RBFN Identification of an Industrial Polymerization Reactor Model

John D. Bomberger

*E. I. DuPont de Nemours & Co., Inc.
Experimental Station
Wilmington, DE 19880-0101, U.S.A*

Dale E. Seborg

*Department of Chemical Engineering
University of California
Santa Barbara, CA 93101, U.S.A.*

Babatunde A. Ogunnaike

*E. I. DuPont de Nemours & Co., Inc.
Experimental Station
Wilmington, DE 19880-0101, U.S.A*

Methods developed for radial basis function network (RBFN) identification are applied to a complex multiple-input, multiple-output (MIMO) simulation of a solution copolymerization reactor. For RBFN identification, k-means clustering and stepwise regression analysis are used. The practicality of applying these methods to large industrial identification problems is discussed, considering the restrictions of industrially practical input sequence design. The RBFN model has three inputs and two outputs, and the dimensionality of the identification problem poses some difficulties for nonlinear empirical model identification; specifically, the large amount of data required is a problem for plant testing and may cause computational difficulties for identification algorithms as well.

1. Introduction

Often, radial basis function network identification is applied only to single-input, single-output (SISO) example processes or simulations. This is also true for applications to real processes (Keulers, 1993; Eikens and Karim, 1994; Bomberger *et al.*, 1996). Application to SISO problems may be illuminating, but also can be somewhat misleading regarding the ease of use and data requirements of a particular identification method. In this chapter, RBFN identification methods are applied to a multivariable simulation of a solution copolymerization in a continuous stirred tank reactor (CSTR) developed by Congalidis *et al.* (1989). The simulation has seven input variables, four output variables, and a single disturbance variable. The simulation is treated as if it is an actual chemical process, and limitations on the amount of time available for plant identification tests and the type and duration of imposed changes in the plant operating conditions are considered.

To model the multivariable simulation, multiple-input, single-output (MISO) RBFNs are used. An RBFN takes the form of Fig. 1, and is a simple type of network model with single hidden layer of nodes and a linear output layer, with unity weighting between the input and hidden layer (Moody and Darken, 1989). Alternatively, an RBFN can be expressed as a weighted sum of radial basis functions

$$\hat{y}(i) = b_0 + \sum_{j=1}^M b_j \phi \left\{ \left[\mathbf{x}(i) - \mathbf{c}_j \right], \beta_j \right\} \quad (1)$$

where b_0 is the bias term, b_j are the connection weights between the j -th hidden layer node and the scalar output as shown in Fig. 1, $\mathbf{c}_j$ is the center of the j -th radial basis function (RBF), and β_j is a parameter controlling the width of the j -th RBF. The radial basis function used in this work is the Gaussian RBF:

$$\phi \left\{ \left[\mathbf{x}(i) - \mathbf{c}_j \right], \beta_j \right\} = \exp \left\{ - \frac{\left[\mathbf{x}(i) - \mathbf{c}_j \right]^T \left[\mathbf{x}(i) - \mathbf{c}_j \right]}{\beta_j} \right\} \quad (2)$$

The RBF center is analogous to the Gaussian mean, and the width parameter to the variance.

Nonlinear autoregressive models with exogenous inputs (NARX models) are used to build a dynamic, discrete time model of the simulation (Sjöberg *et al.*, 1995). For the NARX model, the RBFN inputs at sample i , $\mathbf{x}(i)$, are a

Figure 1. Radial basis function network.

vector of past values of the process inputs, $\mathbf{u}(i)$, and outputs, $\mathbf{y}(i)$. The relevant inputs and outputs are included in the network input vector,

$$\mathbf{x}(i) = \begin{bmatrix} y_1(i-1) & \dots & y_1(i-m_1) & \dots & y_{n_y}(i-1) & \dots & y_{n_y}(i-m_{n_y}) & u_1(i-\theta_1-1) & \dots & u_1(i-\theta_1-n_1) & \dots & u_{n_u}(i-\theta_{n_u}-1) & \dots & u_{n_u}(i-\theta_{n_u}-n_{n_u}) \end{bmatrix}^T \quad (3)$$

Parameters m_j and n_k are the number of past values of the j -th output or k -th input that are included in the NARX model; the number of process inputs and outputs are n_u and n_y , respectively. Model orders of $m_j = 0$ or $n_k = 0$ indicate that the corresponding j -th output or k -th input is not relevant to the process output being modeled. A scalar network output is used so that one MISO RBFN model has to be identified for each output of the process, and each MISO model may have a different input vector. The dead time for each input is θ_k , and may also be different for each input and each model. For RBFN modeling, the input and output data are usually normalized between zero and one.

