HYBRID NEURAL NETWORK MODELS OF BIOPROCESSES: A COMPARATIVE STUDY

Grosfils A.[1], Vande Wouwer A.[2] and Bogaerts Ph.[1]

[1] Université Libre de Bruxelles, General Chemistry and Biosystems Department
Ave. F.-D. Roosevelt, 50 C.P. 165/61, B-1050 Brussels (Belgium)
Tel. 32-2-650-36-81, Fax. 32-2-650-35-75, e-mail: aline.grosfils@ulb.ac.be,
[2] Faculté Polytechnique de Mons, Control Engineering Department
Bld. Dolez, 31 B-1700 Mons (Belgium)

Abstract: Modeling of bioprocesses for engineering applications is a very difficult and time consuming task, due to their complex nonlinear dynamic behaviour. In the last years several propositions for hybrid models were published and discussed, in order to combine analytical prior knowledge with the learning capabilities of neural networks. This paper proposes a comparison between several hybrid models based on the two most widespread neural networks, the MultiLayer Perceptron and the Radial Basis Function network. This evaluation relies on simulations of fed-batch bacterial cultures. *Copyright © 2005 IFAC*

Keywords: biotechnology, reactor modeling, mathematical models, neural network, maximum likelihood estimator.

1. INTRODUCTION

Macroscopic models of bioprocesses (cultures of bacteria, yeast or animal cells in bioreactors) are very useful to build engineering tools like simulators, software sensors or controllers. Three kinds of macroscopic models are commonly used in bioprocess modeling. Among them, a first principles model consists in the system of mass balances for the main species appearing in the culture (Bastin and Dochain, 1990). The advantage of such a model is the use of the available prior knowledge about the system; its main drawback is the requirement for a selection of a pseudo-stoichiometry and a kinetic model structure *a priori* badly known. A second macroscopic model structure is the black-box neural network structure (Haykin, 1999; Norgaard, *et al.*, 2000; Montague and Morris, 1994). Neural networks are powerful approximators of arbitrary nonlinear functions but they have strong requirements, in terms of quantity and quality, on the available experimental data. Moreover, they present a lack of transparency because they do not use any prior knowledge. As for the third kind of macroscopic models, a hybrid model combines a partial first principles model, which incorporates the available prior knowledge about the process being modeled, with a neural network which serves as an estimator of unmeasured process parameters (Psichogios and Ungar, 1992; Oliveira, 2004). Among several architectures of hybrid models the parallel and the serial structures are the most common ones (Vande Wouwer *et al.*, 2004). In the parallel structure (Fig. 1), a complete first principles model is connected in parallel with a neural network. The first principles model provides an estimation of the output, while the neural network is trained to compensate the remaining errors between the model and the observed process behaviour. In the serial structure (Fig. 2), an incomplete first principles model is used. The unknown or hardly known terms (i.e. the pseudo-stoichiometry and/or the kinetics) are represented by a neural network.

Combining the advantages of the first principles model and the neural network model without their drawbacks, serial hybrid models, in which the pseudo-stoichiometry and the kinetics are modeled by the neural network, are studied in this work.

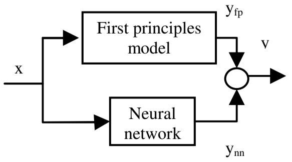

Fig. 1. Parallel hybrid model

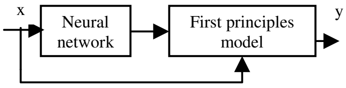

Fig. 2. Serial hybrid model

Models based on the two most commonly used neural networks (Haykin, 1999; Suykens *et al.*, 1996), the MultiLayer Perceptron and the Radial Basis Function network, are compared thanks to simulations of fed-batch bacterial cultures.

