Different Methods of Neural Network Based Modelling for Polymerization Process

CIPRIAN-GEORGE PIULEAC, SILVIA CURTEANU*
"Gh. Asachi" Technical University, Department of Chemical Engineering, 71A Bd. D. Mangeron, 700050, Iasi, Romania

In this work, different types of neural networks and modeling methodologies are used and compared: feedforward and recurrent networks, stack neural networks and a hybrid model composed from a simplified phenomenological model and a neural network. For each situation, the performance of the networks was evaluated through mean squared error and correlation between training data and neural network predictions. Accurate results were obtained with different types of neural models, but our approach recommends feedforward neural networks which are simple to train and use. The well known free radical polymerization of methyl methacrylate, accompanied of gel and glass effects and achieved in a batch bulk process is considered as example. In a hybrid methodology, the kinetic model used until the onset of the gel effect is associated with a neural model which replaces the diffusional effects representing the difficult part to model in the process.

Keywords: neural networks, free radical polymerization modeling, polymethyl methacrylate, hybrid model.

The development of complete mechanistic models for polymerization reactors is not an easy task. The difficulties lie on the complex reactions occurring simultaneously inside the reactor, the large number of kinetic parameters, which are usually not easy to determine, as well as the poor understanding of chemical and physical phenomena for mixtures involving polymers.

To overcome the difficulties in the mechanistic modeling of polymerization processes, data based empirical models could be utilized. Neural networks possess the ability to learn what happens in the process without actually modeling the physical and chemical laws that govern the system. Therefore, they are useful for modeling complex nonlinear processes where understanding is limited.

The open literature presents many attempts concerning neural network applications for polymerization processes: direct modeling with different types of neural networks [1-4], neural networks based soft sensors [5], inferential modeling [6, 7], inverse neural network modeling [4, 8, 9], optimization [10-12], process control [13-15]. These types is of applications are reviewed in a precedent work [16].

Neural networks for nonlinear process modeling can be broadly divided into two categories: static networks, useful for steady-state modeling, and dynamic networks that are more appropriate for dynamic models. The first category includes the common multilayer feedforward neural networks in which information propagates only in one direction. The most important category of dynamic networks is represented by recurrent networks. In globally recurrent networks, the lagged network outputs are fed back to the network input nodes through time delay units. The network output depends not only on the network inputs, but also on the previous network outputs. Thus the predictions from a recurrent neural network are long range or multi-step ahead predictions. In locally recurrent networks, the output of a hidden neuron is fed back to its input through one or several time delay units. One important advantage of such networks, compared with fully recurrent networks, is that they have a smaller number of weights and can therefore be trained more efficiently.

Many researchers have shown that simply combining multiple neural networks can generate more accurate predictions than using any one of the individual networks alone. Such a combination of neural networks is known as a stacked neural network - where the final model prediction represents a combination of the prediction from the individual ones.

A method to improve the model generalization capability is given by a combination between a simplified mechanistic model with a neural network thus obtaining a hybrid model that may gather the best characteristics of both phenomenological and empirical approaches.

Several examples of using different types of neural networks for polymerization reactions are enumerated in the following. A model predictive control strategy based on a feedforward neural network model is proposed for an industrial polypropylene process [13]. In [17] it is described the application of feedforward neural networks and hybrid models to the finishing stage of nylon-6,6 polycondensation in a twin-screw extruder. In [18] is presented an example of how simple feedforward neural networks are able to help with the optimization of emulsion polymerization reactors, and more specifically, with the inverse problem, i.e., starting from the polymer properties and productivity to determine the reactor operating conditions. A one-hidden-layer artificial neural network using a back-propagation structure has been trained on experimental data for the identification of styrene conversion and polymer average molecular weight produced in a bulk polymerization initiated by bifunctional peroxides [19]. It is presented an algorithm consisting of a three-layer feedforward artificial neural network which uses supervised learning with reinforcement in a unique topology in order to improve and apply the temperature control of a free radical solution polymerization of styrene and to examine its performance on the basis of adaptive heuristic criticism control [20]. Since recurrent neural networks offer accurate long-range predictions, they are recommended in batch process optimal control where the ultimate interest lies on the final product quality; an example is the optimal control of a batch emulsion copolymerization reactor [11].

* *email: scurtean@ch.tuiasi.ro*

The contribution of Ng and Hussain [14] consists of the using an inverse neural network in a hybrid with a first principle model for the direct control of a nonlinear semi-batch polymerization process. These hybrid models were utilized in the control strategy to track the set point of the temperature of the polymerization reactor under nominal conditions and with various disturbances. To address the difficulties in batch polymerization reactor modeling and optimal control, there were used recurrent neural networks and hybrid models containing neural network-based discrete-time models [21, 22, 23].Bootstrap aggregated neural networks are used to model a batch polymerization reactor from limited batches of process operational data [7, 12]. This is a novel technique for building robust non-linear models in which several different pairs of training and testing data sets are formed and for each pair, a neural network model is developed. One of the paper presents a technique for developing neural-network-based process models using both dynamic and static process operating data and feedforward neural networks [24]. Long-term prediction models based on mixed order locally recurrent neural networks in which different hidden neurons having different numbers of feedbacks are applied to a continuous polymerization process [1].

Our work deals with neural networks based modeling of the free radical polymerization of methyl methacrylate (MMA). Different types of neural networks and neural methodologies are used and compared: feedforward and recurrent networks, stacks neural networks and a hybrid model composed from a phenomenological model which does not take into account the diffusional effects and a neural network which replaces the gel and glass effects. For each situation, the networks performance is evaluated from two points of view: the accuracy of the results and the difficulty in handling the network. As a main conclusion, our approach demonstrates the possibility of accurately model a free radical polymerization process with feedforward networks having simple topologies - 1 or 2 intermediate layers.