The next section reviews the methods of RBFN identification used in this chapter. In Section 3 the multivariable simulation used is briefly

described. The results of RBFN model identification using the data from the simulation are discussed in Section 4.

2. RBFN Identification Methods

The chief feature of many RBFN identification methods is the placement of the RBF centers. Once the centers have been chosen and the other model parameters (m , n , θ , and β) have been specified in some manner, it is possible to use linear estimation methods, such as linear least squares, to determine the connection weights of the network. Two center selection techniques that are based on this approach are k-means clustering and linear regression methods.

The k-means clustering algorithm (Moody and Darken, 1989; Leonard and Kramer, 1991; Chen, *et al.*, 1992) partitions the data set used for identification into subsets of clusters, and finds a set of cluster centers. The cluster centers are used as the centers of the RBF nodes in the hidden layer of the network. The number of RBF centers (*i.e.*, the number of hidden nodes in the network), M , must be specified. The data are partitioned such that each data point in the network input space defined by $\mathbf{x}(i)$ in Eq. 3 is assigned to the cluster with the nearest center. There are both “off-line” (Leonard and Kramer, 1991) and “on-line” (Moody and Darken, 1989; Chen, *et al.*, 1992) versions of the clustering algorithm, but the two versions both minimize the total squared distance between the data points assigned to each cluster and the cluster centers. In the off-line version, the center of each cluster is initialized to a randomly chosen point in the input space. Next, each data point is assigned to the cluster with the nearest center. When all the data points have been assigned, each cluster center is moved to the mean of all the data points belonging to the cluster. The data points are then re-assigned to the cluster with the nearest center, and the procedure repeats until it converges. In general, the final center locations are not members of the identification data set, but are located within the spacial boundaries of the set.

Linear regression is another common method for determining the RBF centers. In this method, the RBF centers used in the network are chosen iteratively from a set of possible centers, which is typically the data set used for identification with each data point expressed in the manner of Eq. 3, and the connection weights are simultaneously determined using a version of

least squares. Each RBF center describes the associated RBF, provided that the model orders, time delay, and width parameter are already specified. Two common examples of linear regression used for RBFN identification are forward regression and stepwise regression. In forward regression (Chen *et al.*, 1991; Zhu and Billings, 1996) the next RBF to be added to the network is chosen to be the one that explains the most of the variance of the prediction error. The procedure is iterated until an error tolerance is satisfied or additional RBFs do not improve the model. Stepwise regression (Pottmann and Seborg, 1992) operates similarly, but at each iteration all the RBFs in the network are statistically analyzed and those which no longer contribute substantially to prediction error reduction are removed. Both linear regression methods automatically determine the number of hidden layer nodes, M .

In this paper, a k-means clustering algorithm and a stepwise regression algorithm with orthogonal least squares are both used for RBFN identification. For the k-means clustering algorithm, the initial cluster centers are chosen from the data set used for identification, expressed as the network input vector $\mathbf{x}(i)$. Furthermore, data clusters with no members (other than the originally chosen random data point) are removed and not included in the final RBFN model. The p-nearest-neighbor heuristic is used to determine the value of the width parameter β_j (Moody and Darken, 1989); this heuristic accounts for the desired amount of overlap between RBF nodes, and can be used after the RBF center positions are fixed. The stepwise algorithm used is similar to the method of Pottmann and Seborg (1992), but with fewer statistical tests; only overall and partial F-tests (Draper and Smith, 1981) and the overall and partial Akaike Information Criteria (Akaike, 1972) are used. The number and placement of the RBFs in the network are chosen from a set of possible RBFs, the data set $\mathbf{x}(i)$, for $i = 1, \dots, N$; the connection weights and bias term are then estimated using orthogonal least squares. If long identification data sets were to be used, however, stepwise regression with each sample in the identification set as a possible RBF center would be impractical. Two remedies for this problem are the use of a subset of randomly chosen samples from the identification data set as possible centers in the model, or the use of clustering algorithms to reduce the large data set to a number of clusters, the means of which can act as possible centers. These modifications reduce the computational requirements of the stepwise regression algorithm without necessarily reducing the model accuracy (Bomberger, 1997). The value of the width

Figure 2. Stream diagram of the copolymerization reactor.

parameter is the same for all the RBFs in the network, $\beta_j = \beta$, and is specified before stepwise regression begins. The value for β may be estimated by trial and error, from various heuristics, or from approximate gradient norms (Bomberger and Seborg, 1997). In this paper, however, different values of β are tried for each model identified using stepwise regression.