In the next section, our serial hybrid modeling approach is presented and the characteristics of the two classes of neural networks considered in this study are discussed in the section 3. Section 4 develops neural parameters identification procedures based on a slightly adapted maximum-likelihood estimator, which determines parameters values while taking the measurements errors into account and improving models generalisation. The simulator of fed-batch bacterial cultures used to build the database is presented in section 5. Section 6 exposes and compares different serial hybrid models on the basis of the validation and cross-validation results as well as the parameters estimation errors. Finally, some conclusions are drawn in section 7.

2. SERIAL HYBRID MODELING OF BIOPROCESS

A very general approach to describe the dynamics of a bioprocess has been proposed in (Bastin and Dochain, 1990). It consists in the system of mass balances for the main culture components:

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

where $\boldsymbol{\xi}$ is the vector of components concentrations, $\mathbf{K}$ the pseudo-stoichiometric coefficients matrix, $\boldsymbol{\varphi}$ the reaction rates, D the dilution rate, $\mathbf{F}$ the vector of external feed rates and $\mathbf{Q}$ the vector of gaseous outflow rates.

Due to the lack of understanding the biological phenomena occurring in the culture, the reaction term $\mathbf{K}\boldsymbol{\varphi}(\boldsymbol{\xi},t)$ is usually arduous to model. To avoid the difficult selection of an appropriate pseudo-stoichiometry and a complex kinetic model structure, the strongly nonlinear reaction term can be represented in (1) thanks to a neural network (Psichogios and Ungar, 1992; Van de Wouwer *et al.*, 2004):

$$\frac{d\xi(t)}{dt} = \mathbf{NN}(\xi,t) - D(t)\xi(t) + \mathbf{F}(t) - \mathbf{Q}(t) \qquad (2)$$

where $\mathbf{NN}(\xi,t)$ is a neural network.

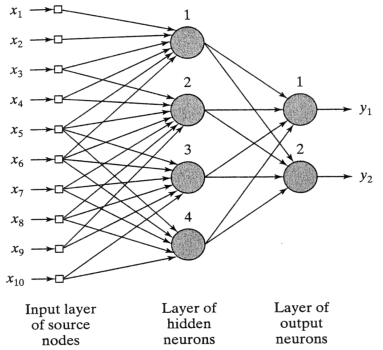

Fig. 3: feedforward neural network (Haykin, 1999)

3. NEURAL NETWORKS CHOICE

The nonlinearities of $\mathbf{K}\boldsymbol{\varphi}(\boldsymbol{\xi},t)$ being purely static, it appears that a simple feedforward neural network is satisfactory. Indeed, a neural network with internal dynamics is not necessary.

A typical feedforward neural network (Fig. 3) consists of massively interconnected simple processing elements (neurons or nodes) arranged in a layered structure, where the strength of each connection is given by an assigned weight. The weights are the internal parameters of the network. The input neurons are connected to the output neurons through layers of hidden nodes. Each neuron receives information in the form of inputs from neurons of previous layers or from the outside world, and processes it through some activation function. Among the different existing feedforward neural networks, the most famous are the MultiLayer Perceptron (MLP) and the Radial Basis Function network (RBF).

The MLP is the most widespread feedforward neural network. Fully connected (each node is connected to each neuron of the following layer), its number of hidden layers can be chosen by the user. Moreover, various activation functions can be used in the different neurons of a MLP; so, it is frequent to observe nonlinear activation functions in the hidden layers while the activation functions of the output neurons are linear (Norgaard *et al.*, 2000). In this work, only MLPs presenting one hidden layer with sigmoid activation functions and a linear output layer are considered. Indeed, Cybenko (1989) proved that such a MLP is a universal approximator, able to approximate any nonlinear continuous function arbitrarily accurately. The mathematical expression of such a network is the following (Suykens *et al.*, 1996):

$$y_i(t) = \sum_{r=1}^{n_{hl}} w_{ir}\, f\left(\sum_{j=1}^{m} v_{rj}\, x_j(t) + \beta_r \right) + b_i \quad i = 1, \ldots, n_{out} \tag{3}$$