This paper brings as novelty the application of different modeling methods that are based on neural networks (as network types and possibilities of model construction), as well as the comparison between results and performances of the models in order to select the optimal variant. In contrast with many other works in the literature, which recommend the use of stacked recurrent networks (which are complicated entities) for dynamic process modeling, the present approach points out the efficient use of some simple neural networks – feedforward networks with two hidden layers. A special proposal is also the hybrid model considered, based on a simple neural network, whose main advantage is to avoid quantifying the gel and glass effects that are usually met in free radical polymerization. The results is that such neural network models are proved to devise with accuracy the actual behavior of the investigated polymerization system.

Neural networks

Neural networks perform computation in a very different way than conventional computers, where a single central processing unit sequentially dictates every piece of action. Neural networks are built from a large number of simple processing elements (neurons) that individually deal with parts of a big problem. A processing element simply multiplies an input by a set of weights, and nonlinearly transforms the result into an output value. The power of neural computation comes from the massive *interconnection* among the processing elements, which share the load of the overall processing task, and from the adaptive nature of the parameters (weights). The way in which neurons are connected to form a network represents the neural network topology (architecture). More precisely, the topology of a neural network consists of the framework of neurons together with its interconnection structure. The neural network topology plays a fundamental role in its functionality and performance [25].

The form of the interconnection provides one of the key variables for dividing neural networks into families. The most general case is the fully connected neural network. By definition, any processing element can feed or receive activations of any other including itself. Therefore, when the weights are represented in matrix form (the weight matrix), it will be fully populated. A 6 processing elements fully connected network is presented in the figure 1 [25]. This network is called a recurrent network and $w_{11}...w_{66}$ are the weights of the network. In recurrent networks some of the connections may be absent, but there are feedback connections. An input presented to a recurrent network at time t will affect the networks output for future time steps greater than t. Therefore, recurrent networks need to be operated over time. Dynamic networks are important tools for a number of important engineering applications which require the processing of time-varying information. To cope with time varying signals, neural network topologies have to be enhanced with short term memory mechanisms. There is probably the area where neural networks will provide an undisputed advantage, since other technologies are far from satisfactory.

If the interconnection matrix is restricted to feedforwarding activations (no feedback nor self connections), the neural network is defined as feedforward. Feedforward networks are instantaneous mappers; i.e. the output is valid immediately after the presentation of an input. A special class of feedforward networks is the layered class, which is called as multilayer perceptron (MLP). Multilayer perceptrons have processing elements arranged in layers. The layers without direct access to the external world, i.e. connected to the input or output, are called hidden layers (processing elements 4, 5 in the fig. 2). Layers that receive the input from the external world are called the input layers (processing elements 1, 2, 3 in the fig. 2); layers in contact with the outside world are

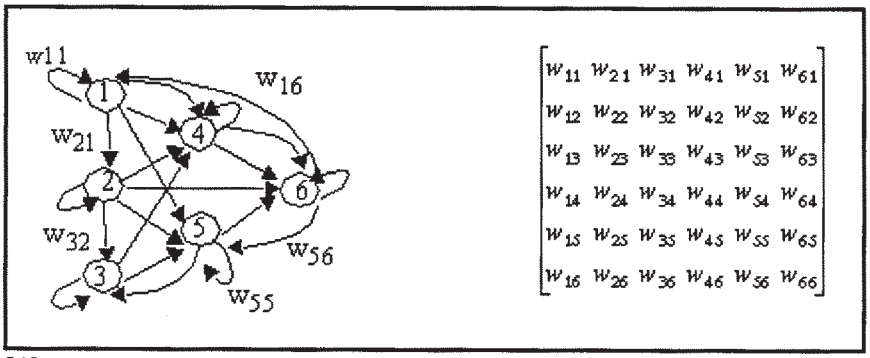

Fig. 1. A fully connected neural network and the weight matrix.

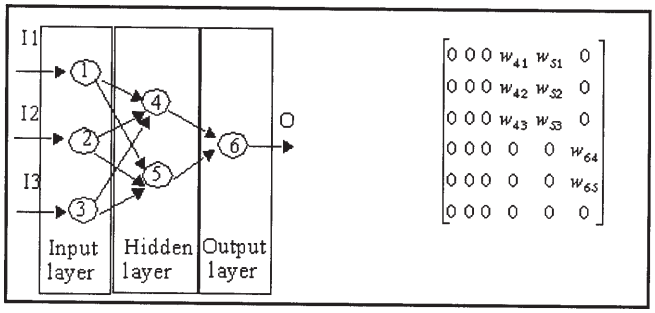

Fig. 2. A multilayer perceptron and its weight matrix.

called output layers (processing element 6 in the figure 2 [25]).

Notice that most entries in the weight matrix of an MLP are zero. In particular, any feedforward network has at least the main diagonal, and the elements below populated with zeros. Feedforward neural networks are therefore a special case of recurrent networks.

Generalized feed-forward networks (GFN) are a generalization of the MLP such that connections can jump over one or more layers (fig. 3 [25]). Generalized feedforward networks often solve the problem much more efficiently than MLP. A standard MLP requires hundreds of times more training epochs than the generalized feedforward network containing the same number of processing elements.

