Modelling of methanol synthesis in a network of forced unsteady-state ring reactors by artificial neural networks for control purposes

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

Fissore D., Barresi A. A., Manca D. (2004). Modelling of methanol synthesis in forced unsteady-state ring reactors networks by Artificial Neural Networks for control purposes. Chemical Engineering Science (Elsevier), 59(19), 4033-4041. DOI: 10.1016/j.ces.2004.05.033.

Modelling of methanol synthesis in a network of forced unsteady-state ring reactors by Artificial Neural Networks for control purposes

Davide Fissore[1], Davide Manca[2], Antonello A. Barresi[1,*]

[1]Politecnico di Torino, Dipartimento di Scienza dei Materiali e Ingegneria Chimica, Corso Duca degli Abruzzi 24, 10129 Torino, Italy.
Tel. +39-011-5644658, Fax: +39-011-5644699, E-mail: antonello.barresi@polito.it

[2]Politecnico di Milano, Dipartimento di Chimica, Materiali e Ingegneria Chimica, Piazza Leonardo da Vinci 32, 20133 Milano Italy
Tel. +39-02-23993271, Fax: +39-02-70638173, E-mail: davide.manca@polimi.it

[*] author to whom all the correspondence should be addressed.

Abstract

A numerical model based on Artificial Neural Networks (ANN) was developed to simulate the dynamic behaviour of a three reactors network (or ring reactor), with periodic change of the feed position, when low pressure methanol synthesis is carried out. A multilayer, feedforward, fully connected ANN was designed and the History Stack Adaptation (HSA) algorithm was implemented and tested with quite good results both in terms of model identification and learning rates. The influence of the ANN parameters was addressed, leading to simple guidelines for the selection of their values. A detailed model was used to generate the patterns adopted for the learning and testing phases. The simplified model was finalised to develop a Model Predictive Control (MPC) scheme in order to maximise methanol yield and to fulfil process constraints.

Keywords

Artificial Neural Network, History Stack Adaptation, Forced unsteady-state chemical reactor, Methanol synthesis, Mathematical modelling, Dynamic simulation.

1. Introduction

Forced unsteady-state operation of catalytic reactors has been discussed in the chemical engineering literature since the mid Sixties. Virtually, almost all reactor inputs can be forced periodically in order to produce a transient operation, but variations in flow rate, feed composition and flow direction are generally considered. The last case, usually referred to as periodically reversed flow operation, was developed by Boreskov & Matros (1983), who firstly described the behaviour of a catalytic fixed bed reactor under transient conditions forced by a periodic reversal of the gas flow.

Two are the major advantages of unsteady-state operation: first, the possibility of exploiting the thermal storage capacity of the catalyst bed, which acts as a regenerative heat exchanger, thus allowing auto-thermal behaviour even at low reactants concentration (Kolios, Frauhammer & Eigenberger, 2000); second, an approach toward optimum temperature distribution, which makes possible the creation of favourable thermodynamic conditions for exothermic equilibrium-limited reactions. It was shown (Vanden Bussche & Froment, 1993) that the application of reverse flow operation to methanol synthesis could be economically attractive in comparison with steady-state technology. Nevertheless, the reverse-flow reactor presents the problem of wash-out, i.e. the drop in product concentration upon flow reversal, due to the removal of unconverted gas immediately after the reversal of the flow direction. Vanden Bussche & Froment (1996) proposed the concept of STAR reactor, which can operate in a transient mode giving practically constant exit concentrations and higher conversion than the reverse flow apparatus. Velardi & Barresi (2002) investigated the feasibility of carrying out the low pressure methanol synthesis in a network of three catalytic fixed bed reactors with periodic change in the feeding position: advantages and limitations in comparison with the previously proposed reverse-flow reactor were highlighted, pointing out that a cyclic-steady-state condition and auto-thermal behaviour are possible but they are attainable only when the switching time is chosen in two narrow ranges. Out of these regions, complex steady-states of high periodicity, with low conversion or extinction of the reactors, may occur (Velardi *et al*, 2003). For low values of the switching time, the establishing of optimal temperature profiles in the network produces significantly higher conversions than in the reverse-flow reactor. Furthermore, the performance of the network is weakly affected by wash-out.

In order to exploit the higher yield and selectivity, which may be achieved in such a device, and to deal with unexpected external perturbations in the feed temperature and flow rate, which may lead to reactor extinction or conversion decrease, an advanced process control has to be designed.

Model Predictive Control (MPC) seems to be attractive for this kind of process in order to maximise methanol yield and to fulfil process constraints. The on-line optimisation requires the use of a simplified model since the time, which is usually required for the solution of a detailed model is not compatible with the control time. For this reason, Artificial Neural Networks (ANNs) were used to simulate the process, as they require few milliseconds to evaluate the dynamic behaviour of the system and the response of the process to external disturbances.

ANNs are nowadays extensively employed in different branches of science and technology in such diverse fields as modelling, time series analysis, pattern recognition, signal processing, and control by way of an important property: an ANN may be thought of as a black box that can accept a series of input data and produce one or more outputs. The transformation of the data is performed by several basic processing units, called artificial neurons or simply neurons, which perform identical tasks. The neurons are connected into networks by synapses or connecting links. The problems handled by ANNs may be quite assorted. One of the most important is modelling, i.e. the search for an analytical function or a procedure that gives a specified n-variable output for any m-variable input. Standard modelling techniques require the mathematical function to be known in advance. Conversely, the ANN does not require the knowledge of such a function: the nonlinearity of a single unit transformation and a sufficiently large number of variable parameters (weights and biases) ensure enough "freedom" to adapt the neural network to any relation between input and output data (see for example Haykin, 1999; Zupan & Gasteiger, 1999).

To create a conventional model of a chemical process all phenomena present in the process have to be identified and properly described. Mathematical description of the process - usually containing conservation laws, chemical, physical and chemical-physical processes – is expressed by a set of deterministic (algebraic, differential or integral) equations. In a neural model, the ANN replaces some portions of the conventional model and two general methods can be distinguished: a global neural model (GNM), in which the whole model is represented with a single ANN, and a hybrid neural model

(HNM), in which only the unknown part of the model is replaced by the ANN. The first approach (GNM), since its introduction in the early nineties (Bhat & McAvoy, 1990), is commonly used mainly for control and regulation purposes. This approach makes possible the modelling of the reactor based on the chosen input-output signals only, without any fundamental knowledge about the system being modelled. Nevertheless, in most practical cases a lot of information is known about the modelled system and only some elements are not sufficiently documented. From this observation, an idea of neural hybrid model has been derived by Psichogios & Ungar (1992), in which the ANN is used to approximate those unknown elements. In the present work a GNM was used as this approach is more convenient for rapid modelling of the reactor. The main disadvantage is the poor ability to generalise the results of modelling, which will be valid in the range of input values used in the training phase.

The aim of this work is to apply for the first time ANN to simulate the dynamic behaviour of forced unsteady-state reactors and to give general guidelines for the design of the ANN. Low-pressure methanol synthesis in a three reactors network will be used as a case study. The real process behaviour is dynamically simulated by using a detailed non-linear time domain model. The History Stack Adaptation algorithm (Tadé, Mills & Zomaya, 1996) is used for the training phase and the influence of the main parameters of the algorithm is discussed.

2. Neural network architecture and learning algorithm

The architecture of the ANN is variable depending on the complexity of each individual process and the objectives for using them. In this work a feedforward multilayer network will be used (Figure 1); 4 layers are present: the first (input) layer contains n_1 nodes corresponding to the actual net inputs; two hidden layers contain respectively n_2 and n_3 nodes and the fourth (output) layer contains n_4 nodes that correspond to the number of monitored state variables. Cybenko (1989) showed that neural networks implementing one, sufficiently large, hidden layer can uniformly approximate any continuous function of n real variables to any desired accuracy. In this work two hidden layers have been used with the aim of improving the performance of the ANN. The number of input and output nodes is governed by the functional requirements of the ANN. No general guideline is available for the number of

hidden nodes on each layer: as a consequence the training process was repeated for different numbers of hidden nodes and little, if any, improvement in ANN performance was observed. This ANN is fully connected as every node in each layer of the network is connected to every other node in the adjacent forward layer. No back connections or recycles were implemented.

