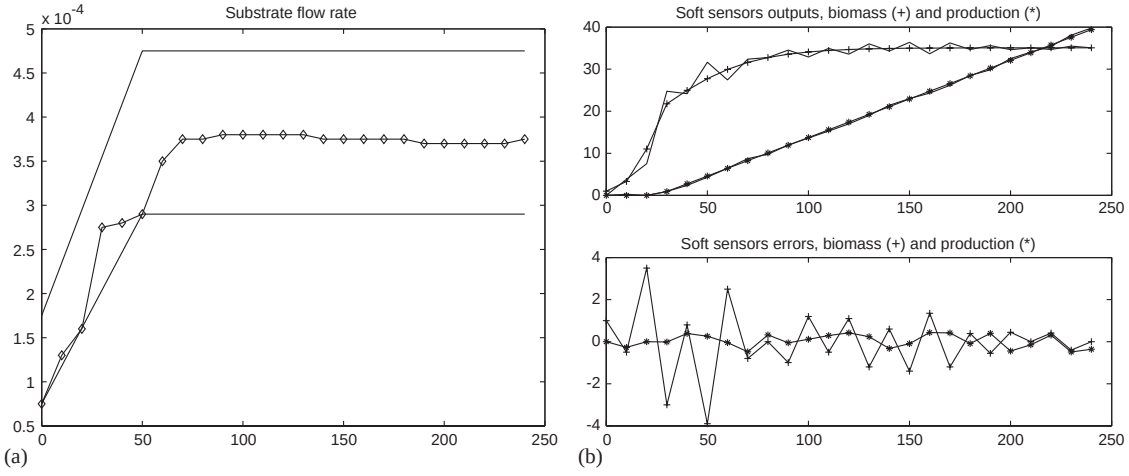

Fig. 6. Soft sensors of the nominal simulated plant. (a) Random feeding law used to simulate fermentations, and bounds for randomly generated feeding laws. (b) Simulation values (solid) and estimations of biomass (+) and production (*).

that will be obtained when using real data. Direct and inverse dynamics were learned using two FasBack neural networks with 10 training cycles (in each of 30 × 240 samples were presented). The inputs, outputs and nodes of each module are shown in Table 2. Note that viscosity is a very informative variable in reality, but its relation to biomass growth it not captured by the simulation model, and therefore it cannot be used in the simulation work.

Satisfactory direct and inverse dynamics approximation allow for the building of an AIMC using these two modules. The fact that the model reflects the general behavior of the plant, rather than accurately describing the actual plant, it penalizes control performance. This is however, overcome because of adaptation, achieving a good tracking of the reference, as shown in Fig. 7, and quantitatively in Table 3.

Moreover, the proposed controller performs well in harder conditions. Fig. 8 shows its capabilities to compensate noise in plant outputs. In addition, Fig. 9 shows the performance in an experiment that simulates the influence of cell damage by lysis and shear forces (Tiller et al., 1994). It was roughly assumed that these effects influence mainly biomass yield on sugar (Y_{XS} in the model by Tiller et al., 1994), supposing that it decreases from 0.5 g/g at time 100 h to 0.1 g/g at the end of the fermentation. Results in both adaptive cases show better tracking of reference than the respective non-adaptive controllers. Though this is a moderate improvement (as shown through RRMSE in Table 3), these controllers achieved similar penicillin production with smoother control laws than the respective non-adaptive controllers.

Table 2
Input vectors (**a** is the input to ART^a and **b** is the input to ART^b) and number of nodes of each neuro-fuzzy module of the plant model and control model of the AIMC structure for the case of the simulated penicillin plant

Plant identifier (direct dynamics)	Number of nodes
$\mathbf{a} = [F(t), CPR(t), CPR(t-1), CPR(t-2)]$	$N_a = 47$[a]
$\mathbf{b} = [X(t+1)]$	$N_b = 34$
Controller (inverse dynamics)	Number of nodes
$\mathbf{a} = [X(t+1), CPR(t), CPR(t-1), CPR(t-2)]$	$N_a = 50$
$\mathbf{b} = [F(t+1)]$	$N_b = 39$

[a] In the case of adaptation of the nominal model, the number of nodes in ART^a increases to 48.

6.2. Application to the real plant

In a second stage FasArt and FasBack were trained using real data to generate neuro-fuzzy modules to

Table 3
Relative root mean square error (RRMSE) for the soft sensors of viscosity *VIS*, penicillin production *P*, biomass *X* and control of the biomass X_c (this index is calculated comparing the actual biomass value to that of the biomass reference)

Experiment	*VIS*	*P*	*X*	X_c
Simulated plant				
Fig. 6b	—	0.014	0.048	—
Fig. 7a	—	0.015	0.052	0.061
Fig. 7b	—	0.014	0.049	0.052
Fig. 8a	—	0.017	0.056	0.054
Fig. 8b	—	0.017	0.054	0.053
Fig. 9a	—	0.016	0.069	0.072
Fig. 9b	—	0.015	0.062	0.061
Real plant				
Average	0.047	0.010	0.17	0.22

"Experiment" column refers to the figures that show these results, except for results reported to the real plant, that are the average of six batches at Antibióticos pilot plant.