where $x_j\ (j = 1 \ldots m)$ are the inputs of the network, $y_i\ (i = 1 \ldots n_{out})$ the outputs of the network, $w_{ir}\ (i = 1 \ldots n_{out}, r = 1 \ldots n_{hl})$ and $b_i\ (i = 1 \ldots n_{out})$ the weights and the biases of the output layer, $v_{rj}\ (r = 1 \ldots n_{hl}, j = 1 \ldots m)$ and $\beta_r\ (r = 1 \ldots n_{hl})$ the weights and the biases of the hidden layer, $f(x) = \dfrac{1}{1 + e^{-x}}$ is the sigmoid activation function.

The most frequent alternative to the MLP is the RBF. Fully connected and universal approximator, this network presents three layers including one hidden layer with radial basis activation functions (in our case, Gaussian functions):

$$y_i(t) = \sum_{j=1}^{n_{hl}} w_{ij}\, f\left(\left\| \mathbf{x}(t) - \mathbf{c}_j \right\| \right) + b_i \quad i = 1, \ldots, n_{out} \tag{4}$$

where $\mathbf{x}(t)$ is the inputs vector of the network, $y_i\ (i = 1 \ldots n_{out})$ the outputs of the network, $w_{ij}\ (i = 1 \ldots n_{out}, j = 1 \ldots n_{hl})$ and $b_i\ (i = 1 \ldots n_{out})$ the weights and the biases of the output layer, $f = e^{-\left\| \mathbf{x} - \mathbf{c}_j \right\|^2 / r_j^2}$ the activation function, $\mathbf{c}_j\ (j = 1 \ldots n_{hl})$ the centres of the activation functions, $\left\| \mathbf{x} - \mathbf{c}_j \right\|$ the Euclidean distance between the input vector $\mathbf{x}$ and the centre $\mathbf{c}_j$, r_j the width of the Gaussian centred on $\mathbf{c}_j$.

4. NETWORKS TRAINING

In this section, identification procedures are developed for the determination of the neural parameters. Moreover, the choice of an optimisation criterion is discussed so as the selection of the number of hidden neurons.

4.1. MLP parameters identification procedure

The MLP identification procedure used in this work is inspired by the RBF identification procedure described in (Vande Wouwer *et al.*, 2004). It consists in a supervised learning and proceeds in several steps:

1. From random values of the weights and biases of the hidden layer, the parameters of the output layer are determined linearly thanks to a least square estimator. This identification step is based on an estimation of the reaction term $\mathbf{K}\boldsymbol{\varphi}(\boldsymbol{\xi}, t)$, a function of the time derivative of the vector $\boldsymbol{\xi}$ which can be obtained by an interpolation model of the measured concentrations (Bogaerts and Hanus, 2000).
2. Starting from the parameters values obtained in step (1), a first nonlinear identification step gives a new estimation of the weights and the biases of the MLP. This time, the identification is based on the simulation of the complete hybrid model and uses an estimator relied on a slightly modified maximum-likelihood criterion which takes the measurements errors into account and improves the generalisation of the MLP (its ability to respond satisfactorily to unknown inputs). This criterion is presented in the subsection 4.3.
3. A second nonlinear identification challenges the weights, biases and initial conditions of the various cultures used for the identification. This last step uses the same estimator as in the second step and can be achieved using a simplex algorithm.