Figure 2 shows the model of a neuron, which forms the basis for designing ANNs. It is possible to identify three basic elements of the neuronal model:

1. a set of synapses or connecting links, each characterised by a weight. Specifically, a signal u_j at the input of the synapse j connected to a neuron k is multiplied by the synaptic weight w_{kj}, which may lie in a range that includes negative as well as positive values;

2. an element for summing the input signals, weighted by the respective synapses of the neuron and corrected by a bias value;

3. an activation function for defining the amplitude of the output of a neuron.

Actually, the neuronal model of Figure 2 also includes an externally applied bias, denoted by b_k whose effect is increasing or lowering the net input of the activation function, depending on whether it is positive or negative, respectively. The use of a bias has the effect of applying an affine transformation to the output of the linear combiner in the model of Figure 2:

$$q_k = u_k + b_k \tag{1}$$

The activation function defines the output of a neuron in terms of q_k and can take several forms. In this study, according to Tadé, Mills & Zomaya (1996) two different activation functions, sigmoid and linear, were used for the hidden and output layers respectively:

$$\varphi(q) = \begin{cases} \mu q & \text{output layer} \\ \dfrac{1}{1+e^{-q}} & \text{hidden layers} \end{cases} \tag{2}$$

Learning of the input-output mapping is accomplished by repeated presentations of a sequence of patterns consisting of the input and corresponding target or desired output belonging to a fixed training set. A robust learning heuristic for multilayered feedforward ANNs is the generalised delta rule (GDR) or backpropagation: this method is a supervised learning

method as the weights are corrected so as to produce prespecified target values for as many inputs as possible (Zurada, 1992). This algorithm was used because of its simplicity and ease of use. The weights and bias of each node are initialised to small random values at the start of the learning procedure. The correction of weights and biases is made after each new input is proposed to the ANN.

During learning, the input vector U is presented to the neural network and the output vector s is immediately compared with the target vector $\overline{S}$ $(\overline{s}_1, \overline{s}_2, ..., \overline{s}_m)$ that is the correct output for U. Once the actual error produced by the network is evaluated, this is used to modify the weights and biases throughout the entire ANN:

$$\begin{aligned} \Delta w_{ji}^{l} &= \alpha \delta_j^l s_i^{l-1} + \beta \Delta w_{ji}^{l(previous)} \\ \Delta b_j^{l} &= \alpha \delta_j^l s_i^{l-1} + \beta \Delta b_j^{l(previous)} \end{aligned} \tag{3}$$

where l is the index of the current layer, j identifies the current neuron and i is the input source, i.e. the index of the neuron in the upper layer. In this equation δ_j^l, the error introduced by the corresponding neuron, is calculated in two ways, depending on whether the last (output) layer or one of the hidden layers is under consideration:

$$\delta_j^{last} = \left(\overline{s}_j - s_j^{last}\right) s_j^{last} \left(1 - s_j^{last}\right) \tag{4}$$

$$\delta_j^{l} = \left(\sum_{k=1}^{r} \delta_k^{l+i} w_{kj}^{l+1}\right) s_j^{l} \left(1 - s_j^{l}\right) \tag{5}$$

According to eq. (3) the correction of the weights in the l-th layer comprises two terms that pull in opposite directions: the first one tends towards a fast "steepest-descent" convergence, while the second is a longer-range function that prevents the solution from getting trapped in shallow local minima. The learning rate is α, whilst β is the momentum factor. By taking into account the correction made on the previous cycle, β can prevent sudden changes in the direction in which corrections are made: this is particularly useful for damping oscillations. The value of the learning rate and momentum constant is generally obtained by trial and error. Some guidelines can be found in the literature (Tadé, Mills & Zomaya, 1996; Haikin, 1999). The expressions (4) and (5) can be derived from the delta rule (see for example Zupan & Gasteiger, 1999) and correspond to a discrete-time gradient descent rule that minimises a function of the error between target output and network output for each output element

and pattern in the training set. The procedure is then repeated with the other input-output pairs.

The convergence of the ANN may be strongly improved by using the history-stack adaptation (HSA) method instead of the single pattern presentation (SPP) described above. When each pattern is presented to the ANN the weights and biases are updated in order to minimise the error for that individual pattern. The variation is determined by the learning rate, α, which is generally set to a small value. Consequently, the information contained in a single pattern cannot be assimilated completely in a single presentation and part of the information value is discarded and not absorbed by the learning procedure. Since the patterns are not fully learned in one presentation, this suggests that a method using several replications of each pattern is most appropriate. The HSA method operates by means of a First-In-First-Out stack (containing n_p patterns), which, at each time step, accepts a new pattern from the process and discards the oldest pattern from the stack. The elements of the stack are each used in n_c cycles to update the weights and biases at each time-step. Consequently, each pattern is used $n_p n_c$ times before it is discarded, thereby improving the potential adaptation performance achieved at each time-step. Tadé, Mills & Zomaya (1996) have suggested a few design guidelines for the parameters n_p and n_c of the stack procedure.

3. Neural network modelling of the reactors network

Velardi & Barresi (2002) pointed out that in the three reactors network a maximum value of carbon to methanol conversion of 58% may be obtained when t_c = 40 s, $T_{G,\,in}$ = 130°C, with v_G = 0.021 m s^{-1}. This value is higher than the conversion of 30-40% that may be achieved in the traditional multi-bed adiabatic reactors. Nevertheless, changes in the inlet gas flow rate and temperature may lead either to lower conversions or to reaction extinction or pseudo-periodic behaviours. Moreover, a tighter control on the outlet methanol conversion is needed to maximise methanol conversion. Because of these requirements as well as the complexity of the dynamics of the process, a Model Predictive Control (MPC) scheme should be adopted. Dufour, Couenne & Touré (2003) designed a MPC algorithm to control a catalytic reverse-flow reactor where the combustion of lean mixtures of volatile organic compounds is carried out: the analogy with the countercurrent reactor when the switching

frequency approaches infinity was used to obtain a simple model to be solved on-line. As a consequence, the switching time could not be used as a manipulated variable and an external heating device was introduced for control purposes and, specifically, to prevent reaction extinction. Contrarily to this approach, in the case of methanol synthesis the control algorithm has to vary the switching time to maximise the selectivity of the reactor, as there are no other parameters that can be manipulated. Actually, it was shown that such a parameter is effective in controlling the process.

As far as the output variables of the ANN are concerned, they are chosen as the system variables to be controlled, i.e. methanol outlet molar fraction. Since the GNM approach was adopted, the input variables are both the process disturbances (gas temperature and flow rate) and the manipulated variable (switching time). Application of the feedforward neural network (which is a static structure) to describe a dynamic behaviour of the reactor, is possible due to the introduction of a pseudo-dynamic structure (Levin & Narendra, 1995):

$$S(t+1) = f\left(S(t), S(t-1), ..., S(t-t_1), U(t), U(t-1), ..., U(t-t_2)\right) \quad (6)$$

where S and U are respectively the output and input vectors, t_1 and t_2 are the appropriate time delays and the unknown nonlinear function f is approximated by the ANN. In this work the output at time t is given by the input at time t-1, t-2 and t-3 as it is sketched in Figure 3; each pattern is made up of the input variables (switching time, inlet flow rate and temperature) evaluated at t, t-1 and t-2, while the output variable is evaluated at t-1, t-2 and t-3 to take into account the time delay of the input on the system response.