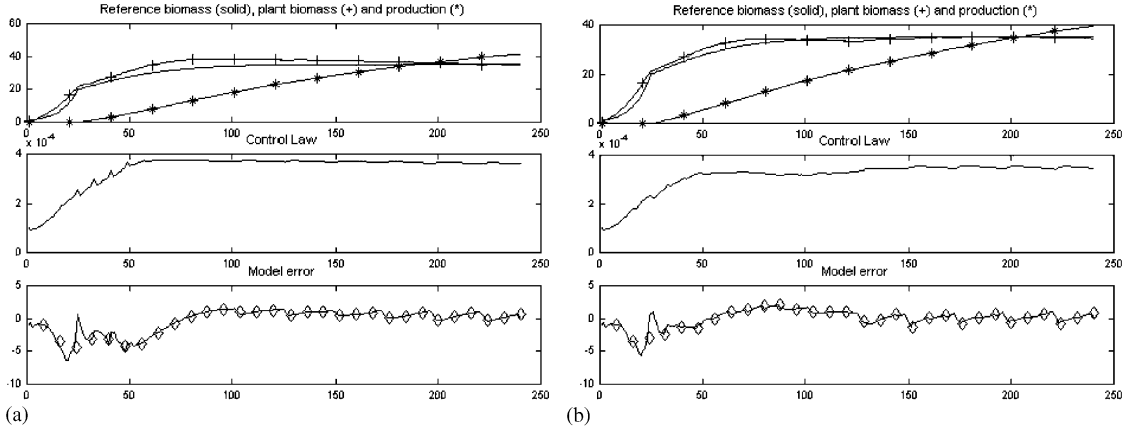

Fig. 7. Control of the nominal simulated plant. (a) Without adaptation (b) With adaptation of the model and the controller. In the upper part '+' signs show biomass measurement at 8 h sampling intervals, the biomass reference is in solid line and '*' signs show penicillin production. In the middle part the feeding control law is shown and in the bottom part the model error, where '◇' signs show error values that are fed back to the controller. In the case of adaptation, the number of nodes in ART^a increases to 48.

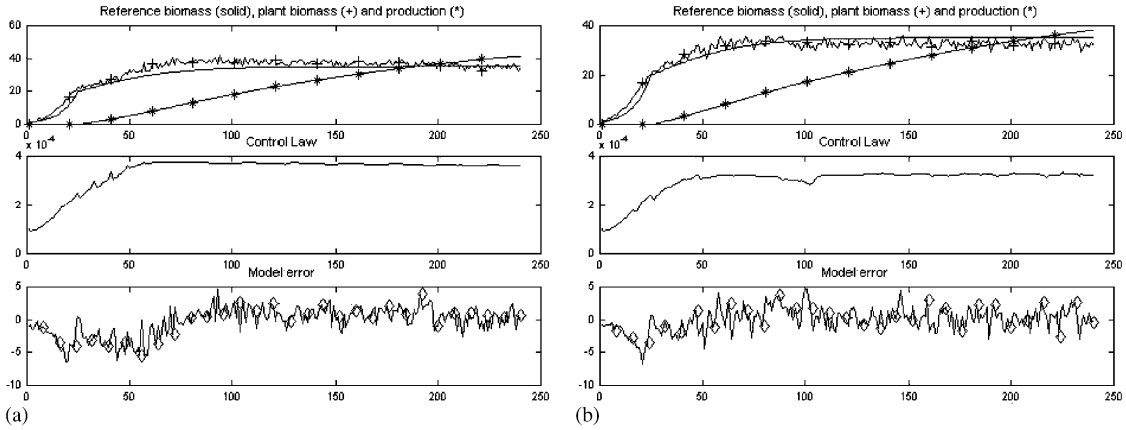

Fig. 8. Control of the simulated plant when there is 5% noise in the measurement of biomass and 0.1% noise in the measurement of *CPR*. (a) Without adaptation (b) with adaptation of the model and the controller. In the upper part '+' signs show biomass measurement at 8 h sampling intervals, the biomass reference is in solid line and '*' signs show penicillin production. In the middle part the feeding control law is shown and in the bottom part the model error, where '◇' signs show error values that are fed back to the controller.

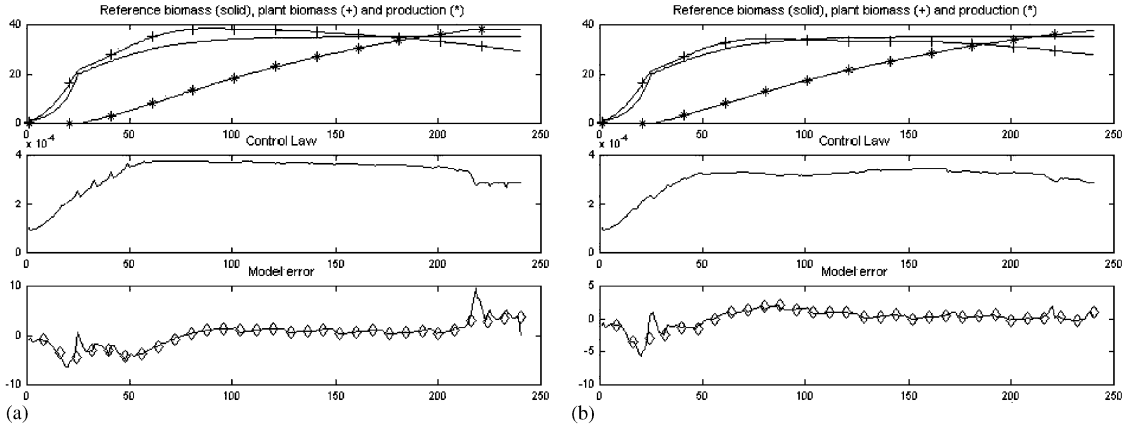

Fig. 9. Control of a time varying simulated plant, where Y_{XS} decreases from 0.5 g/g at time 100 h to 0.1 g/g at the end of the fermentation. (a) Without adaptation (b) With adaptation of the model and the controller. In the upper part '+' signs show biomass measurement at 8 h sampling intervals, the biomass reference is in solid line and '*' signs show penicillin production. In the middle part the feeding control law is shown and in the bottom part the model error, where '◇' signs show error values that are fed back to the controller.