4.2. RBF parameters identification procedure

The RBF identification procedure used in this work is also based on the procedure proposed in (Vande Wouwer *et al.*, 2004). It mixes unsupervised and supervised learning's and presents four identification steps:

1. An unsupervised learning phase realizes a first estimation of the centres and the widths of the Gaussian RBFs. A clustering algorithm, as k-means in MatLab, classifies the data according to similarities among them, organizes these latter into groups and computes the centres of each group. Then the widths are chosen as the mean distances between the different centres.
2. A supervised phase determines the corresponding optimal values of the weights and biases. This identification step is subdivided into:
   - A linear identification step similar to the first identification phase of the MLP parameters.
   - A nonlinear identification step through simulation of the complete hybrid model and numerical minimization of a slightly modified maximum-likelihood cost function quantifying the deviation between the simulated and the real system outputs, taking the measurements errors into account and improving the generalisation of the model.
3. A second nonlinear identification step is then operated, the set of parameters (centres, widths, weights and biases) is estimated again starting from the previous estimations and the above-mentioned slightly modified maximum-likelihood estimator.
4. The last supervised step, based on the same estimator as previously, considers the centres, widths, weights, biases and initial concentrations of the cultures used for the identification. This step can be achieved using a simplex algorithm, supplemented by positivity constraints on some of

the parameters imposed by nonlinear parameter transformation (e.g. logarithmic transformation).

4.3. Optimisation criterion and number of hidden neurons

This section describes the estimator chosen to identify the different parameters nonlinearly.

The simulation model {(2), (3) or (4)} consists of a nonlinear differential system of the form:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t); \boldsymbol{\theta}) \tag{5}$$

where $\mathbf{x}(t) = \xi(t)$ is the state vector containing the concentrations of the main culture components, $\mathbf{u}^T(t) = [D(t) \quad F_1(t) - Q_1(t) \quad \ldots \quad F_N(t) - Q_N(t)]$ a vector containing the dilution rate and the difference between the external feed rates and the gaseous outflow rates, $\boldsymbol{\theta}$ the vector of parameters to be identified and $\mathbf{f}$ the model structure corresponding to {(2), (3) or (4)}.

Let
$$\mathbf{x}(t) = \mathbf{g}(t, \mathbf{u}(t), \mathbf{x}(0); \boldsymbol{\theta}) \tag{6}$$

be the solution (generally obtained by numerical solving) of the differential system (5) starting from the initial concentrations $\mathbf{x}(0)$. Consider the sampled measurements

$$\mathbf{y}_{m,s,k} = \mathbf{g}(t_{s,k}, \mathbf{u}(t_{s,k}), \mathbf{x}_s(0); \boldsymbol{\theta}) + \boldsymbol{\varepsilon}_{y,s,k} \tag{7}$$

where $t_{s,k}$ is the kth sample time of the sth experiment and $\boldsymbol{\varepsilon}_{y,s,k}$ a white noise normally distributed with zero mean and covariance matrix $\mathbf{Q}_{s,k}$.

In order to take the measurements errors into account during the nonlinear identifications, the selected estimator is based on a maximum-likelihood criterion (Bogaerts and Hanus, 2000):

$$V(\theta) = \frac{1}{2}\sum_{s=1}^{S}\sum_{k=1}^{N_S}\Big(\big(\mathbf{y}_{m,s,k} - \mathbf{g}(t_{s,k}, \mathbf{u}(t_{s,k}), \mathbf{x}_s; \boldsymbol{\theta})\big)^T \mathbf{Q}_{s,k}^{-1} \\ \big(\mathbf{y}_{m,sk} - \mathbf{g}(t_{s,k}, \mathbf{u}(t_{s,k}), \mathbf{x}_s; \boldsymbol{\theta})\big)\Big) \tag{8}$$

However, a regularisation term (Norgaard *et al.*, 2000) must be added to this criterion in order to improve the generalisation by minimizing the learning of the noise contained in the training data:

$$W(\boldsymbol{\theta}) = V(\boldsymbol{\theta}) + \boldsymbol{\theta}^T \boldsymbol{\lambda}_I \boldsymbol{\theta} \tag{9}$$

where $\boldsymbol{\lambda}_I$ is a diagonal matrix whose diagonal elements, called weight decays, must be chosen by the user.