3.1 *Patterns generation*

Training of the ANN was performed with data produced by the numerical simulation of a detailed first principles model of the process. Actually, the detailed model cannot be used for control purposes in the MPC procedure, since its numerical structure is too time consuming and not so robust as the ANN one. A one-dimensional heterogeneous model was used to simulate the behaviour of the reactor. The pressure loss inside the network of adiabatic reactors is neglected and a plug flow condition is assumed for the gas phase with dispersive transport of mass and energy. The transient term is taken into account in the gas phase equations and in the energy equation for the solid phase, whilst the solid catalytic surface is considered in pseudo-steady state

condition. The effect of the intraparticle mass transport was included in the model by estimation of the effectiveness factors, using the linearization method proposed by Gosiewski, Bartmann, Moszczynski & Mleczko (1999). Thus, the dynamics of the process can be described by the following set of differential-algebraic equations (DAE):

- Continuity equation for the gas phase:

$$\frac{\partial c_G}{\partial t}+\frac{\partial}{\partial x}c_G v=\sum_{i=1}^{n_r}\frac{k_{G,i}a_v}{\varepsilon}(y_{S,i}-y_{G,i}) \tag{7}$$

- Continuity equation for component *j* in the gas phase:

$$\begin{aligned}\frac{\partial y_{G,j}}{\partial t}=D_{eff}\frac{\partial^2 y_{G,j}}{\partial x^2}-v\frac{\partial y_{G,j}}{\partial x}+\frac{k_{G,j}a_v}{c_G\varepsilon}\left(y_{S,j}-y_{G,j}\right)+\\ -y_{G,j}\sum_{i=1}^{n_r}\frac{k_{G,i}a_v}{c_G\varepsilon}\left(y_{S,i}-y_{G,i}\right)\quad\text{with}\quad j=1\ldots(n_r-1)\end{aligned} \tag{8}$$

- Energy balance for the gas phase

$$\frac{\partial T_G}{\partial t}=\frac{k_{eff}}{\varrho_G\hat{c}_{P,G}}\frac{\partial^2 T_G}{\partial x^2}-v\frac{\partial T_G}{\partial x}+\frac{ha_v}{\varrho_G\hat{c}_{P,G}\varepsilon}\left(T_S-T_G\right) \tag{9}$$

- Mass balance for the solid phase:

$$k_{G,j}a_v\left(y_{S,j}-y_{G,j}\right)=\left[\varrho_S(1-\varepsilon)\right]\sum_{k=1}^{N_R}\eta_k\nu_{j,k}R_k'\quad\text{with}\quad j=1\ldots n_r \tag{10}$$

- Energy balance for the solid phase:

$$\begin{aligned}\frac{\partial T_S}{\partial t}=\frac{\lambda_S}{\varrho_S\hat{c}_{P,S}}\frac{\partial^2 T_S}{\partial x^2}-\frac{ha_v}{\varrho_S\hat{c}_{P,S}(1-\varepsilon)}(T_S-T_G)+\\ +\frac{1}{\hat{c}_{P,S}}\sum_{i=1}^{n_r}\left(\sum_{k=1}^{N_R}\eta_k\nu_{i,k}R_k'\right)(-\Delta\tilde{H}_{f,i})\end{aligned} \tag{11}$$

Danckwerts boundary conditions were assumed for the gas phase in each reactor and the continuity of gas temperature and concentration profiles was imposed among the reactors of the network, i.e. in sections at $x=K\ell$, with $K=1,2$. In the following we refer to L as the total length of the network and to $\ell=\frac{L}{3}$ as the length of a single reactor of the sequence. The origin of the x-axis corresponds to the inlet section to the network. Consequently, it translates from the first reactor of the sequence to the second one when the switching time is

reached. At that time, the boundary conditions are switched in order to simulate the variation of the inlet position. Initially, gas and solid phase temperatures are considered equal and constant along the reactors; the initial reactants concentration is null.

The model is completed by the kinetic equations by Graaf, Stamhuis & Beenackers (1988), corresponding to a dual-site Langmuir-Hinshelwood mechanism, based on three independent reactions: methanol formation from CO, water-gas-shift reaction and methanol formation from CO_2:

(A) $$CO+2H_2 \rightleftharpoons CH_3OH \tag{12}$$

(B) $$CO_2+H_2 \rightleftharpoons CO+H_2O \tag{13}$$

(C) $$CO_2+3H_2 \rightleftharpoons CH_3OH+H_2O \tag{14}$$

The reaction rates for methanol and water from reactions (A), (B) and (C) are given by the following equations, according to Graaf, Sijtsema, Stamhuis & Joosten (1988):

$$R'_{CH_3OH,\,A} = \frac{k'_{ps,A} K_{CO}\left[p_{CO} p_{H_2}^{3/2} - p_{CH_3OH}\Big/\left(p_{H_2}^{1/2} K_{p,A}\right)\right]}{\left(1+K_{CO}p_{CO}+K_{CO_2}p_{CO_2}\right)\left[p_{H_2}^{1/2}+\left(K_{H_2O}\Big/K_{H_2}^{1/2}\right)p_{H_2O}\right]} \tag{15}$$

$$R'_{H_2O,\,B} = \frac{k'_{ps,B} K_{CO_2}\left(p_{CO_2} p_{H_2} - p_{H_2O} p_{CO}/K_{p,B}\right)}{\left(1+K_{CO}p_{CO}+K_{CO_2}p_{CO_2}\right)\left[p_{H_2}^{1/2}+\left(K_{H_2O}\Big/K_{H_2}^{1/2}\right)p_{H_2O}\right]} \tag{16}$$

$$R'_{CH_3OH,\,C} = R'_{H_2O,\,C} = \frac{k'_{ps,C} K_{CO_2}\left[p_{CO_2} p_{H_2}^{3/2} - p_{CH_3OH} p_{H_2O}\Big/\left(p_{H_2}^{3/2} K_{p,C}\right)\right]}{\left(1+K_{CO}p_{CO}+K_{CO_2}p_{CO_2}\right)\left[p_{H_2}^{1/2}+\left(K_{H_2O}\Big/K_{H_2}^{1/2}\right)p_{H_2O}\right]} \tag{17}$$

Transport and dispersion parameters were evaluated similarly to previous works, adopting the same correlations of Velardi & Barresi (2002).

The partial differential equation system (7)-(11) was solved by discretising the domain of the spatial variable x thus obtaining a DAE system. For the algebraic part, given by the mass balances (10) for the solid phase, the non-linear equations solver HYBRID1 from the FORTRAN package MINPACK (Moré, Garbow & Hillstrom, 1981) was used, while the routine LSODE from ODEPACK library (Hindmarsh, 1983) was adopted to solve the differential portion of the system.

The conditions adopted in the simulations are given in Table 1. They are the same as those previously considered for the unsteady state operation of methanol synthesis in a reverse-flow reactor by Vanden Bussche, Neophytides,

Zolotarskii & Froment (1993) and in the three reactors network by Velardi & Barresi (2002).

We have assumed that the disturbances in the inlet gas temperature are in the range ±30 K and for the inlet gas flow rate in the range ±20% around the values corresponding to the optimal configuration. Finally, the value of the switching time in bounded in the interval 40 ± 15 s. The input patterns were generated according to a Generalised Random Sequence that randomly changes the values of T_{in}, $v_{G,in}$ and t_c in the above indicated intervals. Figure 4 shows an example of the values of the inlet gas temperature that have been considered for the pattern generation. Firstly 3000 random values of the switching time and of the inlet gas temperature were generated, then a time series has been built in with the sequence of the switching times and each time that t_c is changed also T_{in} takes a different value. Such data were partitioned into two sets of 1500 values, used respectively for training and testing. The patterns were organised into a stack according to the HSA algorithm. A value of n_p=100 for the patterns in the stack and n_c=10 for the number of repetitions of the stack were implemented, according to Tadé, Mills & Zomaya (1996), due to the complexity and variety of the dynamics of the reactor.