monitor the process, as well as to control the pilot plant. Hence, the IMC was tested in Antibióticos Factory at León, Spain. Experts stated that it was not necessary to achieve very accurate biomass tracking, as long as maintaining the desired trend of the expected penicillin production, and process constraints were not violated. In this sense, based on SPC ideas (Lennox et al., 2001) a biomass reference was proposed that averages the biomass achieved in several successful fermentations. Training data consisted of a total of 28 fermentations, including standard (normal behavior under nominal conditions), and non-standard (normal behavior under non-nominal conditions). Anomalous fermentations (non-standard behavior under nominal or non-nominal conditions) were not used in order not to disturb the knowledge acquired in the fuzzy rules. Finally, the modules proposed in this paper were validated with six fermentations at the pilot plant.

6.2.1. *Monitoring*

Prior to the implementation of an IMC controller at Antibióticos plant, some FasArt models were developed to monitor important variables, such as biomass, penicillin production and viscosity. These models were trained on the same real data to infer the estimated variable from information of nutrients additions, agitation and past measurements of some outlet gases.

The biomass concentration is one of those key variables that cannot be measured on-line. Several methods have been developed by different researchers to reconstruct it from information available on-line. Galvanauskas, Simutis, & Lubbert (1998) conclude that the measurement of the number and size of cells is difficult to implement on-line, while a method based on acid/base consumption during pH control is cheap but not always applicable. On the other hand, the method based on laser turbidimeter signals is cheap and easy to

apply, but the overall preferred method is based on outlet gas measurements, and features the advantages of the previous methods but also provides additional culture information. The biomass software sensor reported here uses similar input information as in the latter method, in addition to past values of laboratory measurements of biomass as an additional input variable. Results in Fig. 10a show a good biomass prediction that in turn facilitates the achievement of a reasonable penicillin profile.

Another relevant variable in the penicillin production process is the viscosity of the culture. Its importance stems from the fact that, for an aerobic bioprocess such as this, it is necessary to maintain dissolved oxygen in the tank above a certain threshold. This can be achieved by an adequate oxygen transfer. Usually this capability is estimated by means of the oxygen transfer coefficient at the middle of the fermentor K_{L_a}. Though this value can be estimated from exhaust gas measurements obtained through paramagnetic and infrared analyzers (Spriet, Betterman, DeBuyser, Visscher, & Vandamme, 1982) or mass spectrometry (Buckland et al., 1985), these methods are not valid in the case of viscous mycelial fermentations (Buckland et al., 1985), as is the case in penicillin production. This is due to the fact that high viscosity considerably reduces such transfer capability and affects K_{L_a} estimation. However, viscosity itself can be used as a good oxygen transfer estimator. Furthermore, a nominal profile of viscosity can be a good indicator of correct process evolution, and thus it can be used for fault detection during the fermentation. Since viscosity measurements are difficult, expensive and involve a large delay, a viscosity soft sensor becomes very useful for the fermentation supervisor (Gómez et al., 1999).

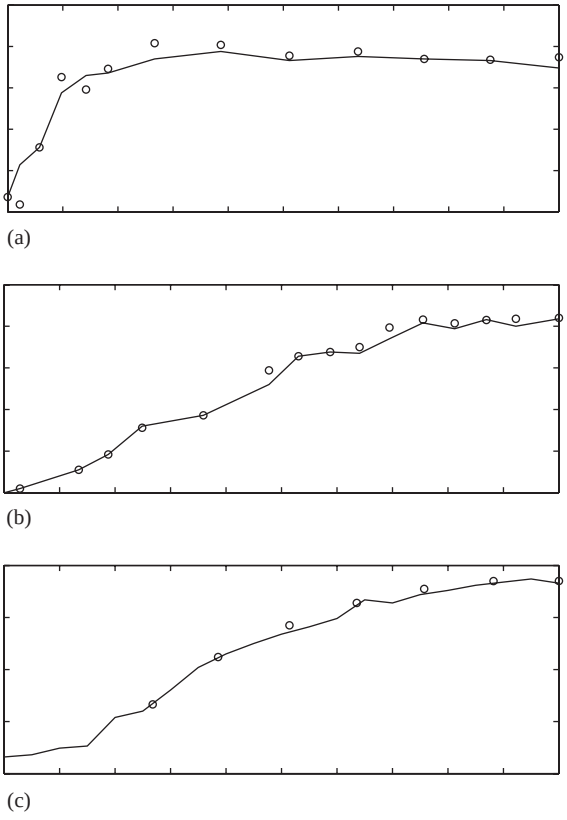

Fig. 10. Soft sensors performance on the real plant. The solid lines are predictions given by FasArt software sensors, while circles denote laboratory measurements of (a) biomass, (b) viscosity and (c) penicillin production.