The estimation of $\boldsymbol{\theta}$ is then:

$$\hat{\boldsymbol{\theta}} = Arg \min_{\boldsymbol{\theta}} W(\boldsymbol{\theta}) \tag{10}$$

An estimation of the covariance matrix for the parameters estimation errors is given by:

$$E\left[\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}^T\right] \approx \left(\sum_{s=1}^{S}\sum_{k=1}^{N}\mathbf{G}_{\boldsymbol{\theta}}^T(\boldsymbol{\varphi}_{s,k}, \hat{\boldsymbol{\theta}})\mathbf{Q}_{s,k}^{-1}\mathbf{G}_{\boldsymbol{\theta}}(\boldsymbol{\varphi}_{s,k}, \hat{\boldsymbol{\theta}}) + 2\boldsymbol{\lambda}_I\right)^{-1} \\ \left(\sum_{s=1}^{S}\sum_{k=1}^{N}\mathbf{G}_{\boldsymbol{\theta}}^T(\boldsymbol{\varphi}_{s,k}, \hat{\boldsymbol{\theta}})\mathbf{Q}_{s,k}^{-1}\mathbf{G}_{\boldsymbol{\theta}}(\boldsymbol{\varphi}_{s,k}, \hat{\boldsymbol{\theta}})\right) \\ \left(\sum_{s=1}^{S}\sum_{k=1}^{N}\mathbf{G}_{\boldsymbol{\theta}}^T(\boldsymbol{\varphi}_{s,k}, \hat{\boldsymbol{\theta}})\mathbf{Q}_{s,k}^{-1}\mathbf{G}_{\boldsymbol{\theta}}(\boldsymbol{\varphi}_{s,k}, \hat{\boldsymbol{\theta}}) + 2\boldsymbol{\lambda}_I\right)^{-T} \tag{11}$$

where
$$\boldsymbol{\varphi}_{s,k} = \left[t_{s,k} \quad \mathbf{u}^T(t_{s,k}) \quad \mathbf{x}_s^T(0)\right] \tag{12}$$

and
$$\mathbf{G}_{\boldsymbol{\theta}}(\boldsymbol{\varphi}_{s,k}, \hat{\boldsymbol{\theta}}) = \left.\frac{\partial \mathbf{g}(\boldsymbol{\varphi}_{s,k}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} \tag{13}$$

which can be calculated according to a method proposed in (Bogaerts and Hanus, 2000).

The use of the estimator (10) can lead to satisfactory results in validation and cross-validation. Nevertheless, if the use of such a criterion can improve the generalisation, the number of hidden neurons plays also a role in this domain. Indeed, too many hidden neurons can lead to overfitting, noise learning and bad generalisation.

5. SIMULATED FED-BATCH BACTERIAL CULTURE

Consider the following reaction scheme:

$$k\,S \rightarrow X \tag{14}$$

where S denotes the substrate, X the biomass, k the pseudo-stoichiometric coefficient. The mass balance corresponding to this reaction scheme is:

$$\frac{dS(t)}{dt} = -k\,\boldsymbol{\varphi}(\xi, t) - D\,S(t) + D\,S^{in} \tag{15}$$

$$\frac{dX(t)}{dt} = \boldsymbol{\varphi}(\xi, t) - D(t)\,X(t) \quad (6) \tag{16}$$

where $S(t)$ and $X(t)$ are the substrate and biomass concentrations, D is the dilution rate, $\boldsymbol{\varphi}$ the reaction rate and S^{in} the substrate concentration in the feed medium. The reaction rate is described by a Monod law:

$$\boldsymbol{\varphi}(\xi, t) = \frac{\mu_{\max} S(t)}{K_m + S(t)} X(t) \tag{17}$$

The numerical values used in the simulations are the following : $k = 0.5g(10^{11}\,cell)^{-1}$, $K_m = 12\,gl^{-1}$, $\mu_{\max} = 1.4\,h^{-1}$, $S^{in} = 20\,gl^{-1}$.