3.2 *Results and discussion*

Since the ANN is intended for control purposes, the sampling time corresponds to the control time, i.e. the time interval between two subsequent control actions. Working with the MPC technique it is necessary to check if an exhaustive search of the optimum of the control objective function can be performed within such an interval. The sampling time has also to be lower than 15 s, which is the lower limit of the switching time: a value of 10 s was chosen as an initial guess. The influence of the parameters α, β and μ and of the number of patterns used during the training was firstly investigated. The results are summarised in Figure 5, where the normalised mean square error between the prediction of the model and that of the ANN (E_{RMS}) is plotted versus the number of patterns used for the learning. It is immediately evident that few hundreds of patterns are sufficient to reach negligible values of the E_{RMS}. On the contrary, the traditional approach, based on the single pattern presentation, requires tenths and even hundreds of thousands of patterns. With reference to the influence of the parameters: the higher is α, the lower is E_{RMS} (graph A). The optimal range of values of β is often poorly understood, despite

its apparent simplicity: even if in the literature values of β in the range 0.7-0.99 are suggested, in this process small values of β produce better results. The choice of the gain μ has clearly an influence on the effective learning rate of the weights of the output node (see eq. (2)). Even if a value of μ = 1 is generally suggested as a starting guess, this parameter can be more rigorously optimised: the gain of the sigmoid function is a quadratic function of the activation φ:

$$\frac{\partial \varphi}{\partial q} = \frac{e^{-q}}{\left(1+e^{-q}\right)^2} = \varphi(1-\varphi) \tag{18}$$

and this function has a maximum at:

$$\frac{\partial^2 \varphi}{\partial q^2} = 0 = -2\varphi + 1 \quad \Rightarrow \quad \varphi = 0.5 \tag{19}$$

The aim is to have a similar gain for the linear node. Obviously it is impossible to have the two gains matching for all the values of α. However, approximately, similar gains over the useful range may improve learning. As a consequence, given the range of the input values to the sigmoid function in a trained network, it is possible to calculate the (integral) average gain over this interval and to assume μ equal to this value. For our process, an optimal value of μ of 0.1 was obtained from this calculus, even if lower values may lead to better performance. In conclusion the "best" choice of the network parameters is α = 0.95, β = 0.1 and μ = 0.1.

Beyond the set of patterns generated for testing the performance of the ANN, further tests were accomplished, comparing the predictions of the ANN with those of the model in presence of different disturbances of the inlet parameters of the RN. Figure 6 shows the results of this comparison when the inlet gas temperature is increased ($\Delta T_{G,in}$ = + 30°C, upper graph) and decreased ($\Delta T_{G,in}$ = -20°C, lower graph): different dynamic behaviours may occur, but in both cases considered the agreement between the model and the ANN is excellent, and the percentage error is always less than 1%. Predictions and actual values are even closer when variations in the flow rate are considered as it is shown in Figure 7 ($\Delta v_{G,in}$ = + 10%, upper graph; $\Delta v_{G,in}$ = - 10%, lower graph). The predictions of the ANN and that of the model were also compared when the switching time is changed, as this is the parameter used by the MPC algorithm for control purposes. Figure 8 shows the result of this comparison when the switching time is increased (Δt_c = + 10s, upper graph) and decreased (Δt_c = - 10s, lower graph).

From these results it is evident that the ANN is able to correctly predict the dynamics of the system even when a pseudo-periodic state and complex dynamics are present with a total CPU time of some milliseconds. This is in our opinion the keynote that makes ANN so appealing for advanced control purposes. Evidently, the performance of the ANN worsens when the input parameters fall outside the range used for the learning. In this case it is necessary to repeat the learning, but the choice of the parameters (α, β, n_p, n_c and number of patterns used during the learning) remains the same; only μ should be changed according to the previous discussion.

5. Conclusions

A multilayer, feedforward, fully connected ANN was designed and trained to simulate the performance of a three reactors network where low-pressure methanol synthesis is carried out. A complete and detailed model was used to generate the patterns for the learning procedure. The influence of the main parameters on the ANN was investigated and discussed, giving some general guidelines for their tuning. The HSA algorithm was preferred to the more conventional Single Pattern Presentation procedure, leading to faster convergence and better representation of the process. The ANN will be used in a future work to implement a MPC system for the process.

Notation

a_v	external particle surface area per unit volume of reactor, m^{-1}
b	bias
c	molar concentration, mol m^{-3}
$\hat{c}_P$	specific heat at constant pressure, J kg^{-1} K^{-1}
D_{eff}	effective mass dispersion coefficient, m^2 s^{-1}
d_P	pellet diameter, m
F_{in}	superficial inlet flow rate, mol m^{-2} s^{-1}
$\Delta\tilde{H}_f$	molar enthalpy of formation, J mol^{-1}
h	gas-solid heat transfer coefficient, J m^{-2} K^{-1} s^{-1}
K	adsorption equilibrium constant, bar^{-1}
K_p	chemical equilibrium constant based on partial pressure
k_{eff}	effective heat dispersion coefficient, J m^{-1} K^{-1} s^{-1}
k_G	gas-solid mass transfer coefficient, mol m^{-2} s^{-1}
k'_{ps}	reaction rate constant
L	total network length, m
l	single reactor length, m
N_R	number of reactions
n_c	number of repetitions of the whole stack
n_p	number of patterns in the stack
n_r	number of components in the mixture
p	pressure, bar
q	input of a neuron (with bias)
R	universal gas constant, J mol^{-1} K^{-1}
R'	reaction rate, mol s^{-1} kg^{-1}
s	output of a neuron
T	temperature, K
t	clock time, s
t_c	switching time, s
v	interstitial velocity, m s^{-1}
u	unbiased input signal to a neuron
x	axial reactor coordinate, m
y	molar fraction
w	synaptic weight

Greeks

α	learning rate
β	momentum constant
δ	error between the prediction of the ANN and the target value
Δ	variation
ε	void fraction of the catalytic bed
η	effectiveness factor
λ	thermal conductivity, J m^{-1} K^{-1} s^{-1}
μ	gain of the linear activation function
ν	stoichiometric coefficient
ρ	density, kg m^{-3}
ϕ	activation function of a neuron

Subscripts and superscripts

A	indicates CH_3OH from CO reaction
B	indicates CH_3OH from CO_2 reaction
C	indicates water-gas-shift reaction
G	gas phase
S	solid phase or solid surface
in	inlet condition
out	outlet value

References

Bhat, N., & McAvoy, T. J., (1990). Use of neural nets for dynamic modelling and control of chemical process systems, *Computers and Chemical Engineering*, *14*, 573-583.

Boreskov, G. K., & Matros, Y. S., (1983). Unsteady-state performance of heterogeneous catalytic reactions. *Catalysis Reviews — Science and Engineering*, *25*(4), 551-590.

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

Dufour, P., Couenne, F., & Touré, Y., (2003). Model predictive control of a catalytic reverse flow reactor. *IEEE Transaction on Control System Technology*, *11*(5), 705-714.

Gosiewski, K., Bartmann, U., Moszczynski, M., & Mleczko, L., (1999). Effect of the intraparticle mass transport limitation on temperature profiles and catalytic performance of the reverse-flow reactor for the partial oxidation of methane to synthesis gas. *Chemical Engineering Science*, *54*(20), 4589-4602.

Graaf, G. H., Sijtsema, P. J. J. M., Stamhuis, E. J., & Joosten, G. E., (1986). Chemical Equilibria in methanol synthesis. *Chemical Engineering Science*, *41*(11), 2883-2890.