A neuro-fuzzy software sensor of viscosity was developed by training FasArt on the same real data mentioned above. Again, feedings and gas measurements were used as inputs. In addition, the viscosity value obtained in the last laboratory measurement was used as an input to the predictor. Results for one unseen fermentation are shown in Fig. 10b, where it can be noticed that prediction is very accurate.

Finally, a software sensor for penicillin was developed. It should be understood that the final objective of the fermentation process is penicillin production, and therefore after identification and control, research should be aimed at process optimization. In this sense, measurements of penicillin should be taken frequently during fermentation. However, as in the case of viscosity and biomass, penicillin laboratory measurements have an important delay and cost. The results of the penicillin soft sensor for one unseen fermentation are shown in Fig. 10c, where it can be seen that prediction is very accurate. This illustrates how soft sensors developed here can be applied to obtain on-line accurate measurements replacing expensive and difficult off-line procedures.

6.2.2. Control

The IMC controller for the real pilot plant was developed following the same methodology as above. To train the neuro-fuzzy system corresponding to the M module, six inputs were used, including information of nutrients additions, agitation and past measurements of some outlet gases. Then, the C module was derived using desired biomass and past measurements of outlet gases as network inputs, while additions to the tank and agitation were the outputs of C (the manipulated variables). Since some of the variables correspond to plant outputs, they had been obtained through laboratory measurements and therefore were sampled at low rates. Thus they had to be linearly interpolated, but only for training purposes. This can be done because in the test stage (on-line control) their values are not necessary (they are computed by the neuro-fuzzy systems).

Fig. 11 shows the performance of the controller in one of the six fermentations carried out for validation in the pilot plant. Although tracking of biomass reference is not very accurate, the general trend is followed, as desired by Antibióticos experts. Moreover, considering

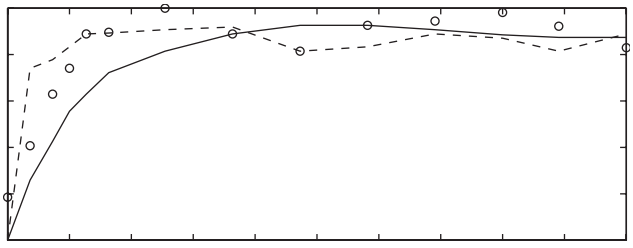

Fig. 11. AIMC performance on the real plant. The solid line is the reference biomass, the dashed line is that predicted by the plant model and circles denote laboratory measurements.

that laboratory measurements involve high uncertainty and the sampling rate is low, the laboratory profile itself is considered by Antibióticos experts only as a trend. Therefore, this result can be considered satisfactory since the achieved penicillin production was as high as that achieved in batches controlled by human experts. These results point out the feasibility of developing automatic controllers that may eventually replace human supervision with similar performance

Moreover, the plant model M produced an output inferred from gas measurements that were affected by noise, causing high frequency components in its prediction. However, the trend was consistent with measured values, showing that IMC is also robust to small perturbations in plant measurements.

7. Rules extraction and manipulation

The black box behavior of neural networks is considered a good feature since it allows to model system dynamics without explicit knowledge or a mathematical model of the system. However, it is also a drawback since it hides the way the controller takes decisions.

Though most neural network architectures have this black box behavior, FasArt and FasBack organize the acquired knowledge in the form of fuzzy rules that are easily interpretable, as illustrated in Fig. 12. The rules shown were generated by FasBack after having been trained to learn direct dynamics using data from the real plant. Rule r_3 was generated during the tropophase and shows how low feed $F(t)$ and fast growth of CPR (starting from low values, as indicated through $CPR(t)$, $CPR(t-1)$ and $CPR(t-2)$) yield low biomass $X(t+1)$, at the fermentation earlier stages. Rule r_{21} was generated during the idiophase, showing how high feed $F(t)$ and stable high CPR are associated to high biomass $X(t+1)$, in the stable stages. Finally, rule r_9 was generated during the transition from tropophase to idiophase, and relates average feed $F(t)$ and oscillating average CPR to an average value of biomass $X(t+1)$. The transition phase is the period in which the system exhibits richer dynamics, thus becoming more difficult to predict. This fact is reflected by a lower confidence for that rule, and also by higher number of rules generated during this stage.

From the few fuzzy rules shown in Fig. 12 for direct dynamics it is also easy to guess how rules describing inverse dynamics may be, by setting the fuzzy condition on feed F as a consequent, while using the fuzzy set related to biomass X as another condition in the IF part of the rule. This issue is specially interesting since it permits the introduction of actions to be performed in the case of violating constraints. To insert these restrictions in the fuzzy rule base, the input state in which the control action should trigger must be selected and expressed as a fuzzy set. Then, the control action itself should be set as a consequent of the rule. Finally, this rule can be easily translated into neuro-fuzzy weights.

As an example, consider the problem of coping with a significantly high viscosity (VIS), which can be reduced by setting a high agitation speed (RPM). The following linguistic rule describes the action proposed by the expert:

IF VIS IS "HighViscosity"
AND…THEN
RPM IS "HighRPM".