Thanks to this model, 27 simulations are performed for various values of the initial concentrations in substrate and biomass ($\mathbf{S}_0 = [10 \quad 1 \quad 0.15]\,gl^{-1}$ and $\mathbf{X}_0 = [1.4 \quad 1 \quad 0.4](10^{11}\,cell)l^{-1}$) as well as for different profiles of the external feed rate $\mathbf{F}$ ($F_1(t) = 0.1$, $F_2(t) = 0.01\,t$, and $F_3(t) = 0.1\,t$). Each simulation corresponds to 16h and is corrupted by a white noise normally distributed with zero mean and constant standard deviation equalled to 0.25. The sampling time is 1h.

Among the 27 simulations, only 10 are selected for parameters identification, the remaining experiences being kept for cross-validation, the study of generalisation. The 10 experiences are chosen in order to have a representative sampling of the system behaviour: $(\mathbf{S}_0(1), \mathbf{X}_0(1), \mathbf{F}_1)$, $(\mathbf{S}_0(1), \mathbf{X}_0(1), \mathbf{F}_2)$, $(\mathbf{S}_0(1), \mathbf{X}_0(1), \mathbf{F}_3)$, $(\mathbf{S}_0(1), \mathbf{X}_0(3), \mathbf{F}_1)$, $(\mathbf{S}_0(1), \mathbf{X}_0(3), \mathbf{F}_2)$, $(\mathbf{S}_0(1), \mathbf{X}_0(3), \mathbf{F}_3)$, $(\mathbf{S}_0(3), \mathbf{X}_0(1), \mathbf{F}_1)$, $(\mathbf{S}_0(3), \mathbf{X}_0(1), \mathbf{F}_3)$, $(\mathbf{S}_0(3), \mathbf{X}_0(3), \mathbf{F}_1)$, $(\mathbf{S}_0(3), \mathbf{X}_0(3), \mathbf{F}_3)$.

6. COMPARISON BETWEEN HYBRID MODELS

As mentioned previously some factors must be selected before building a hybrid first principles neural network model: the number of hidden neurons and the weight decays values. However, there are no systematic procedures able to determine *a priori* the network size or the optimal weight decays values. Indeed, the better way to determine these factors is the trial and error method.

In order to compare hybrid models based on MLP or RBF, models with different numbers of hidden neurons are identified in this work thanks to the procedures of section 4 with different weight decays values.

6.1. Tested weight decays and numbers of hidden neurons

As pointed out above, different numbers of hidden neurons must be tested while keeping in mind that the number of parameters has to be limited to avoid overfitting. So only two and three hidden neurons models are tested here and identified thanks to the criterion (9) with different weight decays values. These latter values are chosen according to the following notice: the regularisation term $\boldsymbol{\theta}^T \boldsymbol{\lambda}_I \boldsymbol{\theta}$ implies that $\boldsymbol{\theta}$ belongs to a normal distribution centred at $\mathbf{0}$, with covariance matrix $\boldsymbol{\lambda}_I^{-1}$. Hence, $\boldsymbol{\theta}^T \boldsymbol{\lambda}_I \boldsymbol{\theta}$ can be interpreted in (9) as too high parameters sanction. In this way, null weight decays are selected for biases, centres and widths while the different weights present positive weight decays. Indeed, while the biases provide a continuous component to the neural network output signal, the weights are only able to induce some variations around this component. To avoid the learning of the noise contained in the training data, low variations and thus low weights values seem to be better. Concerning the centres and the widths of a RBF, there is *a priori* no reason to restrict the state space which has to be covered. Note that the positive weight decays considered in this work are equalled to each other and noted λ. The weight decays values used in this work are presented in Table 1.