For both identification methods, the number of past inputs and outputs used in the network input vector $\mathbf{x}(i)$ in Eq. 3 for dynamic NARX models can be determined using the method of false nearest neighbors (Kennel *et al.*, 1992; Bomberger and Seborg, 1998). The difficulty with this method is that large amounts of data are typically required to produce meaningful model orders (Rhodes and Morari, 1997; Bomberger *et al.*, 1998). In this chapter, model orders are chosen to ensure that each model input has the expected relationship with each model output.

3. Solution Copolymerization Simulation

The simulation is based on a dynamic model of the solution copolymerization of methyl methacrylate (MMA) and vinyl acetate (VAc) in a benzene solvent with azobisisobutyronitrile (AIBN) as initiator and

Table 1. Process variables for the copolymerization simulation.

type	name	description	units
output	G_{pi}	polymer production rate	kg / h
output	y_{ap}	mole fraction A in copolymer	-
output	M_{pw}	weight average mol. weight	-
output	T_r	reactor temperature	K
input	G_a	monomer A (MMA) feed rate	kg / h
input	G_b	monomer B (VAc) feed rate	kg / h
input	G_i	initiator (AIBN) feed rate	kg / h
input	G_t	chain transfer agent (Ac) feed rate	kg / h
input	G_s	solvent (Benzene) feed rate	kg / h
input	G_z	inhibitor flowrate (<i>m</i> -DNB)	kg / h
input	T_j	reactor jacket temperature	K
input	T_{rf}	reactor feed temperature	K

acetaldehyde as the chain transfer agent. The polymerization takes place in a jacketed, perfectly mixed CSTR into which the monomers, initiator, solvent, and chain transfer agent are continuously added. A coolant flows through the jacket to remove the heat of polymerization. Inhibitors, such as *m*-dinitrobenzene (*m*-DNB) may be present in the monomer streams as an impurity. A schematic is shown in Fig. 2 (Congalidis *et al.*, 1989). The dynamic, first principles model has 12 state variables, and is based on a free radical mechanism reaction mechanism with 27 separate reactions. Simplifying assumptions include quasi-steady state, long chain hypothesis, and perfect mixing and constant volume and density in the reactor vessel.

It is assumed that a properly tuned PI controller is used to manipulate the coolant flowrate in order to obtain any desired value of jacket temperature; therefore, the jacket temperature, T_j , and not the coolant flowrate is specified as one of the inputs to the simulation. The other inputs are the monomer A (MMA) feed rate G_a , the monomer B (VAc) feed rate G_b , the initiator feed rate G_i , the chain transfer agent feed rate G_t , the solvent feed rate G_s , the inhibitor flowrate G_z (which is used to represent the inhibitor impurity in the monomer streams), and the reactor feed temperature T_{rf} . The simulation outputs are the important reactor output variables for product quality control: the polymer production rate G_{pi} , the mole fraction of monomer A in the copolymer Y_{ap} , the weight average molecular weight M_{pw} , and the reactor temperature T_r . These variables and their units are given in Table 1.

Table 2. Nominal operating point

G_{pi}	23.9 kg / h
y_{ap}	0.551
M_{pw}	33783
T_r	353.59 K
G_a	18.0 kg / h
G_b	90 kg / h
G_i	0.18 kg / h
G_t	2.7 kg / h
G_s	36.0 kg / h
G_z	0 kg / h
T_j	336.15 K
T_{rf}	353.15 K

In Banerjee *et al.* (1997) the solvent feed rate G_s and the reactor feed temperature T_{rf} are treated as measured disturbance variables, and are used along with the other simulation inputs for state estimation. The inhibitor flowrate is treated as an unmeasured disturbance, and is not used for estimation purposes. The same tactic can be pursued for RBFN identification: G_s and T_{rf} are considered as possible inputs to the RBFN models, but G_z is not. Therefore, $n_u = 11$ and $n_y = 4$, giving a total of 11 possible inputs to the network and four MISO RBFN models. However, identification results for such an RBFN model using limited data (of the type available from plant tests or long-term historical data) have proven to be less than satisfactory (Bomberger *et al.*, 1998). In this chapter, a subset of possible model inputs and outputs will be used: G_a , G_b , and G_i will be used as simulation inputs, with the other flowrates and temperatures set to constant values, and y_{ap} and M_{pw} will be used as outputs for model identification.