This rule must be converted into a fuzzy rule, for which the terms "HighVisc" and "HighRPM" must be mapped into fuzzy sets (in their respective domains). As a result, the following fuzzy rule was introduced in the controller:

IF VIS IS $\Delta\{600, 700, 800\}$ THEN
RPM IS $\Delta\{220, 220, 220\}$

where $\Delta\{min, cen, max\}$ denotes a triangular membership function that is 0 for values smaller than min and larger than max, and is maximum at point cen. Other rules inserted in the controller were

IF VIS IS $\Delta\{800, 900, 1000\}$
THEN RPM IS $\Delta\{220, 220, 220\}$
AND ΔF IS
$\Delta\{-20, -20, -20\}$

IF VIS IS $\Delta\{400, 700, 1000\}$
AND age IS $\Delta\{50, 75, 100\}$
THEN RPM IS
$\Delta\{220, 220, 220\}$ AND
ΔF IS $\Delta\{-20, -20, -20\}$

where ΔF means a change in the feed initially proposed by the controller. These rules can be easily transformed into neural network weights through

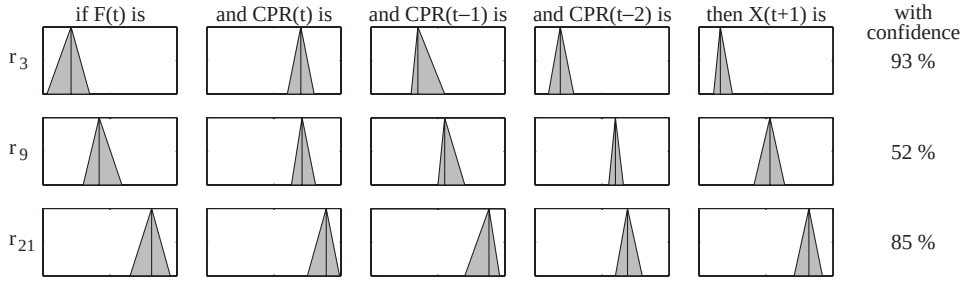

Fig. 12. Rules extracted from FasBack after training with the direct dynamics of the real plant.

normalization of points that determine the triangular membership functions.

Despite broad literature showing that KB approaches had already advanced enough to deal with many of the problems in fermentation processes that remained unsolved by conventional systems, KB approaches are still quite rare in the industry in regular operation (Shioya et al., 1999). Clearly, it would help if KB approaches were "transparent", thus allowing pigeonholing and interpretation of biological phenomena in connection with them. In this sense, this example illustrates how the models generated by FasArt and FasBack are transparent boxes rather than black boxes. Moreover, since FasArt and FasBack have the capability of stable learning (i.e. do not suffer from catastrophic forgetting), they allow modification of the rule database with heuristic knowledge from experts, allowing to combine or refine a priori knowledge with that extracted from the data.

8. Conclusions

The penicillin production process suffers from a lack of good mathematical models, because of its complexity, since it is highly non-linear, has time varying parameters. It also has many important variables that cannot be measured on-line, along with others that can present high levels of noise. While IMC structure features noise rejection, solving the latter problem, neuro-fuzzy systems FasArt and FasBack can be used to build plant model and controller from those variables that are easily measurable, thus allowing for the extending of IMC structure to non-linear plant control. In addition, FasBack on-line adaptive capability can be used to implement adaptive IMC strategies.

This scheme has been validated on a simulated plant, under realistic conditions, as well as in a real plant. The IMC based on these neuro-fuzzy models showed satisfactory reference tracking, even with noisy data, while generating feasible profiles for the manipulated variables, as opposed to a PID controller. Adaptive IMC strategies, that exploit FasBack on-line learning feature, were tested in the simulated plant, showing how adaptation can correct on the fly some of the unexpected variations occurring along the fermentation. Finally, in Antibióticos real pilot plant, the IMC controller maintained the general trend of biomass reference, thus guaranteeing profitable penicillin production, similar to those achieved in batches controlled by human experts. This result points out the feasibility of eventually automating the control of the process that is currently human-supervised.

In addition, soft sensors were developed to estimate on-line important variables that were traditionally measured in the laboratory, as biomass, viscosity and penicillin. The performance of these soft sensors is accurate and very helpful for the human supervision and understanding of the process, or for the implementation of fault prediction tools.

The tools proposed in this paper can work complementarily with the current human supervision of the process: the controller is effective in batches regarded as standard, while soft sensors can help to detect anomalous situations and provide useful information to the human supervisors to aid their decisions.

Finally, it is noteworthy that FasArt and FasBack present the possibility of inclusion, manipulation and generation of linguistic rules, thus offering a means to gain insight of the process, as well as combining knowledge extracted data with existing expert knowledge. This feature can help the adoption of this type of automatic monitoring and control strategies in real production plants.

Acknowledgements

This work was partially supported by ESPRIT project no. 22416 "MONNET" of the European Union and by CICYT project no. 1DPI 2000 04150 P403 of the Spanish government. Authors thank the personnel of Antibióticos, S.A.U. at León, Spain, especially to Estrella Méndez Pérez and Juan Pablo Zumel Arranz for their help in the on-site implementation of the monitoring system and the controllers described in this paper.

The authors also thank the anonymous reviewers for useful suggestions that significantly helped to improve the readability of the paper.