Table 1. Tested weight decays values for hybrid models with 2 or 3 hidden nodes

λ	RBF	MLP
3 neurons	0, 8, 20	0, 8, 20
2 neurons	8, 20	8, 20

Table 2. Parameters values and 95% interval confidence of best 3 hidden neurons hybrid models

MLP	$\mathbf{V} = \begin{bmatrix} 0.053 \pm 0.008 & 0.163 \pm 0.007 \\ 1.2 \pm 0.1 & 0.34 \pm 0.08 \\ 1.2 \pm 0.1 & 0.00 \pm 0.01 \end{bmatrix}$ $\boldsymbol{\beta} = \begin{bmatrix} -0.46 \pm 0.07 \\ -0.98 \pm 0.3 \\ -0.02 \pm 0.3 \end{bmatrix}$ $\mathbf{W} = \begin{bmatrix} -8.11 \pm 0.08 & 3.23 \pm 0.04 & -7.89 \pm 0.07 \\ 16.58 \pm 0.05 & -23.85 \pm 0.01 & 15.25 \pm 0.04 \end{bmatrix}$ $\mathbf{B} = \begin{bmatrix} 8.78 \pm 0.08 \\ -0.1 \pm 0.2 \end{bmatrix}$
RBF	$\mathbf{C} = \begin{bmatrix} 0.34 \in [0.07 \quad 1.55] & 23.12 \in [22.86 \quad 23.38] \\ 5.96 \in [4.48 \quad 7.93] & 1.00 \in [0.02 \quad 51.16] \\ 11.10 \in [10.63 \quad 11.58] & 20.06 \in [19.69 \quad 20.42] \end{bmatrix}$ $\mathbf{r} = \begin{bmatrix} 9.59 \in [9.22 \quad 9.97] \\ 49.81 \in [43.88 \quad 56.54] \\ 11.88 \in [11.26 \quad 12.54] \end{bmatrix}$ $\mathbf{W} = \begin{bmatrix} 3.06 \pm 0.05 & 7.45 \pm 0.1 & -5.33 \pm 0.10 \\ -8.03 \pm 0.07 & -18.414 \pm 0.06 & 13.86 \pm 0.06 \end{bmatrix}$ $\mathbf{B} = \begin{bmatrix} -7.6 \pm 0.2 \\ 18.1 \pm 0.3 \end{bmatrix}$

6.2. Results and discussion

In this subsection, the different hybrid models appearing in the Table 1 are compared on the basis of the validation and cross-validation results as well as the parameters estimation errors.

The results obtained for hybrid models with two hidden neurons are not satisfactory in simple validation. Indeed, the difference between the training data and the simulated signals is too high. However, for 3 hidden neurons models, the results are better. Two models distinguish themselves: a

RBF hybrid model and a MLP hybrid model, both identified with a high value of λ, are superior in simple validation (Fig 4). Their parameters values are contained in Table 2 in their 95% confidence interval.

Although the 95% confidence intervals seem to be better for the MLP, the RBF is preferred. Indeed, the cross-validation results (Fig. 5) show that the RBF generalises better than the MLP. Moreover, remind another RBF quality: the first estimation of the centres and the widths are done automatically thanks to a clustering algorithm while the first estimation of the hidden parameters of the MLP is obtained randomly which implies the requirement of different identifications from diverse initial estimations of hidden parameters.

7. CONCLUSION

In this paper, several hybrid models based on MLP or RBF neural networks are studied and compared thanks to simulated data of fed-batch bacterial cultures. The considered hybrid models have a serial structure: the neural network models the whole reaction term including the pseudo-stoichiometry and the kinetics.

After presenting neural parameters identification procedures, the choice of the optimisation criterion and the number of hidden neurons is discussed. On the basis of this discussion, several models with different numbers of hidden neurons, identified for various weight decays values, are compared. As a result of this comparison, RBF hybrid models appear superior to MLP hybrid models. Indeed, for the same number of hidden neurons a hybrid model based on a RBF identified thanks to high weight decays gives better results in cross-validation than a hybrid model based on a MLP. Hence, the RBF allows having a better generalisation than the MLP.