For simulation, a MATLAB® Simulink® model of the process is used. A sampling period of 1 h is used, to approximate the frequency of sampling necessitated by offline laboratory analysis of polymer properties. Gaussian additive measurement noise is added to all the simulated input and output variables, with a standard deviation of 1% the nominal value of the variable. The nominal values for the input and output variables are shown in Table 2. (Congalidis *et al.*, 1989; Banerjee *et al.*, 1997).

4. Identification of Copolymerization Simulation

As mentioned previously, an effort is made to treat the solution copolymerization simulation as an actual chemical plant for identification purposes; self-imposed constraints on the input sequence design and limitations of test length are respected, the sampling time is long, and noise is added to the simulated measurements. The identification is performed for open-loop conditions. In an actual polymer plant without an advanced control system, “recipe control” would probably be used, where the process inputs are specified to ensure a particular product quality, but feedback or feedforward control to correct for disturbances is not attempted. Instead, infrequent adjustments to the process would be made based on laboratory analysis of the product (Congalidis *et al.*, 1989).

4.1. Identification and Validation Data

Two types of input sequence designs are used: RUS and RDS. The RUS is a random, uniformly distributed sequence of step changes. A clock is used to time the steps, and at each clock period there is a specified probability for a step change in the input. The end value of the step change is uniformly distributed between lower and upper limits on the input. The RDS is a random, discretely distributed sequence of step changes. Again, a clock is used to time the steps, but a step must be made at the end of every clock period. The end value of the step change is one of a finite number of specified values set between lower and upper limits. The lower and upper limits on the variable inputs (G_a , G_b , and G_i) used in all three data sets are shown in Table 3.

Two sets of data are used for RBFN identification. Set 1 in Fig. 3 uses RUS inputs and is 360 samples long. Set 2 in Fig. 4 uses RDS inputs and is 360 samples long. A validation data set, Set 3, is also used; it has RUS inputs and is 360 samples long (Fig. 5).

Table 3. Lower and upper limits on variable flowrates (kg /h)

name	lower	upper
G_a	16	20
G_b	60	120
G_i	0.12	0.24

Figure 3. Identification data Set 1: RUS inputs.

Figure 4. Identification data Set 2: RDS inputs.

Figure 5. Validation data Set 3: RUS inputs.

Table 4. RBFN identification results using stepwise regression

output	β	M	iterations	I_1 (%)
<i>identification of Set 1</i>				
y_{ap}	0.1	40	80	2.45
	0.5	19	26	2.87
	1.0	16	24	3.03
	5.0	4	4	3.79
M_{pw}	0.1	47	94	1.26
	0.5	22	39	1.56
	1.0	17	26	1.53
	5.0	4	5	1.86
<i>identification of Set 2</i>				
y_{ap}	0.1	34	48	1.13
	0.5	22	42	1.35
	1.0	17	27	1.60
	5.0	9	12	1.81
M_{pw}	0.1	34	95	0.71
	0.5	24	37	0.68
	1.0	12	18	1.08
	5.0	5	5	1.07

4.2. Results

Stepwise regression with orthogonal least squares and k-means clustering are used to identify RBFN models from these data sets. For each of the models, the model orders $\mathbf{m} = [2 \ 2]^T$ and $\mathbf{n} = [1 \ 1 \ 1]^T$ are used, where

$$\mathbf{m} = [m_1 \ \cdots \ m_{n_y}]^T \quad (4)$$

$$\mathbf{n} = [n_1 \ \cdots \ n_{n_u}]^T \quad (5)$$

These vectors contain the number of past values to use for each output and input. The input vector is defined to be $\mathbf{u} = [G_a \ G_b \ G_i]^T$ and the output vector is $\mathbf{y} = [y_{ap} \ M_{pw}]^T$. In all cases, the dead time is $\theta_k = 0$, for $k = 1, \dots, n_u$.