Appendix A. Mathematical model of penicillin production

The Tiller model (Tiller et al., 1994), is a segregated model that distinguishes two types of microorganism populations. One is the population X_1, that grows and produces penicillin, and another is X_2 that produces penicillin without growing. The dynamics of such populations are given by the following equations:

$$X = X_1 + X_2, \tag{A.1}$$

$$\frac{dX_1}{dt} = (\mu_S + \mu_{PM})X_1 - (D + k_{12})X_2, \tag{A.2}$$

$$\frac{dX_2}{dt} = k_{12}X_1 - (D + k_{ly})X_2, \tag{A.3}$$

where D is the dilution rate, and the parameters of population change k_{12} and lysis k_{ly} depend on the reaction average age A according to the equations:

$$k_{ly} = a_{ly} + b_{ly}A, \tag{A.4}$$

$$k_{12} = f_{12}A, \tag{A.5}$$

$$A(t) = \frac{1}{X(t)}\int_0^t X(\tau)\,d\tau. \tag{A.6}$$

The kinetic equations associated to the biomass follow a Monod law:

$$\mu_S = \mu_{Smax}\frac{S}{K_S + S}, \tag{A.7}$$

$$\mu_{PM} = \mu_{PMmax}\frac{PM}{K_{PM} + PM}, \tag{A.8}$$

$$\mu = (\mu_{Smax} + \mu_{PM})\frac{X_1}{X_1 + X_2}, \tag{A.9}$$

where S is the main substrate associated with the carbon source, and PM is the pharmamedia, where all the substrates are lumped. The evolution of such substrates is given by

$$\frac{dS}{dt} = -\frac{\mu_S}{Y_{XS}}X_1 - \left(\frac{\pi}{Y_{PS}} + m\right)X + F - DS, \tag{A.10}$$

$$\frac{dPM}{dt} = -\frac{\mu_{PM}}{Y_{XPM}}X_1 + k_{ly}X_2 - DPM. \tag{A.11}$$

The penicillin production is given by

$$\frac{dP}{dt} = \pi X - (D + k)P, \tag{A.12}$$

where the production kinetic law π is represented by a trapezoidal law, which can be implemented with four fuzzy sets as is shown in Fig. 13. Finally, the evolution of the main gases involved in the fermentation are given by the equations:

$$OUR = \frac{\mu_S + \mu_{PM}}{Y_{XO}}X_1 + \left(\frac{\pi}{Y_{PO} + m_O}\right)X, \tag{A.13}$$

$$CPR = \frac{\mu_S + \mu_{PM}}{Y_{XC}}X_1 + \left(\frac{\pi}{Y_{PC} + m_C}\right)X, \tag{A.14}$$

where OUR is the oxygen uptake rate and CPR is the carbon dioxide production rate. All model parameters are given in Table 4.

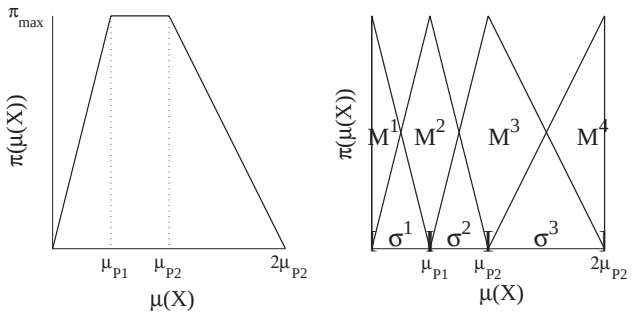

Fig. 13. Production specific rate (left) $\pi(\mu(X))$ as a function of the growth specific rate $\mu(X)$, as stated by Tiller et al. (1994). This profile can be interpreted as a fuzzy system with four triangular rules (right) (M^1, M^2, M^3, M^4), where the consequents are singletons with values $y^1 = 0$, $y^2 = \pi_{max}$, $y^3 = \pi_{max}$, $y^4 = 0$, and π_{max} is the maximum production specific rate.

Table 4
Parameters for the simulation of the fermentor, taken from the first cultive parameters in Tiller et al. (1994)

Parameter	Units	Value	Parameter	Units	Value
μ_{Smax}	h^{-1}	0.06	Y_{PC}	g/g	0.2
π_{max}	h^{-1}	0.0046	k	h^{-1}	0.0006
K_S	g/l	0.07	μ_{P1}	h^{-1}	0.003
μ_{PMmax}	h	0.03	μ_{P2}	h^{-1}	0.014
K_{PM}	g/l	2.0	f_{12}	h^{-2}	0.00046
Y_{XS}	g/g	0.47	a_m	h^{-2}	0.00015
Y_{XPM}	g/g	0.51	b_m	h^{-1}	0.001
Y_{PS}	g/g	1.2	m_O	$g\,l^{-1}\,h^{-1}$	0.02975
Y_{XO}	g/g	1.25	m_C	$g\,l^{-1}\,h^{-1}$	0.0221
Y_{PO}	g/g	6.25	a_{ly}	h^{-1}	−0.0008
Y_{XC}	g/g	0.9	b_{ly}	h^{-1}	3×10^{-6}

The parameters μ are associated to specific growing rates, π to production, K are kinetic constants, Y yields, m maintenance coefficients and a, b and f are regulation coefficients depending on the age.

References

Agrawal, R., Deepika, N. U. A., & Joseph, R. (1999). Strain improvement of *Aspergillus* Sp. and *Penicillium* Sp. by induced mutation for biotransformation of α-Pinene to Verbenol. *Biotechnology and Bioengineering*, *63*(2), 249–253.

Araúzo-Bravo, M. J., Gómez-Sánchez, E., Cano-Izquierdo, J. M., López-Coronado, J., López-Nieto, M. J., & Collados-Vieja, A. (1999a). Control of the penicillin production with adaptive IMC using fuzzy neural networks. In *Proceedings of the 14th world congress of IFAC international federation of automatic control*, vol. O, Beijing, China (pp. 499–504).