Graaf, G. H., Stamhuis, E. J., & Beenackers, A. A. C. M., (1988). Kinetics of low-pressure methanol synthesis. *Chemical Engineering Science*, *43*(12), 3185-3195.

Haykin, S., (1999). *Neural networks – A comprehensive foundation*, Prentice Hall, London.

Hindmarsh, A. C., (1983). *ODEPACK, a systematized collection of ODE solvers*. Stepleman R. S. et al. Eds., Scientific Computing, Amsterdam.

Kolios, G., Frauhammer, J. & Eigenberger, G., (2000). Autothermal fixed-bed reactor concepts. *Chemical Engineering Science, 55*, 5945-5967.

Levin, A. U., & Narendra, K. S., (1995). Identification using feedforward networks. *Neural Computation*, *7*(2), 349-356.

Moré, J. J., Garbow, B. S. & Hillstrom, K. E., (1980). *User guide for MINPACK*; ANL-80-74, Argonne National Laboratory.

Psichogios, D. C., & Ungar, L. H., (1992). A hybrid neural network-first principle approach to process modelling, *AIChE Journal*, *38*, 1499-1511.

Tadé, M. O., Mills, P. M., & Zomaya, A. Y., (1996). *Neuro-adaptive process control: a practical approach*, John Wiley & Sons.

Vanden Bussche, K. M., & Froment, G. F., (1996). The STAR configuration for methanol synthesis in reversed flow reactors. *The Canadian Journal of Chemical Engineering*, *74*(5), 729-734.

Vanden Bussche, K. M., Neophytides, S. N., Zolotarskii, I. A., & Froment, G. F., (1993). Modelling and simulation of the reversed flow operation of a fixed-bed reactor for methanol synthesis. *Chemical Engineering Science*, *48*(19), 3335-3345.

Velardi, S. A., & Barresi, A. A., (2002). Methanol synthesis in a forced unsteady-state reactor network, *Chemical Engineering Science*, *57*, 2995-3004.

Velardi S. A., Barresi A. A., Manca D., & Fissore D., 2004. Complex Dynamic Behaviour of Methanol Synthesis in the Ring Reactor Network, *Chemical Engineering Journal, 99,* 117-123.

Zupan, J., & Gasteiger, J., (1999). *Neural networks in chemistry and drug design*, Wiley-VCH, Weinheim.

Zurada, J. M., (1992). *Introduction to artificial neural systems*. West Publishing Company.

Table 1 Conditions used in the simulations (base case).

Total length	L	0.5	m
Void fraction	ε	0.4	
Catalyst density	ϱ_S	1750	kg m^{-3}
Catalyst void fraction		0.5	
Pellet diameter	d_P	0.0054	m
Total pressure		5	MPa
Superficial inlet flow rate	F_{in}	32.65	mol m^{-2}s^{-1}
Feed composition			
CO		4.5	% mol
CO_2		2.0	% mol
CH_3OH		0.0	% mol
H_2O		0.0	% mol
H_2		93.5	% mol
Kinetic and equilibrium constants			
$k'_{\mathrm{ps,A}}$	$2.69\cdot10^{7}\cdot\exp[-109900/(RT)]$		
$k'_{\mathrm{ps,B}}$	$7.31\cdot10^{8}\cdot\exp[-123400/(RT)]$		
$k'_{\mathrm{ps,C}}$	$4.36\cdot10^{2}\cdot\exp[-65200/(RT)]$		
K_{CO}	$7.99\cdot10^{-7}\cdot\exp[58100/(RT)]$		
$K_{\mathrm{CO_2}}$	$1.02\cdot10^{-7}\cdot\exp[67400/(RT)]$		
$K_{\mathrm{H_2O}}/K_{\mathrm{H_2}}^{1/2}$	$4.13\cdot10^{-11}\cdot\exp[104500/(RT)]$		
$\log_{10} K_{p,\mathrm{A}}$	$5139/T-12.621$		
$\log_{10} K_{p,\mathrm{B}}$	$-2073/T-2.029$		
$\log_{10} K_{p,\mathrm{C}}$	$3066/T-14.650$		