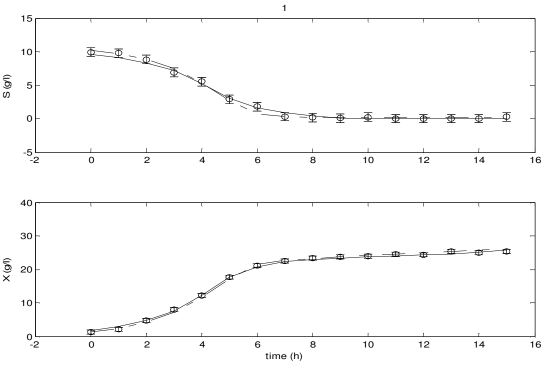

Fig. 4. Simple validation of the 3 hidden nodes models, culture $(\mathbf{S}_0(1),\mathbf{X}_0(1),\mathbf{F}_1)$, o: measurements (with their associated 95% confidence intervals), ._: MLP simulated values, __: RBF simulated values.

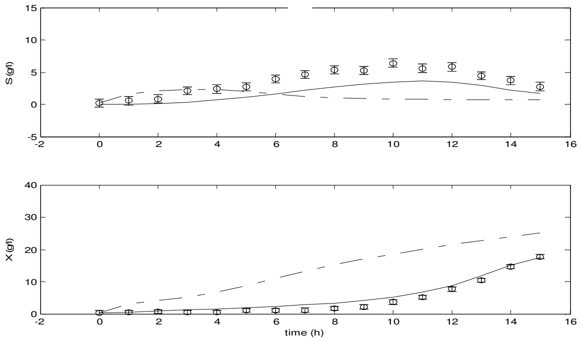

Fig. 5: Cross-validation of the 3 hidden nodes models, culture $(\mathbf{S}_0(3),\mathbf{X}_0(3),\mathbf{F}_2)$, o: measurements (with their associated 95% confidence intervals), ._: MLP simulated values, __: RBF simulated values.

ACKNOWLEDGEMENTS

The authors are grateful to F.R.I.A for its support.

REFERENCES

Bastin, G. and D. Dochain (1990). *On-line estimation and adaptative control of bioreactor*. Elsevier, Amsterdam.

Bogaerts Ph. and R. Hanus (2000). Macroscopic modelling of bioprocesses with a view to engineering applications. in: *Focus on Biotechnology* (M. Hofman and J. Anne, Eds), vol. IV Engineering and Manufacturing for Biotechnology (Ph. Thonart and M. Hofman, Eds), Kluwer Academic Publishers, 77-110.

Cybenko G. (1989), Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, **2** (4), 303-314.

Haykin S. (1999). *Neural Networks: A comprehensive foundation* Prentice Hall, 2d edition.

Montague G. and J. Morris (1994). Neural-network contributions in biotechnology. *Trends in Biotechnology*, 12, 312-324.

Norgaard M., O. Ravn, N. K. Poulsen and L. K. Hansen (2000). Neural Networks for Modelling and Control of Dynamic Systems: A Practionner's Handbook. Springer, London.

Oliveira R. (2004), Combining first principles modelling and artificial neural networks: a general framework, *Computers & Chemical Engineering,* Vol **28** (5), 755-766

Psichogios D. C. and L. H. Ungar (1992). A hybrid neural network-first principles approach to process modelling. In: *A.I.Ch.E Journal*, Vol. **38**, 10, 1499-1511.

Suykens J. A. K., Vandewalle J.P.L and B.L.R. De Moor (1996), *Artificial Neural Networks for Modelling and Control of Non-Linear Systems.* Kluwer Academic Publishers.

Vande Wouwer A., C. Renotte and Ph. Bogaerts (2004). Biological reaction modeling using radial basis function networks. In: *Computers and Chemical Engineering,* **28**, 2157-2164.