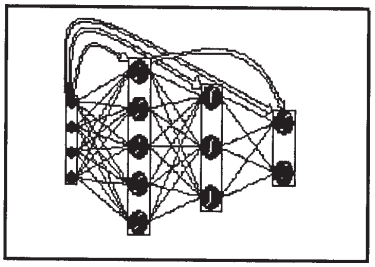

Fig. 3. A generalized feedforward network

Modular feedforward networks are a special class of MLP. These networks process their input using several parallel MLPs, and then recombine the results. This tends to create some structure within the topology, which will foster specialization of function in each sub-module. In contrast to the MLP, modular networks do not have full interconnectivity between their layers. Therefore, a smaller number of weights are required for the same size network (*i.e.* the same number of processing elements). This tends to speed up training times and reduce the number of required training exemplars. There are many ways to segment a MLP into modules. Several examples are presented in figure 4 [25].

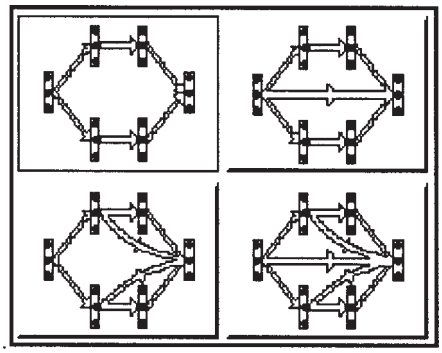

Fig. 4. Modular feedforward networks, with different configurations

In the training phase, the neural network learns the behaviour of the process. The training data set contains both input patterns and the corresponding output patterns (also called target patterns). Neural training leads to finding values of connection weights that minimize differences between the network outputs and the target values. The most extensively adopted algorithm for the learning phase is the back-propagation algorithm. By adapting its weights, the neural network works towards an optimal solution based on a measurement of its performance. The training phase is considered complete when the error of all the training patterns is less than a prespecified error criterion or a maximum number of epochs had been reached. There are mainly three practical aspects related to learning. The first one is the choice of the training set and its size. The second is the selection of learning constants, and the third is when to stop the learning. Unfortunately, there are no "formulas" to select these parameters. In this reason, only some general rules apply and a lot of experimentation is necessary.

The purpose of developing a neural model is to devise a network (set of formulae) that captures the essential relationships into the data. These formulae are then applied to new sets of inputs to produce corresponding outputs. This is called generalization and represents subsequent phase after training - validation or testing phase. Since a neural network is a nonlinear optimization process made up of a learning phase and a testing phase, the initial data set must be split into two subsets: one for training and one for testing. A learning algorithm should lead to a good fit to the training samples and, simultaneously, to a network that has a good generalization capability. A network is said to generalize well when the input-output relationship, found by the network, is correct for input/output patterns of validation data which were never used in training the network (unseen data).

Experimental part

To model monomer conversion, number average molecular weight and weight average molecular weight in free radical polymerization of MMA, neural networks with different topologies are built. Simulation data of different temperature and initiator concentrations were used in building and training neural networks. Apparently, unsuitable for the use of simulation data, many authors working with neural networks are utilizing simulation data to illustrate their modeling methodologies. We apply this technique in the present paper too. The obtained simulation data, in a suitable quantity, is useful to the comparison of the performance of various types of neural models (network types and modeling techniques based on neural networks).

The reaction operation conditions for batch bulk polymerization of methyl methacrylate chosen for neural network training were: $50 \leq T \leq 90$°C (temperature) with a step of 5°C in data collection through simulation, $10\ I_0 \leq 50$ mol/m^3 (initial concentration of initiator) with step of 5 mol/m^3 and $0 \leq t \leq 500$ min , step 1 min (time). Our kinetic model developed in a previous paper [26] was used to produce simulation data. A large amount of training data results in this way, in order to exceed at least 3-4 times the number of weights for neural networks. The inputs of the networks were the reaction conditions, T, I_0, t and the outputs were monomer conversion and number and weight average polymerization degrees (x, DP_n, DP_w). The number of hidden layers and units were established by training a different range of networks and selecting the one that best balanced generalization performance against network size.

Results and discussion

Table 1 presents several examples for the trained neural networks. The column Topology contains the number of neurons in input, hidden and output layers, MSE is the mean squared error and **r** represents the correlation between training data and the answers of the networks. For instance, MLP(3:42:14:3) indicates 3 neurons in input layer, 42 neurons for the first hidden layer, 14 neurons for the second hidden layer and 3 neurons for the three outputs variables.

The mean squared error was computed using the following formula:

$$MSE = \frac{1}{M \cdot L} \sum_{k=1}^{M} \sum_{p=1}^{L} \left(d_k^p - y_k^p \right)^2 \qquad (1)$$

where M is the number of nodes in output layer, L is the number of exemplars in the data set (number of patterns), d_k^p is the desired output for exemplar p at processing element k, and y_k^p is the network output for exemplar p at processing element k.

Table 1
DIFFERENT MLP TOPOLOGIES TRAINED FOR MMA POLYMERIZATION

No.	Topology	MSE	r
1	MLP(3:50:3)	0.002003	0.9909
2	MLP(3:100:3)	0.002192	0.99031
3	MLP(3:12:4:3)	0.000820	0.9953
4	MLP(3:24:8:3)	0.000292	0.9983
5	**MLP(3:42:14:3)**	**0.000085**	**0.9996**

The best topology, corresponding to the best performance, was identified as MLP(3:42:14:3). The good agreement between the simulation data and the results of neural network in the training phase proves that the network learned well the behaviour of the process. Some examples are given in figures 5, 6 and 7, for the three parameters of interest: monomer conversion and number and weight average polymerization degrees (DP_n and DP_w, respectively).