List of figures

Figure 1

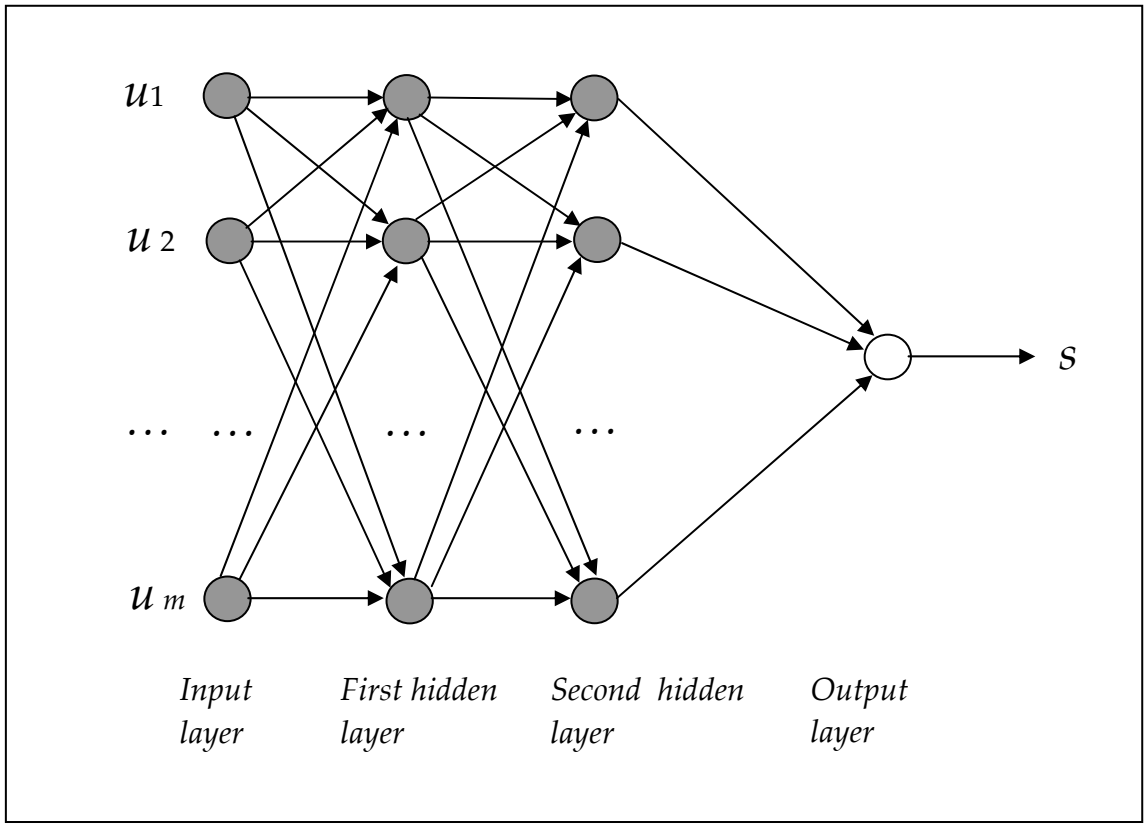

u_1
u_2
u_m
s
Input layer
First hidden layer
Second hidden layer
Output layer

Figure 2

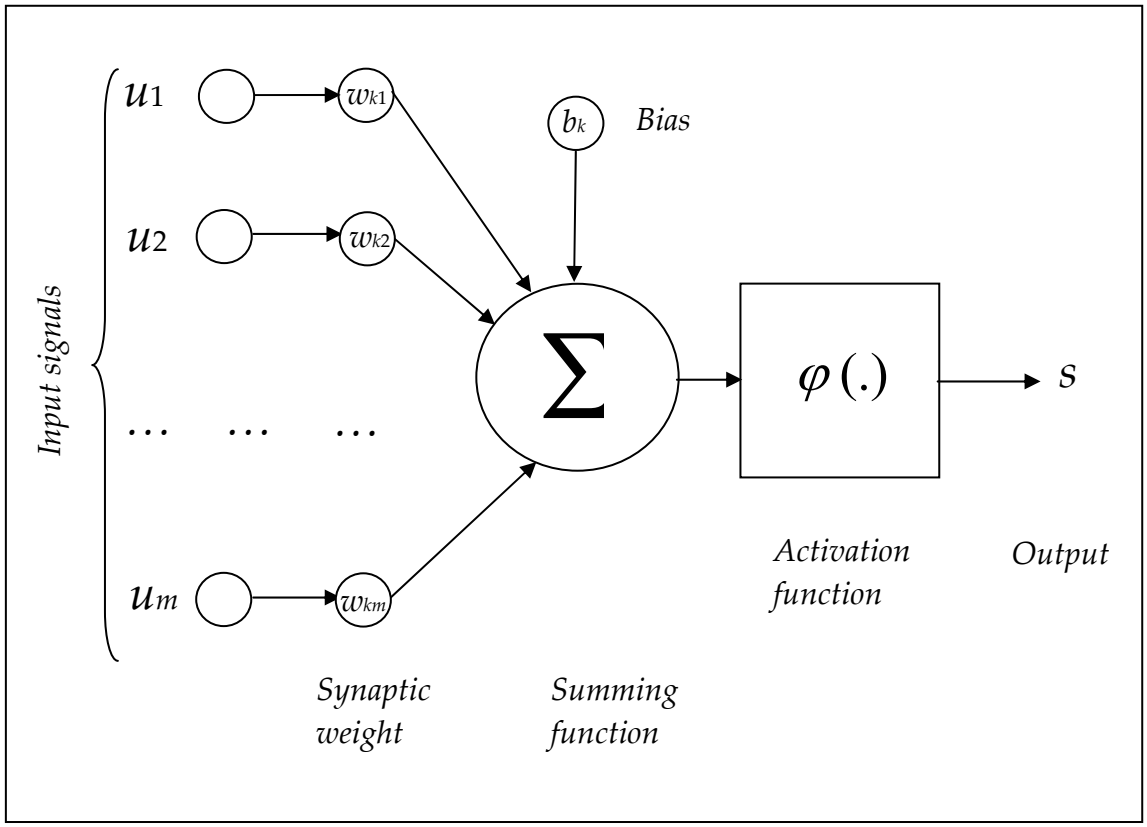

u_1
u_2
u_m
w_{k1}
w_{k2}
w_{km}
b_k
Bias
Input signals
$\sum$
$\varphi(.)$
s
Activation function
Output
Synaptic weight
Summing function

Figure 3

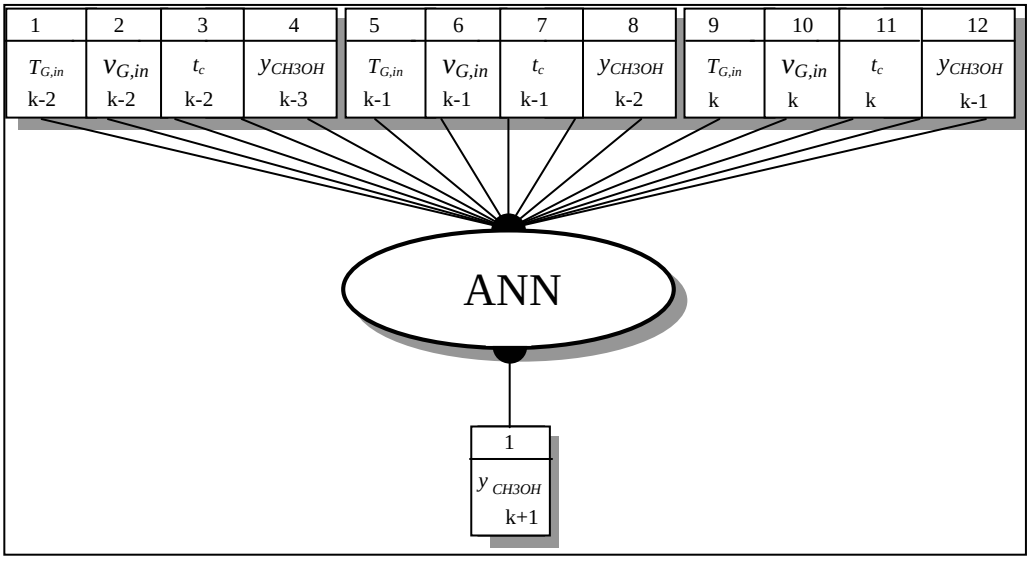

1
2
3
4
5
6
7
8
9
10
11
12
$T_{G,in}$ k-2
$v_{G,in}$ k-2
t_c k-2
y_{CH3OH} k-3
$T_{G,in}$ k-1
$v_{G,in}$ k-1
t_c k-1
y_{CH3OH} k-2
$T_{G,in}$ k
$v_{G,in}$ k
t_c k
y_{CH3OH} k-1
ANN
1
y_{CH3OH} k+1

Figure 4

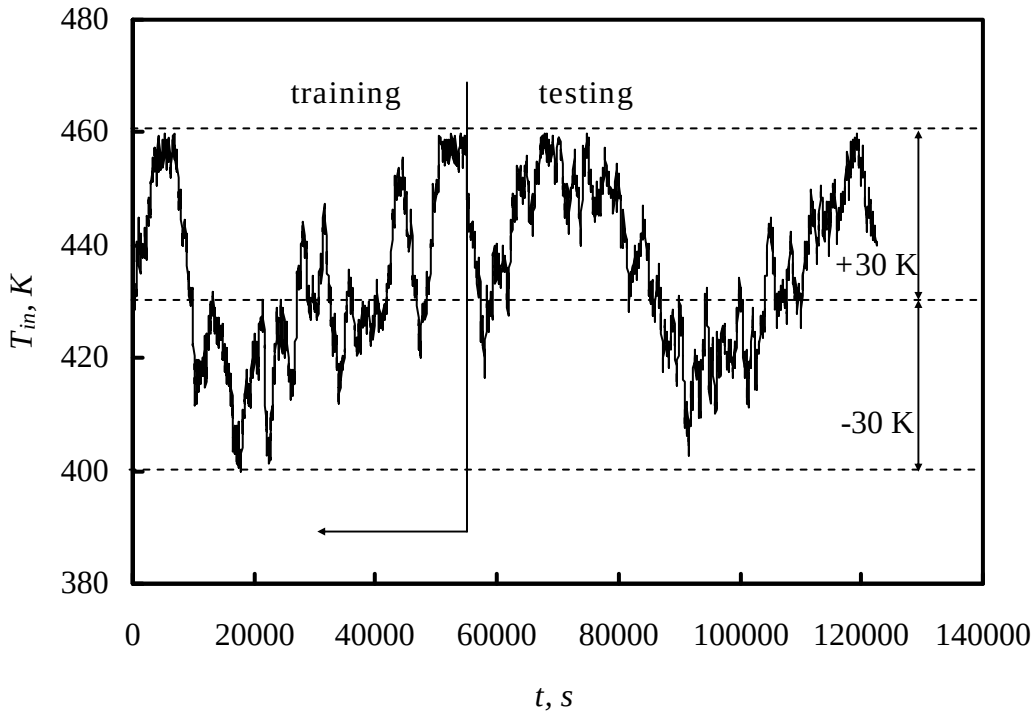

480
460
440
420
400
380
training
testing
+30 K
-30 K
T_{in}, K
0
20000
40000
60000
80000
100000
120000
140000
t, *s*