The identification results for data in Sets 1 and 2 using stepwise regression with varying values for the width parameter β are shown in Table 4. Shown in the table are: the output for each MISO model; the width parameter β ; the number of RBFs in the model, M ; the number of iterations required for training; and the error index I_1 for one-step-ahead prediction

error over the identification data set. The error index for each output is the variance of the prediction error divided by the variance of the modeled output:

$$I_{type} = \frac{\sum_{i=1}^N [\hat{y}(i) - y(i)]^2}{\sum_{i=1}^N [y(i) - \bar{y}]^2} \times 100\% \quad (6)$$

where N is the number of data points in the data set. For one-step-ahead predictions, $type = 1$, and for multiple-step-ahead predictions $type = m$.

In Table 5 the results for RBFN model identification using k-means clustering on data Sets 1 and 2 are shown. In addition to the values shown in Table 4, the number of initial data clusters, M_0 , is shown. The number of RBFs in the model may not match the number of initial clusters, because clusters that do not contain data points are pruned during the identification procedure. For each RBF in the network, the width parameter β_j is determined using the p-nearest neighbor heuristic with $p = 2$; the mean value

Table 5. RBFN identification results using k-means					
output	M_0	M	$\bar{\beta}$	iterations	I_1 (%)
<i>identification of Set 1</i>					
y_{ap}	10	10	0.5864	8	10.43
	20	20	0.4919	9	7.72
	30	29	0.4210	13	4.34
	40	38	0.4040	7	2.78
M_{pw}	10	10	0.5959	9	5.22
	20	20	0.4911	9	3.36
	30	30	0.4084	7	2.39
	40	40	0.3486	10	2.02
<i>identification of Set 2</i>					
y_{ap}	10	9	0.5692	9	9.66
	20	20	0.4530	11	4.35
	30	29	0.3255	8	2.78
	40	40	0.2738	9	1.96
M_{pw}	10	10	0.5908	9	6.57
	20	20	0.4442	5	4.00
	30	30	0.3464	8	1.47
	40	40	0.3011	7	1.01

of the width parameters for all the RBFs in the network, $\bar{\beta}$, is shown in Table 5 as well.

The RBFN models were validated on data Set 3, which was not used for training, as an indication of model accuracy. The validation tests used one-step-ahead prediction. Results for the models identified using stepwise regression and k-means clustering are shown in Tables 6 and 7, respectively. It is also possible to validate the models using multiple-step-ahead prediction. It is a prediction of the model output given only initial conditions and the system inputs as a function of time, $u_k(i)$. To make multiple-step-ahead predictions, the network input vector $\mathbf{x}(i)$ in Eq. 3 is rewritten as a function of predicted system outputs and inputs:

$$\mathbf{x}(i) = [\hat{y}_1(i-1) \ \cdots \ \hat{y}_1(i-m_1) \ \cdots \ \hat{y}_{n_y}(i-1) \ \cdots \ \hat{y}_{n_y}(i-m_{n_y}) \ u_1(i-\theta_1-1) \ \cdots \ u_1(i-\theta_1-n_1) \ \cdots \ u_{n_u}(i-\theta_{n_u}-1) \ \cdots \ u_{n_u}(i-\theta_{n_u}-n_{n_u})]^T \quad (7)$$

Multiple-step-ahead predictions are a much tougher test of model accuracy than one-step-ahead prediction tests. Additionally, in the case of a multivariable system, one MISO model is needed for each system output when making multiple-step-ahead predictions. This permits many possible combinations of models when more than one MISO model is available for each output. In this case, each MISO model in Tables 4 and 5 was paired with the all models for the other output in each data set and for each identification method. The results were analyzed to pick the pairings that resulted in the most accurate RBFN models on both the identification and

Table 6. Error index for one-step-ahead prediction on validation data Set 3

I_1 (%)		
β	y_{ap}	M_{pw}
<i>identified on Set 1 using stepwise regression</i>		
0.1	81.68	55.61
0.5	22.43	5.38
1.0	9.67	3.74
5.0	3.28	2.88
<i>identified on Set 2 using stepwise regression</i>		
0.1	22.97	31.92
0.5	6.27	3.43
1.0	3.02	3.91
5.0	1.35	1.90

validation data sets. The most accurate models are shown in Table 8. The error indices $I_{m, \text{val}}$ and $I_{m, \text{val}}$ are the averages of the error indices for y_{ap} and M_{pw} for the validation and identification data sets, respectively; the overall error index, $I_{m, o}$, is the average of $I_{m, \text{val}}$ and $I_{m, \text{val}}$. Comparing to Table 8 to Tables 4-7, it is worthwhile to note that good one-step-ahead prediction