For the validation phase, new data have been generated by simulation; it is more important to evaluate the performance of the neural networks on unseen data that training data. In this way, we can appreciate the most important feature of a neural model - the generalization capability. Table 2 shows good agreement between simulation data (not used in the training phase) and neural network predictions, highlighting the capability of neural

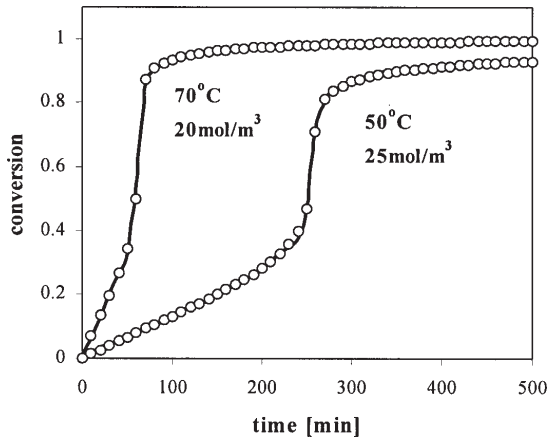

Fig. 5. The monomer conversion obtained from phenomenological model (circles) and as prediction of MLP (3:42:14:3) (continuous lines) on training data at different reaction conditions (T = 50 and 70°C, I_0 = 20 and 25 mol/m^3)

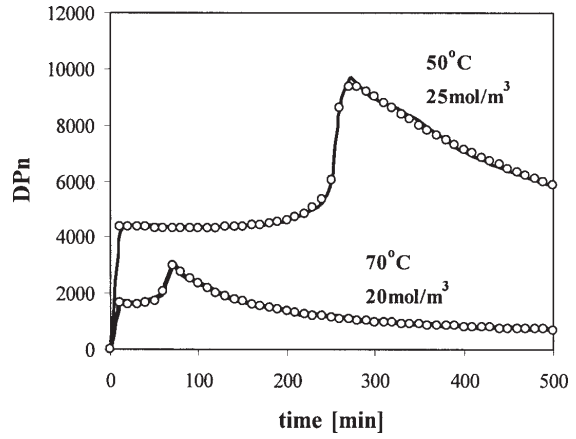

Fig. 6. The number average polymerization degree obtained from phenomenological model (circles) and as predictions of MLP (3:42:14:3) (continuous lines) on training data at different reaction conditions (T = 50 and 70°C, I_0 = 20 and 25 mol/m^3)

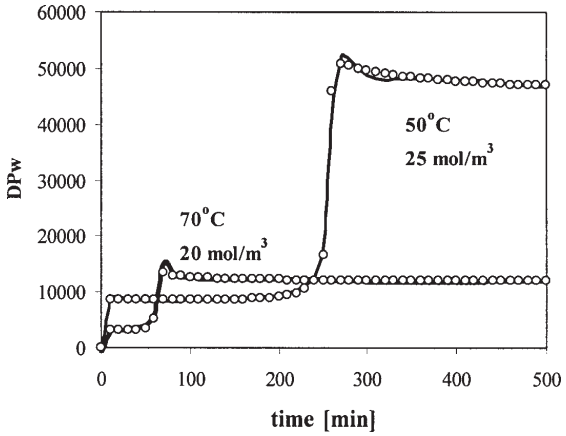

Fig. 7. The weight average polymerization degree obtained from phenomenological model (circles) and as predictions of MLP (3:42:14:3) (continuous lines) on training data at different reaction conditions (T = 50 and 70°C, I_0 = 20 and 25 mol/m^3)

network to memorize the nonlinear behaviour of complex processes. In table 2, the relative errors were calculated as:

$$error = \frac{\left| (p_{model} - p_{net}) \right|}{p_{model}} \cdot 100 \qquad (2)$$

where $p = x, DP_n$ and DP_w for the columns x error, DP_n error and DP_w error. The notations used in table 2 are: x, DP_n, DP_w for phenomenological model and x_net, DP_n_net, DP_w_net for neural model. Averaging these errors, the following results are obtained: 0.4436 % for monomer conversion, 2.6402 % for DP_n and 2.6078 % for DP_w. Also, the correlations for the two data types (model results and neural networks predictions) were: 0.99927 for monomer conversion, 0.99991 for DP_n and 0.99974 for DP_w.

Table 3 contains some GFNs (Generalized Feedforward Networks) trained for MMA polymerization, with reaction conditions as inputs (T, I_0 and t) and x, DP_n and DP_w as outputs. The best network, with smallest MSE and high value for r (closed to 1) was GFN(3:42:14:3), having the same topology with the best MLP. Figures 8, 9 and 10 contains examples for GFN (3:42:14:3) predictions.

Stack (modular) neural networks were also designed and tested. Table 4 shows the performances of four neural networks with the topologies from figure 4 and the same dimensionality (3 input neurons, 42 neurons in the first

Table 2
DATA OF THE PHENOMENOLOGICAL MODEL AND PREDICTIONS OF MLP(3:42:14:3) OBTAINED IN THE VALIDATION PHASE AT DIFFERENT REACTION CONDITIONS