Figure 5

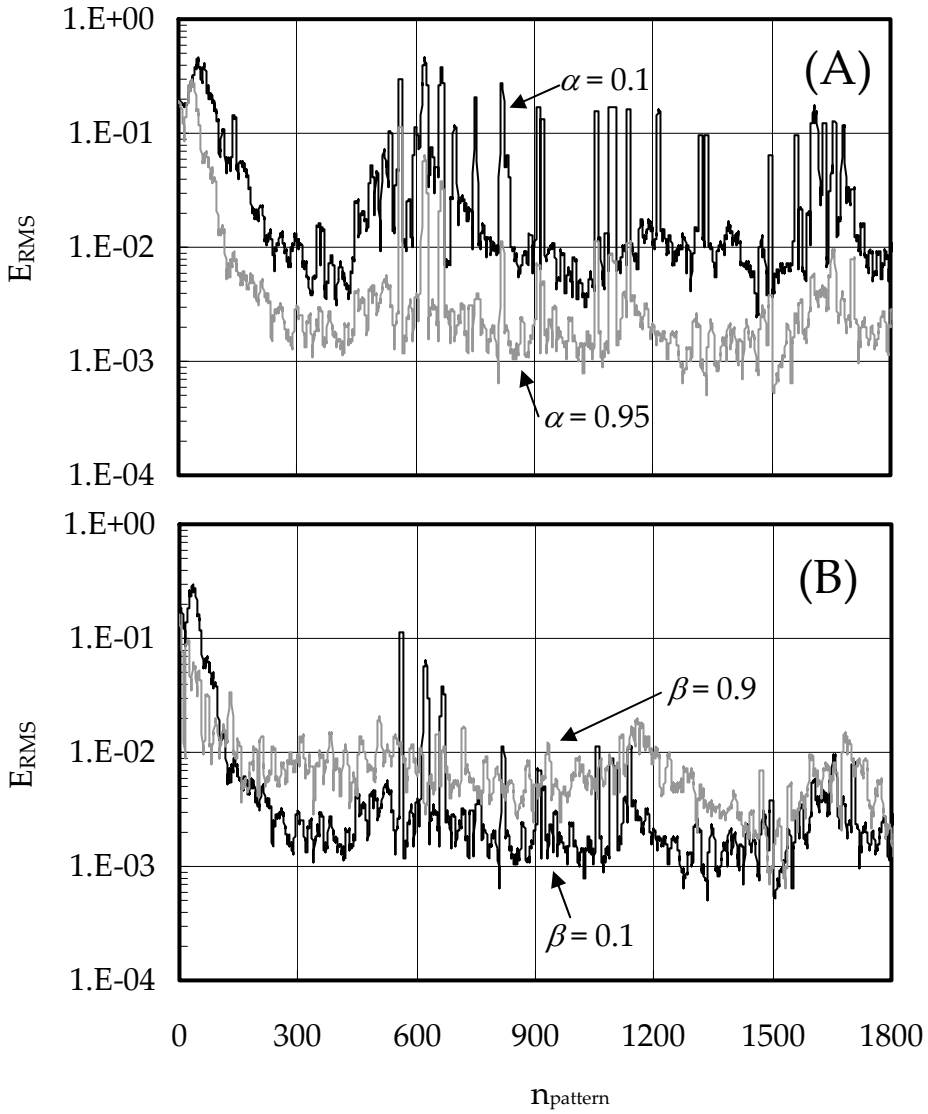

1.E+00
1.E-01
1.E-02
1.E-03
1.E-04
E_RMS
(A)
α = 0.1
α = 0.95
(B)
β = 0.9
β = 0.1
0
300
600
900
1200
1500
1800
n_pattern

Figure 6

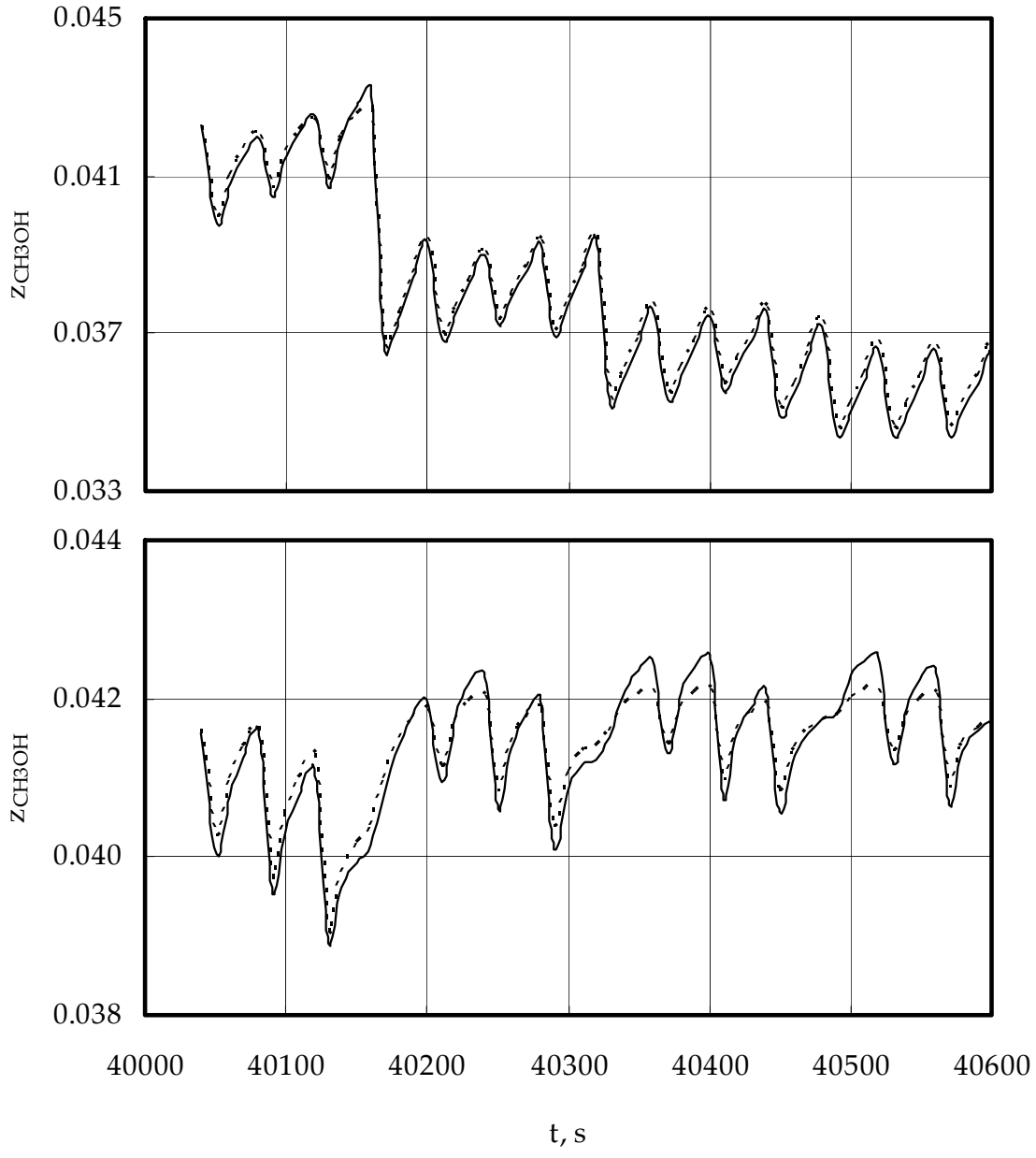

0.045
0.041
0.037
0.033
zCH3OH
0.044
0.042
0.040
0.038
zCH3OH
40000
40100
40200
40300
40400
40500
40600
t, s