Araúzo-Bravo, M. J., Gómez-Sanchéz, E., Dimitriadis, Y. A., Cano-Izquierdo, J. M., & López-Coronado, J. (1999b). FLAS: A Fuzzy Linear Adaptive System for identification of non-linear noisy functions. In *Proceedings of the IEEE international conference on systems, man and cybernetics*, vol. 3, Tokyo, Japan (pp. 10–16).

Azimzadeh, F., Galán, O., & Romagnoli, J. A. (2001). On-line optimal trajectory control for fermetation process using multi-linear models. *Computers & Chemical Engineering*, *25*, 15–26.

Bajpai, R. K., & Reuß, M. (1981). Evaluation of feeding strategies in carbon-regulated secondary metabolite production through mathematical modelling. *Biotechnology and Bioengineering*, *23*, 717–738.

Birol, G., Undey, C., Parulekar, S. J., & Cinar, A. (2002). A morphologically structured model for penicillin production. *Biotechnology and Bioengineering*, *77*(5), 538–552.

Boskovic, J. D., & Narendra, K. S. (1995). Comparison of linear, nonlinear and neural-network-based adaptive controllers for a class of fed-batch fermentation processes. *Automatica*, *31*(6), 817–840.

Braatz, R. (1996). *Control handbook*. Boca Raton, FL: CRC Press.

Buckland, B., Brix, T., Faster, H., Gbewinyo, K., Hunt, G., & Jain, D. (1985). Fermentation exhaust gas analysis using mass spectrometry. *Bio/Technology*, *3*, 982–988.

Camacho, E. F., & Bordons, C. (1994). *Model Predictive Control in the Process Industry*. London: Springer.

Cano-Izquierdo, J. M., Dimitriadis, Y. A., Araúzo-Bravo, M. J., Abajo-Manzano, F., & López-Coronado, J. (1996). Fuzzy adaptive system ART-based: Theory and application to identification of biochemical systems. In *Proceedings of CESA-96 IMACS multi-conference computational engineering in systems applications*, vol. 2, Lille, France (pp. 918–923).

Cano-Izquierdo, J. M., Dimitriadis, Y. A., Gómez-Sánchez, E., & López-Coronado, J. (2001). Learning from noisy information in FasArt and FasBack neuro-fuzzy systems. *Neural Networks*, *14*(4–5), 407–425.

Carpenter, G. A., Grossberg, S., Markuzon, N., & Reynolds, J. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, *3*, 698–713.

Carpenter, G. A., & Tan, H. A. (1995). Rule extraction: From neural architecture to symbolic representation. *Connection Science*, *7*(1), 3–27.

Cunha, C. C. F., Glassey, J., Montague, G. A., Alabert, S., & Mohan, P. (2002). An assessment of seed quality and its influence on productivity estimation in an industrial antibiotic fermentation. *Biotechnology and Bioengineering*, *78*(6), 658–669.

Economou, C. G., Morari, M., & Piasson, B. O. (1986). Internal model control. 5. Extension to nonlinear systems. *Industrial Engineering Chemical Process Design and Development*, *25*, 404–411.

Farlow, S. J. (Ed.). (1984). *Self-organizing methods in modeling, GMDH type algorithms*. New York: Marcel Dekker, Inc.

Galvanauskas, V., Simutis, R., & Lubbert, A. (1998). Direct comparison of four different biomass estimation against conventional dry weight measurements. *Process Control and Quality*, *11*(2), 119–124.

Garcia, C. E., & Morari, M. (1982). Internal model control. 1. A unifying review and some new results. *Industrial Engineering Chemical Process Design and Development*, *21*(2), 308–323.

Garcia, C. E., & Morari, M. (1985). Internal model control. 2. A design procedure for multivariable systems. *Industrial Engineering Chemical Process Design and Development*, *24*(2), 472–484.

Gómez, E., Araúzo-Bravo, M. J., Cano-Izquierdo, J. M., Dimitriadis, Y. A., López-Coronado, J., & López-Nieto, M. J. (1999). Control of the penicillin production using fuzzy neural networks. In *Proceedings of the IEEE international conference on systems, man and cybernetics*, vol. 6, Tokyo, Japan (pp. 446–450).

Gómez-Sánchez, E., Cano-Izquierdo, J. M., Araúzo-Bravo, M. J., Dimitriadis, Y. A., & López-Coronado, J. (1998). Adaptive IMC using fuzzy neural networks for the control of non linear systems. In N. Måartenson, R. Mackay, & S. Bjőrgvinsson (Eds.), *Changing the ways we work* (pp. 792–801). Amsterdam: IOS Press.

Grossberg, S. (1976). Adaptive pattern classification and universal recoding. I: Parallel development and coding of neural features detectors. *Biological Cybernetics*, *23*, 121–134.

Grossberg, S. (1980). How does a brain build a cognitive code? *Psychological Review*, *87*, 1–51.

Grossberg, S. (1982). *Studies of mind and brain: Neural principles of learning, perception, development, cognition and motor control*. Boston, MA, USA: Reidel Press.

Havlik, R. S. I., & Lubbert, A. (1992). A fuzzy-supported extended Kalman filter: A new approach to state estimation and prediction exemplified by alcohol formation in beer brewing. *Journal of Biotechnology*, *24*, 211–234.

Heijnen, R. K., Roels, J. A., & Stouthamer, A. H. (1979). Application of balancing methods in modelling the penicillin fermentation. *Biotechnology Bioengineering*, *21*, 2175–2201.