Table 7. Error index for one-step-ahead prediction on validation data Set 3

I_1 (%)		
M_0	y_{ap}	M_{pw}
<i>identified on Set 1 using k-means</i>		
10	37.07	27.89
20	32.85	18.16
30	40.71	33.84
40	34.01	15.75
<i>identified on Set 2 using k-means</i>		
10	13.25	12.56
20	7.59	13.90
30	6.52	7.50
40	8.45	6.38

Table 8. Error indices for multiple-step-ahead prediction

β for y_{ap}	β for M_{pw}	$I_{m, \text{id}}$ (%)	$I_{m, \text{val}}$ (%)	$I_{m, o}$ (%)
<i>stepwise regression models identified on Set 1</i>				
0.5	5.0	6.29	18.03	12.16
5.0	5.0	9.60	16.37	12.99
<i>stepwise regression models identified on Set 2</i>				
5.0	5.0	1.65	23.01	12.33
0.5	0.5	9.44	16.37	12.91
M_0 for y_{ap}	M_0 for M_{pw}	$I_{m, \text{id}}$ (%)	$I_{m, \text{val}}$ (%)	$I_{m, o}$ (%)
<i>k-means clustering models identified on Set 1</i>				
40	40	16.40	21.28	18.84
30	40	18.14	20.89	19.52
<i>k-means clustering models identified on Set 2</i>				
30	30	8.78	20.42	14.60
30	40	11.84	24.08	17.96

accuracy is not necessarily indicative of good multiple-step-ahead prediction accuracy.

4.3. Discussion

It is apparent that the RBFN model identification does not work equally well in all the cases presented. The accuracy of the model depends on the model parameters (especially the width parameter β and the number of RBFs M), the identification data, and the identification method.

Figure 6 shows the one-step-ahead prediction of two RBFNs identified on Set 1 using stepwise regression. The prediction of the RBFN with $\beta = 0.1$ is compared to the prediction of the RBFN with $\beta = 1.0$ and the actual data for validation data Set 3. It is apparent from this Fig. 6 and the results in

Figure 6. One-step-ahead predictions for validation data Set 3 for RBFNs identified on Set 1 using stepwise regression and $\beta = 0.1$ and $\beta = 1.0$.

Table 6 that increasing β generally results in improved model accuracy on the validation data. This can be explained by the localization of the Gaussian RBFs in the network input space, and the differences between the validation data Set 3 and the identification data Set 1.

Because Gaussian RBFs decay exponentially with the squared distance from the RBF center, the RBFs are localized in the input space of the network, as shown in Eq. 3. When the RBFs have decayed to near zero values, the network may be unable to accurately predict the model output. This situation can occur when the network extrapolates from the data used for identification, as is the case here. In Fig. 7, the distribution of the data in Set 1 is shown; there is little or no data available for $y_{ap} < 0.45$ or $M_{pw} < 30,000$. In the data distribution for Set 3 shown in Fig. 8 many data points are available for $M_{pw} < 30,000$ or $y_{ap} < 0.4$. The RBFNs identified using Set 1 must therefore extrapolate to predict the outputs present in Set 3. Because RBFs with larger values of β decay to zero more slowly, they are able to extrapolate somewhat better; therefore, RBFNs with larger values of β exhibit better accuracy over the validation data in Set 3.

The effect of the data distribution is evident in other ways as well. Two RBFN models identified using stepwise regression are compared in Fig. 9. One model was identified on Set 1, and the other on Set 2. Although both models use the same β value, the model identified on Set 2 is much more accurate. The data distribution for Set 2 in Fig. 10 has many more data points for $y_{ap} < 0.45$ or $M_{pw} < 30,000$ than Set 1; this means that the RBFNs identified on Set 2 are better trained in this region than are the RBFNs identified on Set 1. In general, the RDS type of input sequence used for the identification data in Set 2 permits a more uniform distribution of data throughout the network input space than an RUS input, especially for data sets of limited length.

Figure 7. Normalized histogram of identification data Set 1.

Figure 8. Normalized histogram of validation data Set 3.