T [°C]	I_0 [mol/m^3]	t [min]	x	DP_n	DP_w	x_net	DP_n_net	DP_w_net	x error	DP_n error	DP_w error
60	10	300	0.95	5827	36678	0.94	5776	36504	0.3155	0.8742	0.4737
70	10	210	0.97	2501	18184	0.98	2461	17742	0.4425	1.5854	2.4312
80	10	320	1.00	929	9210	1.00	892	9225	0.0611	3.9833	0.1624
50	15	380	0.89	11826	63851	0.89	11938	62617	0.0887	0.9439	1.9330
60	15	280	0.95	4259	29906	0.95	4227	30053	0.2844	0.7426	0.4921
70	15	190	0.97	1851	14581	0.97	1844	14522	0.5077	0.3866	0.4044
80	15	430	1.00	586	7216	1.00	601	7302	0.0193	2.7316	1.1908
50	20	380	0.90	9158	54597	0.90	9160	54090	0.1928	0.0211	0.9275
60	20	240	0.94	3714	25641	0.95	3685	25844	0.6909	0.7958	0.7919
70	20	260	0.98	1112	12107	0.98	1152	12122	0.0848	3.5349	0.1275
80	20	350	1.00	465	5980	1.00	458	6076	0.0051	1.4075	1.6139
50	25	320	0.88	8589	49084	0.88	8668	48061	0.2519	0.9138	2.0847
60	25	240	0.94	3015	22378	0.95	2948	22332	0.5161	2.2115	0.2052
80	25	390	1.00	364	5118	1.00	363	5189	0.0015	0.2436	1.3852
90	25	380	1.00	325	2653	1.00	322	2623	0.0982	0.9002	1.1263
50	30	380	0.91	6306	42994	0.91	6272	43111	0.0923	0.5323	0.2715
60	30	400	0.97	1633	19447	0.97	1611	19266	0.0119	1.3888	0.9332
70	30	240	0.98	800	9265	0.98	826	9418	0.1324	3.2041	1.6521
80	30	370	1.00	308	4486	1.00	299	4562	0.0042	2.9983	1.6946
50	35	440	0.93	4792	38582	0.93	4828	38641	0.0244	0.7498	0.1525
70	35	320	0.99	557	8247	0.99	570	8152	0.0076	2.2986	1.1597
50	40	340	0.91	5306	36136	0.91	5301	36285	0.0935	0.0943	0.4145
60	40	260	0.95	1801	16308	0.95	1770	16258	0.1283	1.7137	0.3113
70	40	430	0.99	400	7441	0.99	409	7488	0.0808	2.2995	0.6340
50	45	170	0.31	3360	6733	0.32	3328	6701	3.1182	0.9525	0.4622
60	45	110	0.86	3300	16321	0.87	3300	17261	0.8068	0.0071	5.7585
70	45	290	0.99	468	6868	0.99	482	6826	0.0297	2.7980	0.6162
50	50	310	0.90	4663	31319	0.90	4659	31222	0.1311	0.0798	0.3091
60	50	460	0.97	886	13600	0.97	878	13959	0.0133	0.8132	2.6391

Table 3
DIFFERENT GFNS TOPOLOGIES TRAINED FOR MMA POLYMERIZATION

No.	Topology	MSE	r
1	**GFN(3:42:14:3)**	**0.000634**	**0.9971**
2	GFN(3:100:3)	0.003884	0.9826
3	GFN(3:90:30:3)	0.00497	0.9975

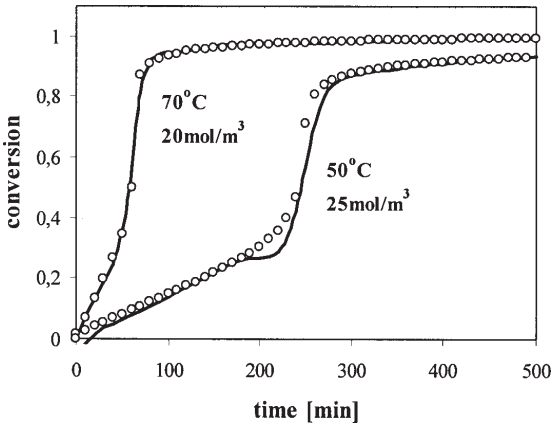

Fig. 8. Monomer conversion obtained from phenomenological model (circles) and as predictions of GFN (3:42:14:3) (continuous lines) at different reaction conditions (T = 50 and 70°C, I_0 = 20 and 25 mol/m^3)

hidden layer, 14 neurons in the second hidden layer and 3 outputs neurons). Accurate predictions were obtained with the network corresponding to the connection 2 in figure 4.

Figure 11 shows an example, the modular network 2 in table 4 being used at T = 70°C and I_0 = 25 mol/m^3.

Recurrent neural networks are also tested for polymerization reaction, especially because this type of topology is recommended for modeling dynamic process. Accurate results are obtained, as well as with feedforward neural networks. Figures 12 and 13 present variation in time of monomer conversion and average polymerization degrees at T = 50°C and I_0 = 25 mol/m^3.

Kipping in mind that recurrent networks are more difficult to design and train, we can conclude that feedforward neural networks represent an acceptable alternative of modeling polymerization processes.

A special software application - *NeuroSolutions* - was used in this paper in order to design and obtain predictions of neural networks. In this program, the following specifications are necessary: the network type, the input and desired output values, the stop condition of the training, the number of processing elements in hidden layers, the activation functions, the learning rule, the maximum number of epochs and some configuration parameters to display the neural model development. We built and trained many networks, changing the above options, and then we selected the best one that balances the size and the performance.

Another problem focused in this paper refers to the use of different activation functions in the networks and different learning rules in the training phase. By activation, we have described the way in which information, or data, flows through the network. Generally, each neuron receives weighted input from every other neuron in the network, applies a non-linear threshold and presents its output for the others to input (fig. 14) [25].

Neural networks are designed by first defining the neuron interconnections, and then assigning a learning procedure to adapt its weights.