Figure 7

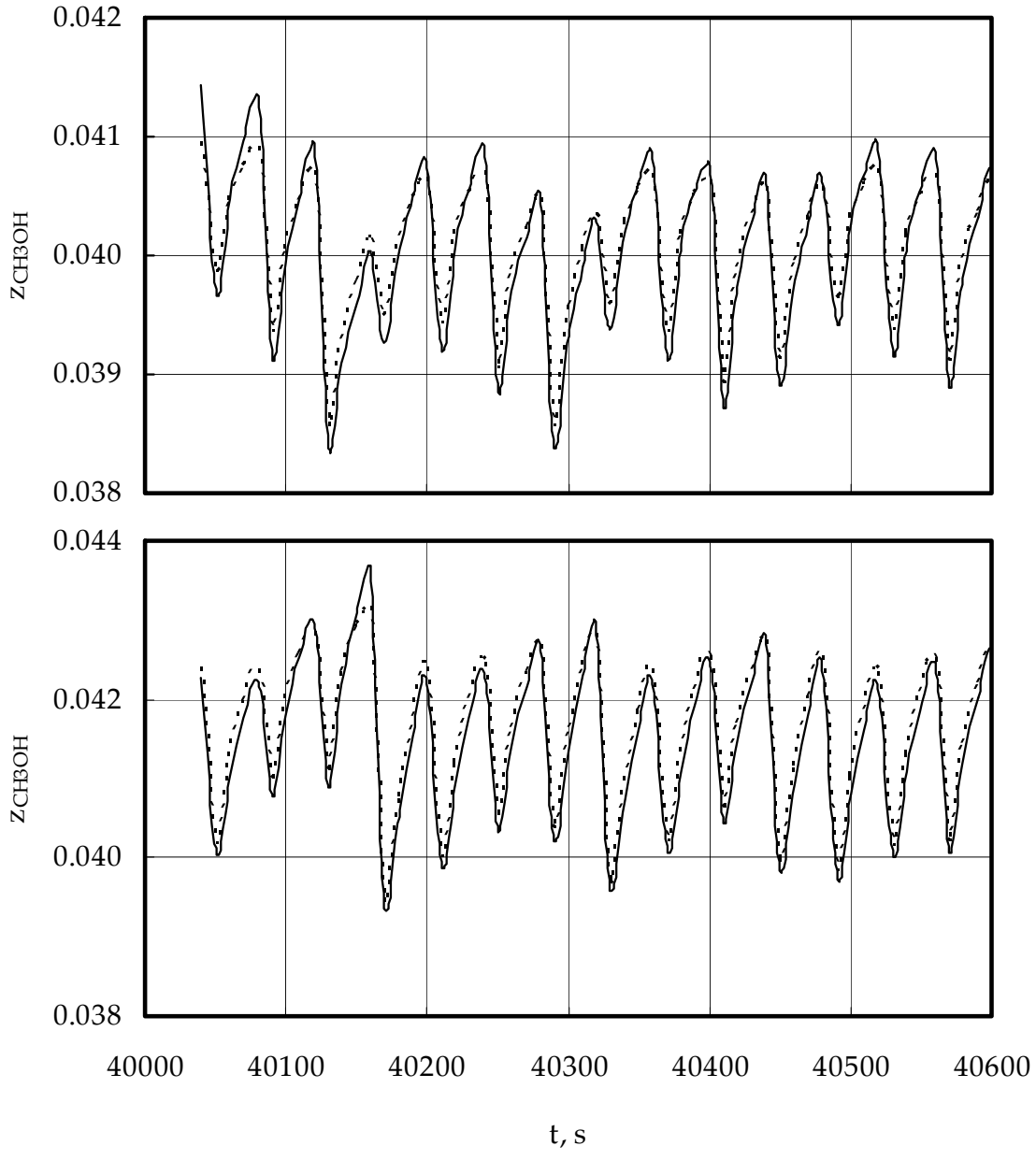
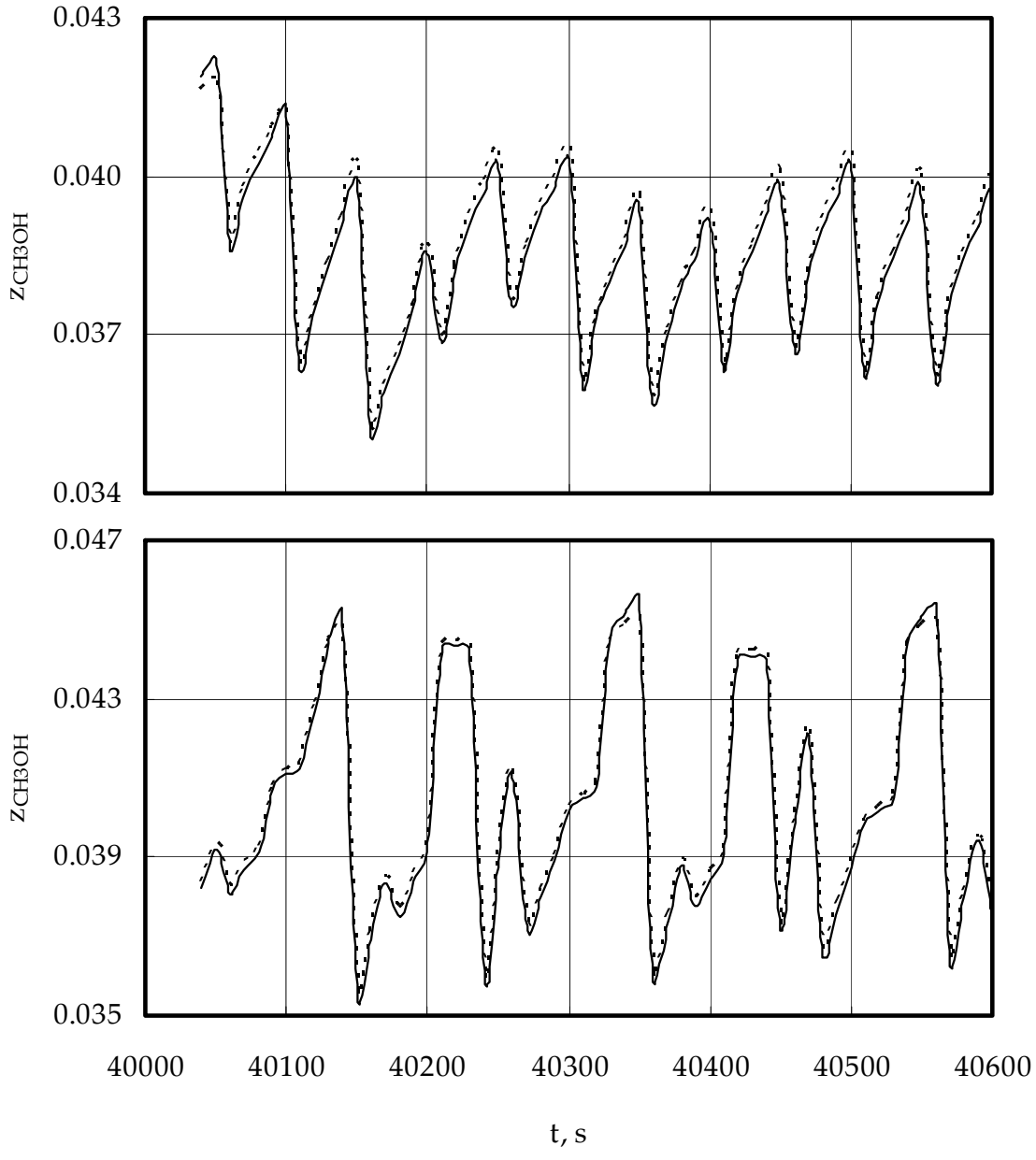

0.042
0.041
0.040
0.039
0.038
zCH3OH
0.044
0.042
0.040
0.038
zCH3OH
40000
40100
40200
40300
40400
40500
40600
t, s

Figure 8


0.043
0.040
0.037
0.034
zCH3OH
0.047
0.043
0.039
0.035
zCH3OH
40000
40100
40200
40300
40400
40500
40600
t, s