Figure 9. One-step-ahead predictions for validation data Set 3 of RBFNs identified on Set 1 and Set 3 using stepwise regression with $\beta = 0.5$.

Model accuracy also depends on the identification method used. Given the same identification data, the two methods will not necessarily generate equally accurate models. Figure 11 compares RBFN models on the validation data Set 3, using the most accurate models from both the stepwise regression and k-means clustering identification methods. Both models were identified on Set 2 data. While both RBFNs are fairly accurate, the RBFN identified using stepwise regression provides better predictions. This result probably occurs because stepwise regression optimizes the number and placement of RBF centers from a set of possible centers to find the best model for given model orders and width parameter, in terms of accuracy and parsimony. The k-means clustering method, on the other hand, finds a fixed number of RBF centers based on the natural clustering of data points in the input space; this ensures that RBF centers are usually located only where there is a significant amount of data. Also, in the k-means clustering

algorithm, the choices for β_j are made using the p-nearest-neighbor heuristic, and are not in any way optimized to find values that might improve model accuracy for a given set of RBF centers.

As noted previously, multiple-step-ahead prediction is a more stringent test of model accuracy than one-step-ahead prediction. In Fig. 12 a comparison of one-step-ahead prediction and multiple-step-ahead prediction for the best RBFN models is made. The models were identified using stepwise regression on Set 2, with $\beta = 0.5$ for both y_{ap} and M_{pw} . While both predictions are fairly accurate, the one-step-ahead prediction is substantially better. The multiple-step-ahead prediction is worse because the networks were trained to make one-step-ahead predictions only, and because errors in the prediction are compounded in future samples.

An inadequate amount of data, or range of data, is the major source of difficulty for nonlinear model identification of multivariable processes. This especially applies to RBFN model identification, where the radial basis functions may be significantly localized in the network input space defined by the input vector $\mathbf{x}(i)$, inhibiting model extrapolation. In contrast, for linear model identification, limited data is not as important a problem. This result occurs because the superposition principle central to linear system modeling requires that the size or direction of change in a process does not alter the general process behavior. Similarly, for linear MIMO systems, changes in the process inputs act in an additive fashion and it is not necessary that all possible combinations of changes for the n_u inputs be made. Evidence is provided by the commercial success of linear multivariable control software like Aspen Technology's DMC, where linear models of large multivariable systems are identified and used for model predictive control.

It is promising, however, that the RBFN nonlinear modeling technique is able to accurately predict the nonlinear behavior of the copolymerization simulation in regions for which only limited data exists. An RBFN model may have learned little of the overall behavior of the process, but, trained only on historical data, the model is still able to predict the behavior of the process for similar operating conditions (Bomberger *et al.*, 1998). Pavilion Technologies, a commercial neural network software and engineering company, uses the ability of neural network models to easily learn nonlinear behavior by identifying steady state process models from historical operating data. These models can then be used to optimize process inputs to obtain desirable process outputs, or can be coupled with dynamic models for use in nonlinear model predictive control strategies (Keeler *et al.*, 1997).

Figure 10. Normalized histogram of identification data Set 2.

Figure 11. Comparison of one-step-ahead prediction for validation data Set 3 for models identified on Set 2 using stepwise regression with $\beta = 5.0$ and k-means clustering with $M_0 = 30$.

5. Conclusions

In the literature, RBFN models have been shown to be easily applicable to single-input, single-output and other low dimensional identification problems. This research has focused on identification of a multivariable, industrially relevant simulation of a solution copolymerization reactor. The RBFN model has three inputs and two outputs, and the dimensionality of the identification problem poses some difficulties for nonlinear empirical model identification. The primary difficulty is the amount of data required, which is an aspect of the “curse of dimensionality.” The data requirement is a problem for plant testing and may cause computational difficulties for

Figure 12. Comparison of one-step-ahead prediction and multiple-step-ahead prediction for validation data Set 3 for models identified using stepwise regression on Set 2 ($\beta = 5.0$ for both y_{ap} and M_{pw}).

identification algorithms as well. Both the stepwise regression algorithm and the k-means clustering algorithm work robustly and quickly, even for high dimensional network input vectors. The stepwise regression algorithm exhibits somewhat more consistent results and has better overall accuracy at the cost of increased computational requirements which can be prohibitive for large data sets.