In Table 5 are presented the performances of the MLP (3:42:14:3) containing different activation functions (hyperbolic tangent, sigmoid, linear hyperbolic tangent,

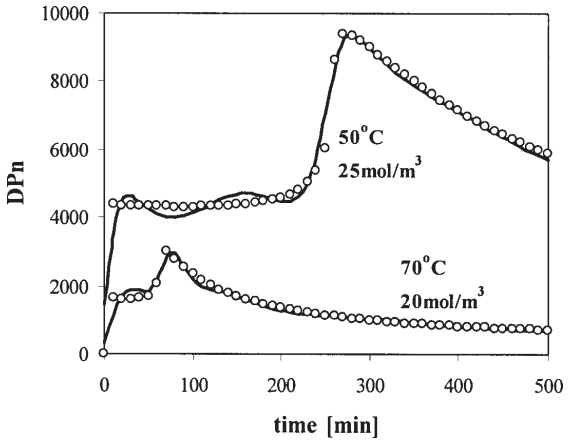

Fig. 9. Number average polymerization degree obtained from phenomenological model (circles) and as predictions of GFN (3:42:14:3) (continuous lines) at different reaction conditions (T = 50 and 70°C, I_0 = 25 and 20 mol/m^3)

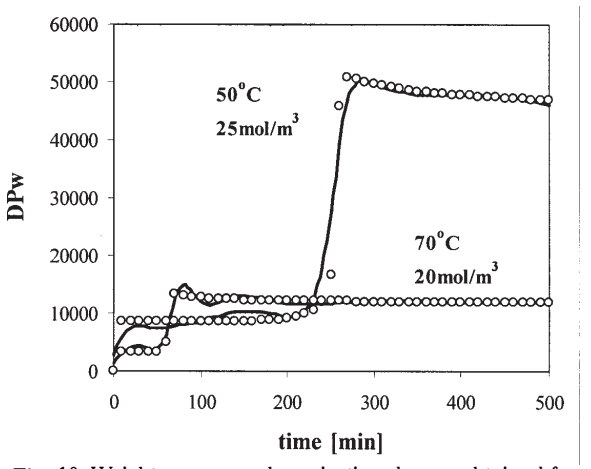

Fig. 10. Weight average polymerization degree obtained from phenomenological model (circles) and as predictions of GFN (3:42:14:3) (continuous line) at different reaction conditions (T = 50 and 70°C, I_0 = 25 and 20 mol/m^3)

No.	Topology	MSE	r
1	$\left(3:\begin{smallmatrix}21\\21\end{smallmatrix}:\begin{smallmatrix}7\\7\end{smallmatrix}:3\right)$ connection 1 in figure 4	0.004603	0.9799
2	$\left(\mathbf{3:}\begin{smallmatrix}\mathbf{21}\\\mathbf{21}\end{smallmatrix}\mathbf{:}\begin{smallmatrix}\mathbf{7}\\\mathbf{7}\end{smallmatrix}\mathbf{:3}\right)$ **connection 2 in figure 4**	**0.00013**	**0.9996**
3	$\left(3:\begin{smallmatrix}21\\21\end{smallmatrix}:\begin{smallmatrix}7\\7\end{smallmatrix}:3\right)$ connection 3 in figure 4	0.001796	0.9916
4	$\left(3:\begin{smallmatrix}21\\21\end{smallmatrix}:\begin{smallmatrix}7\\7\end{smallmatrix}:3\right)$ connection 4 in figure 4	0.00023	0.9989

Table 4
DIFFERENT MODULAR NEURAL NETWORKS TRAINED FOR MMA POLYMERIZATION

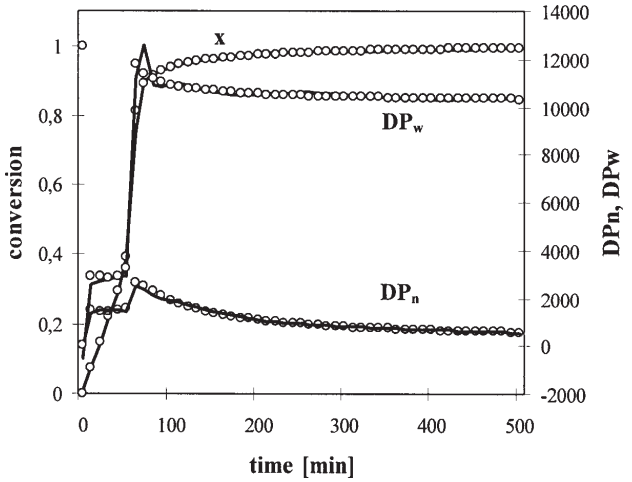

Fig. 11. Monomer conversion and average polymerization degrees obtained from phenomenological model (circles) and as predictions of modular network (continuous lines) at T = 70°C and I_0 = 25 mol/m^3

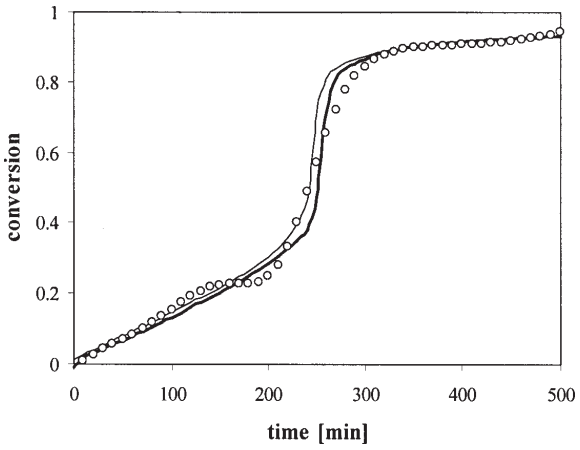

Fig. 12. Monomer conversion obtained from phenomenological model (circles) and as predictions of feedforward neural network (thin line) and recurrent network (thick line) at T = 50°C and I_0 = 25 mol/m^3

linear sigmoid, softmax) and trained with different algorithms (momentum, learning step, conjugate gradient, momentum and adaptive learning rate (deltabardelta), quick propagation), available in NeuroSolutions.

The best performance corresponds to hyperbolic tangent as activation function for hidden layers and deltabardelta as learning algorithm.