Horiuchi, J. I., & Kishimoto, M. (2002). Application of fuzzy control to industrial bioprocesses in Japan. *Fuzzy Sets Systems*, *128*, 117–124.

Hunt, K. J., & Sbarbaro, D. (1991). Neural networks for non linear internal model control. *IEE Proceedings*, *138*(5), 431–438.

Isidori, A. (1995). *Nonlinear control systems*. London: Springer.

Jonhson, A. (1987). The control of fed-batch fermentation process—A survey. *Automatica*, *23*, 675–691.

Karniel, A., Meir, R., & Inbar, G. F. (2001). Best estimated inverse versus inverse of the best estimator. *Neural Networks*, *14*, 1153–1159.

Lee, D. S., Jeon, C. O., Park, J. M., & Chang, K. S. (2002). Hybrid neural network modeling of a full-scale industrial wastewater treatment process. *Biotechnology and Bioengineering*, *78*(6), 670–682.

Lennox, B., Montague, G. A., Hiden, H. G., Kornfeld, G., & Goulding, P. R. (2001). Process monitoring of an industrial fed-batch fermentation. *Biotechnology and Bioengineering*, *74*(2), 125–135.

Linko, S., Zhu, Y.-H., & Linko, P. (1999). Applying neural networks as software sensors for enzyme engineering. *TIBTECH*, *17*, 155–162.

Massimo, C., Montague, G., Willis, M., Tham, M., & Morris, J. A. (1992). Towards improved penicillin fermentation via artificial neural networks. *Computers & Chemical Engineering*, *16*(4), 283–291.

McAvoy, T. J., Su, H. T., Wang, N. S., & He, M. (1992). A comparison of neural networks and partial least squares for deconvoluting fluorescence spectra. *Biotechnology and Bioengineering*, *40*, 53–62.

Montague, G. A., Morris, J. A., & Bush, J. (1988). Considerations in control scheme development for fermentation process control. *IEEE Control Systems Magazine*, *8*, 44–48.

Montague, G. A., Morris, J. A., Wright, A. R., Aynsley, M., & Ward, A. (1986). Modelling and adaptive control of fed-batch penicillin fermentation. *The Canadian Journal of Chemical Engineering*, *64*, 567–580.

Morari, M., & Zafiriou, E. (1998). *Robust process control*. Englewood Cliffs, NJ: Prentice-Hall.

Mou, D. G. (1975). *Toward an optimum penicillin fermentation by monitoring and controlling growth through computer-aided mass balancing*. Ph.D. Thesis, MIT, Cambridge, MA.

Narendra, K., & Parthasarathyu, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, *1*(1), 4–27.

Nestaas, E., & Wang, D. I. C. (1983). Computer control of the penicillin fermentation using the filtration probe in conjunction with a structured process model. *Biotechnology and Bioengineering*, *25*, 781–796.

Nielsen, J. (1997). *Physiological engineering aspects of Penicillium chrysogenum*. Singapore: World Scientific.

Nielsen, J., & Villadsen, J. (1992). Modelling of microbial kinetics. *Chemical Engineering Science*, *47*(17/18), 4225–4370.

Paul, G. C., Kent, C. A., & Thomas, C. R. (1993). Viability testing and characterization of germination of fungal spores by automatic image analysis. *Biotechnology and Bioengineering*, *42*, 11–23.

Revilla, G., López-Nieto, J. M., Luengo, M. J., & Martín, J. F. (1984). Carbon catabolite repression of penicillin biosyntesis by *Penicillium chrysogenum*. *Journal of Antibiotics*, *37*(7), 781–789.

Richalet, J. (1993). Industrial applications of model based predictive control. *Automatica*, *29*, 1251–1274.

Rosenblatt, M. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*, 386–408.

Rumelhart, D., & McClelland, J. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition*. Boston: MIT Press.

Shioya, S., Shimizu, K., & Yoshida, T. (1999). Knowledge-base design and operation of bioprocess systems. *Journal of Bioscience and Bioengineering*, *87*(3), 261–266.

Spriet, J. A., Betterman, J., DeBuyser, D. R., Visscher, P. L. D., & Vandamme, E. J. (1982). A computer aided nointerfering on-line technique for monitoring oxygen-transfer characteristics during fermentation processes. *Biotechnology and Bioengineering*, *24*, 1605.

Tiller, V., Meyerhoff, J., Sziele, D., Schügerl, K., & Bellgardt, K. (1994). Seggregated mathematical model for fed-batch cultivation of high-production strain of *Penicillium chrysogenum*. *Journal of Biotechnology*, *34*, 114–131.

Vlassides, S., Ferrier, J. G., & Block, D. E. (2001). Using historical data for bioprocess optimization: Modeling wine characteristics using neural networks and archived process information. *Biotechnology and Bioengineering*, *73*(1), 55–68.

Wang, L. (1994). *Adaptive fuzzy systems and control*. New York: Prentice-Hall.

Warnes, M. R., Glassey, J., Montague, G. A., & Kara, B. (1998). Application of radial basis function and feedforward artificial neural networks to *Escherichia coli* fermentation process. *Neurocomputing*, *20*, 67–82.

Willis, M., Montague, G., Morris, J. A., & Tham, M. (1991). Artificial neural networks: A panacea to modelling problems? In *American control conference*, Boston (pp. 2237–2342).

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, *8*, 338–353.

Zuo, K., & Wu, W. T. (2000). Semi-realtime optimization and control of a fed-batch fermentation system. *Computation of Chemical Engineering*, *24*, 1105–1109.