References

Akaike, H., Information theory and an extension to the maximum likelihood principle. In *2nd Int. Symp. on Information Theory, Tsahkador, Armenia, USSR*,

- Sept. 2-8, 1971, B. N. Petrov and F. Csáki, ed., Akadémiai Kiadó, Budapest, (1973), 267-281.
- Banerjee, A., Y. Arkun, B. Ogunnaike, and R. Pearson, Estimation of nonlinear systems using linear multiple models. *AIChE J.*, **43**, 5, (1997), 1204-1226.
- Bomberger, J. D., *Radial Basis Function Networks for Process Identification*. Ph.D. Dissertation, University of California, Santa Barbara (1997).
- Bomberger, J. D., G. Lightbody, D. E. Seborg, and G. W. Irwin, Experimental evaluation of neural nonlinear modeling techniques. In *Chemical Process Control V*, ed. Kantor, J. C., C. E. Garcia, and B. Carnahan, AIChE Sympos. Ser., 93(316), (1997), 280-283.
- Bomberger, J. D. and D. E. Seborg, Determination of model order for NARX models directly from input-output data. *J. Proc. Cont.*, **8**, 5-6, (1998), 459-468.
- Bomberger, J. D. and D. E. Seborg, Determination of the width parameter for Gaussian radial basis function networks using approximate gradient norms. Paper presented at AIChE Annual Meeting, Nov. 16-21, Los Angeles (1997).
- Bomberger, J. D., D. E. Seborg, and B. A. Ogunnaike, RBFN identification of a solution copolymerization model. *Proc. of the 1998 ACC*, June 24-26, Philadelphia, PA (1998).
- Chen, S., S. A. Billings, and P. M. Grant, Recursive hybrid algorithm for non-linear system identification using radial basis function networks. *Int. J. Control* **55**, 5, (1992), 1051-1070.
- Chen, S., C. F. N. Cowan, and P. M. Grant, Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Trans. Neural Networks*, **2**, 2, (1991), 302-309.
- Congalidis, J. P., J. R. Richards, and W. H. Ray, Feedforward and feedback control of a solution copolymerization reactor. *AIChE J.*, **35**, 6, (1989), 891-907.
- Draper, N. R. and H. Smith, *Applied Regression Analysis*. 3rd. Ed., John Wiley, New York (1998).
- Eikens, B. and M. N. Karim, Real-time control of a waste water neutralization process using radial basis functions. *Preprints IFAC-ADCHEM '94*, May 25-27, Kyoto, Japan, (1994), 125-130.
- Keeler, J., G. Martin, G. Boe, S. Piche, U. Mathur, and D. Johnson, The Process Perfector: the next step in multivariable control and optimization. Technical Report, Pavilion Technologies, Inc., Austin, TX (1997).
- Kennel, M. B., R. Brown, and H. D. I. Abarbanel, Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Phys. Rev. A*, **45**, 6, (1992), 3403-3411.
- Keulers, M., Structure and parameter identification of a batch fermentation process using non-linear modelling. *Proc. of the 1993 ACC*, June 2-4, San Francisco, (1993), 2261-2265.
- Leonard, J. A. and M. A. Kramer, Radial basis function networks for classifying process faults. *IEEE Control Sys. Mag.*, **11**, April, (1991), 31-38.
- Moody, J. and C. J. Darken, Fast learning in networks of locally-tuned processing units. *Neural Computation*, **1**, (1989), 281-294.

- Pottmann, M. and D. E. Seborg, Identification of nonlinear processes using reciprocal multiquadric functions. *J. Process Control*, **2**, 4, (1992), 189-203.
- Rhodes, C. and M. Morari, The false nearest neighbors algorithm: an overview. *Computers Chem. Engng.*, **21**, Suppl., (1997), S1149-S1154.
- Sjöberg, J., Q. Zhang, L. Ljung, A. Benveniste, B. Deylon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsy, Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, **31**, 12, (1995), 1691-1724.
- Zhu, Q. M. and S. A. Billings, Fast orthogonal identification of nonlinear stochastic models and radial basis function neural networks. *Int. J. Control*, **64**, 5, (1996), 871-886.

Acknowledgements

The authors gratefully acknowledge the financial support of the National Science Foundation (Grant # CTS-9424094) and DuPont. The cooperation of Dr. Yaman Arkun (Dean of the College of Engineering, KOC University, Turkey) in supplying the MATLAB® code for the copolymerization simulation is appreciated.