A method to improve the model generalization capability is represented by a combination between a simplified mechanistic model with a neural network, thus obtaining a hybrid model that may gather the best characteristics of both phenomenological and empirical approaches. For MMA polymerization, at low conversions, the kinetic model is well known, but, at high conversions, the gel and glass phenomena appear, representing a difficult part to model. Therefore, we propose the following hybridization procedure: until critical conversion, at which the gel effect appears, the simplified phenomenological model is valid [26]. After this conversion, modeling is performed with a MLP (4:42:14:3), which has temperature, initiator concentration, time and initial conversion as inputs and final conversion and average polymerization degrees as outputs. It is necessary to introduce conversion both as an input (initial value) and as an output parameter (final value

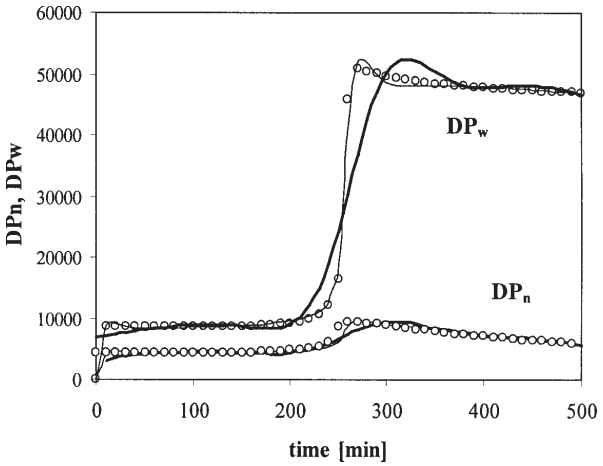

Fig. 13. Average polymerization degrees obtained from phenomenological model (circles) and as predictions of feedforward neural network (thin line) and recurrent network (thick line) at T = 50°C and I_0 = 25 mol/m^3

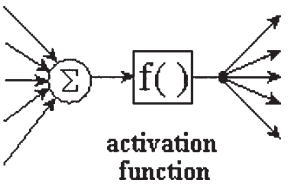

Fig. 14. The mapping of a neuron

No.	Learning rule	Activation function	MSE	r
1	momentum	tanh	0.001216	0.9986
2	step	tanh	0.002539	0.9875
3	conjugate gradient	tanh	0.002806	0.9865
4	**deltabardelta**	**tanh**	**0.000085**	**0.9996**
5	quickprop	tanh	0.002704	0.9875
6	deltabardelta	sigmoid	0.00070	0.9986
7	deltabardelta	lineartanh	0.001006	0.9946
8	deltabardelta	linearsigmoid	0.000645	0.9853
9	deltabardelta	softmax	0.04338	0.9875

Table 5
PERFORMANCE OF MLP(3:42:14:3) FOR DIFFERENT LEARNING RULE AND ACTIVATION FUNCTIONS

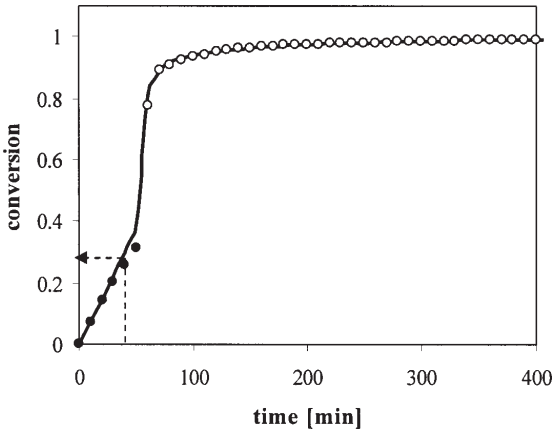

Fig. 15. Monomer conversion obtained at T = 70°C and I_0 = 25 mol/m^3 with a hybrid model composed of a simplified phenomenological model (•) and a MLP (4:42:14:3) network (o), successively used until x_{crit} = 0.24; continuous line – complete phenomenological model

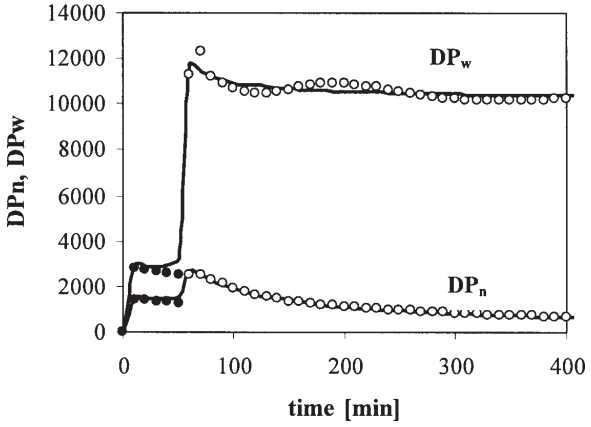

Fig. 16. Number and weight average polymerization degrees obtained at T = 70°C and I_0 = 25 mol/m^3 with a hybrid model composed of a simplified phenomenological model (•) and a MLP(4:42:14:3) network (o), successively used until x_{crit} ; continuous line – complete phenomenological model

corresponding to the current time) because the neural model is not used from zero conversion, but from an initial conversion equal to the critical conversion. Figures 15 and 16 present an example for this procedure applied at T = 70°C and I_0 = 25 mol/m^3. The way for determining the critical conversion is also pointed out in the figure 15 – the conversion at which the real conversion differs from that of the simplified kinetic model (model without gel and glass effects).

Conclusions

Performances of different types of neural networks (feedforward, recurrent and stacks), containing different activation functions and trained with different algorithms were evaluated for the free radical polymerization of methyl methacrylate. As a main conclusion, a feedforward neural network with two hidden layers, trained with adaptive learning rule and using hyperbolic tangent as activation function for hidden neurons provides accurate results for polymerization process under study. A hybrid model composed from a simplified phenomenological model and a neural network, used successively before and after critical conversion, is also recommended for accurately render the real process.

Acknowledgement: This work was done by financial support provided through Contract PC 71-006/2007, in the frame of the National Program for Research, Development and Innovation II.

References

1.ZHANG, J., MORRIS, A.J., MARTIN, E.B., Long-term prediction models based on mixed order locally recurrent neural networks, Comput. Chem. Eng., **22**, 1998, p.1051

2.KURODA, C., KIM, J., Neural network modeling of temperature behavior in an exothermic polymerization process, Neurocomputing, **43**, 2002, p.77

3.CURTEANU, S., LEON, F., GÂLEA, D., Neural network models for free radical polymerization of methyl methacrylate, Eurasian Chemico-Technological Journal, **5** , 2003, p. 225

4.CURTEANU, S., Direct and inverse neural network modeling in free radical polymerization, Central European Journal of Chemistry**, 2**, 2004, p. 113

5.ARAUJO, P.H.H., SAYER, C., DE LA CAL, J.C., ASUA, J.M., LIMA, E.L., PINTO, J.C., Utilization of neural networks as soft sensors to monitor emulsion polymerization reactions (average particle diameter and conversion), II ENPROMER, Florianópolis, Brazil, 1999

6.ZHANG, J., MARTIN, E.B., MORRIS, A.J., KIPARISSIDES, C., Inferential estimation of polymer quality using stacked neural networks, Comput. Chem. Eng., **21**, 1997, p. 1025

7.ZHANG, J., Developing robust non-linear models through bootstrap aggregated neural networks, Neurocomputing, **25**, 1999, p. 93

8.FERNANDES, F.A.N., LONA, L.M.F., Application of neural networks for the definition of the operating conditions of fluidized bed polymerization reactors, Polym. Reaction Eng., **10**, 2002, p. 181

9.ZHANG, J., MORRIS, A.J., MARTIN, E.B., KIPARISSIDES, C., Prediction of polymer quality in batch polymerization reactors using robust neural networks, Chem. Eng. J., **69**, 1998, p. 135

10.TIAN, Y., ZHANG, J., MORRIS, J., Modeling and optimal control of a batch polymerization reactor using a hybrid stacked recurrent neural network model, Ind. Eng. Chem. Res., **40**, 2001**,** p. 4525

11.TIAN, Y., ZHANG, J., MORRIS, J., Optimal control of a batch emulsion copolymerization reactor based on recurrent neural network models, Chem. Eng. Proc., **41**, 2002, p. 531

12.ZHANG, J., A reliable neural network model based optimal control strategy for a batch polymerization reactor, Ind. Eng. Chem. Res., **43**, 2004, p. 1030 – 1038.

13.WEI, J., XU, Y., ZHANG, J., Neural networks based model predictive control of an industrial polypropylene process, In: Proceedings of the 2002 IEEE International Conference on Control Applications, Glasgow, Scotland, U.K, 2002.

14.NG, C.W., HUSSAIN, M.A., Hybrid neural network – prior knowledge model in temperature control of a semi-batch polymerization process, Chem. Eng. Proc., **43**, 2004, p. 559

15.HORN, J., Trajectory tracking of a batch polymerization reactor based on input-output-linearization of a neural process model, Comput. Chem. Eng., **25**, 2001, p. 1561

16.CURTEANU, S., POPA, A.M., Applications of neural networks in polymerization reaction engineering, Rev. Roum. Chim, **49**, 2004, p. 3

17.NASCIMENTO, C.A.O., GIUDICI, R., N. Scherbakooff, Modeling of industrial nylon-6,6 polymerization process in a twin-screw extruder reactor. II. Neural networks and hybrid models, J. Appl. Polym. Sci.**, 72**, 1997, p. 905-912.

18.FERNANDES, F.A.N., LONA, L.M.F., PENLIDIS, A., Inverse modeling applications in emulsion polymerization of vinyl acetate, Chem. Eng. Sci., **59**, 2004, p. 3159

19.DHIB, R., HYSON, W., Neural network identification of styrene free radical polymerization, Polym. React. Eng., **10**, 2002, p. 101-113.

20.ZEYBEK, Z., YUCE, S., HAPOGLU, V., ALPBAZ, M., Adaptive heuristic temperature control of a batch polymerization reactot, Chem. Eng. Proc.**, 43**, 2004, p. 911

21.TIAN, Y., ZHANG, J., MORRIS, J., Dynamic on-line reoptimization control of a batch MMA polymerization reactor using hybrid neural network models, Chem. Eng. Technol., **27,** 2004, p. 1030

22.XIONG, Z., ZHANG, J., Neural network model-based on-line re-optimisation control of fed-batch processes using a modified iterative dynamic programming algorithm, Chem. Eng. Proc., **44,** 2005, p. 477

23.XIONG, Z., ZHANG, J., A batch-to-batch iterative optimal control strategy based on recurrent neural network models, J. Proc. Control, **15**, 2005, p. 11

24.ZHANG, J., Developing robust neural network models by using both dynamic and static process operating data, Ind. Eng. Chem. Res., **40**, 2001, p. 234

25.PRINCIPE, J., EULIANO, N., LEFEBVRE, C., Neural and Adaptive Systems: Fundamentals through Simulations, John Willy and sons, 2000, ISBN-10: 0471351679

26.CURTEANU, S., BULACOVSCHI, V., CONSTANTINESCU, M., Free Radical Polymerization of Methyl Methacrylate. Modeling and Simulation at High Conversion, Hung. J. Ind. Chem., **27**, 1999, p. 287

Manuscript received: 4.03.2010