Catalysis Today 164 (2011) 275–281

Contents lists available at ScienceDirect

Catalysis Today

journal homepage: www.elsevier.com/locate/cattod

Artificial neural network modeling of forced cycling operation between propane steam reforming and CO_2 carbon gasifier

Viswanathan Arcotumapathy, Feraih Alenazey, Adesoji A. Adesina*

Reactor Engineering and Technology Group, School of Chemical Engineering, The University of New South Wales, NSW 2052, Australia

ARTICLE INFO

Article history:
Received 1 July 2010
Received in revised form 5 December 2010
Accepted 11 December 2010
Available online 12 January 2011

Keywords:
Artificial neural network
Periodic operation
Steam reforming

ABSTRACT

This investigation has employed artificial neural network (ANN) modeling to describe the complex relationship between the forced cycling parameters and the reactor performance during periodic operation between propane steam reforming and CO_2—carbon gasifying agent. Experimental data from our laboratory were assessed against different ANNs and based on a 2-way ANOVA treatment of various error indices, a two-hidden layer network with 5 neurons emerged as the best model for both descriptive and predictive purposes. Cycle split has the most significant (85%) positive effect on the improvement in H_2 and CO production and the appearance of resonant peaks while cycle period appeared to have detrimental effect on product yield.

1. Introduction

Hydrocarbon steam reforming over Ni-based catalyst for H_2 or syngas production is industrially carried out using excess steam to avoid unwanted coke deposition [1]. The latter ultimately leads to poor reactor performance due to catalyst deactivation. In order to minimize energy costs (associated with steam generation) and improve reactor operation, CO_2 may be used as a carbon gasifier [2]. This has the added advantage of reducing overall greenhouse gas (GHG) emissions from the reforming plant. In a recent study [3], we demonstrated that periodic composition cycling as illustrated in Fig. 1, between the CO_2 gasifying agent (C_1) and propane steam reforming feed (C_2) has significant benefits in terms of increased syngas yield, tailored H_2:CO ratio suitable for downstream conversion in a gas-to-liquid (GTL) fuels plant and superior catalyst stability and longevity.

The time-average rate behaviour with respect to cycle frequency was characterized by resonant peaks depending on the cycle split (symmetry) employed. Modeling of periodically operated reactors is a multifarious exercise strongly reliant on the choice of reaction mechanism. Mihail and Paul proposed adsorption/desorption models [4] which Jain et al. [5,6] latter rejected as inadequate to explain the resonance effects seen in their experimental studies for SO_2 oxidation. Other attempts at providing mechanistically based models to describe the complex phenomena observed during forced cycling of catalytic reactors have also been met with mixed success. Indeed, Silveston et al. [7] have indicated that adsorption/desorption type models cannot accurately capture the nonlinear behaviour often implicated in periodically operated catalytic processes. However, optimal control of the forced cycling reactor requires a reliable model of its performance in order to harness the benefits (improved product yield and selectivity) associated with this dynamic operation. In this work, we have taken an artificial neural network (ANN) approach which does not require detailed understanding of the reaction mechanism. In particular, ANN permits the utilization of fewer latent variables which contain inherent information about the process. Indeed, ANN models are significantly more potent in capturing the process attributes with higher predictability and better description than multivariate data analysis as espoused by Bulsari [8].

Although a detailed mechanistic basis for this nonlinear behaviour is presently unavailable, a parsimonious model for process optimization may be secured via ANN analysis of existing data. ANN is a mathematical analogue of how the human brain recognizes and reproduces cause-effect relational patterns upon training (multiple-input–multiple-output systems). The ANN modeling of complex nonlinear behaviour or pattern in chemical engineering systems has been reported [8–13]. Even though nonlinear estimators like Holographic Research Strategy (HRS) and Genetic Algorithm (GA) are available, Tompos et al. [14] reported the robustness of these methods only after securing the relationship between the composition–activity of the catalysts via ANNs in their study. Moreover, they found that the predictive ability of the ANNs assisted in the enhancement of both HRS and GA methods to identify the optimized catalyst composition. The aim of the present work is to procure an ANN model to describe and predict the non-

* Corresponding author at: School of Chemical Engineering, The University of New South Wales, Sydney, NSW, Australia 2052. Tel.: +61 2 9385 5268; fax: +61 2 9385 5966.
E-mail address: a.adesina@unsw.edu.au (A.A. Adesina).

0920-5861/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.cattod.2010.12.027

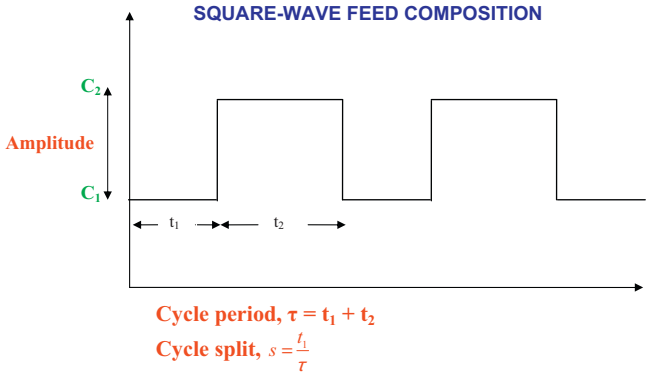

Fig. 1. Operating variables during forced cycling of a reactor.

linear dependency of key reaction metrics on cycle parameters (frequency and split) during periodic composition cycling between propane steam reforming feed and CO_2, a carbon gasifier without recourse to formal mechanistic details. In particular, since ANN is a mathematical imitation of the biological neural data processing operation, optimization of the network architecture used in this study was patterned after the Fibonacci search strategy which has meaningfully captured the behaviour in many natural ecosystems [15–17]. Moreover, the integer nature of optimum number of neurons in an ANN lends itself better to a Fibonacci-type algorithm rather than a continuous function-dependent method.

2. Theoretical basis and numerical procedure

2.1. Data scaling

The input variables were taken as the cycle period and split (fraction of the period spent under CO_2 feed for carbon gasification). In view of the difference in units, magnitude and range of the input variables, each variable needs to be rescaled so that values fall within the interval [0.01, 0.99]. The transformation was carried out using:

$$\hat{x} = \frac{x - m}{M - m} \tag{1a}$$

where $\hat{x}$ is the rescaled variable, with M and m chosen such that:

$$\frac{x_{\min} - m}{M - m} = 0.01 \quad \text{and} \quad \frac{x_{\max} - m}{M - m} = 0.99 \tag{1b}$$

where $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the original variable, x respectively. The variables in the output vector, H_2, CH_4 and CO were also similarly rescaled.

2.2. Methodology to select optimum neurons and architecture

The performance of an ANN is strongly dependent on the number of neurons employed and the network architecture. For instance, a simple feed-forward network with supervised learning algorithms such as back-propagation algorithm is competent in learning the relationship between the given inputs and targets for a nonlinear process data set [8–13]. Thus, a systematic procedure to optimize the number of neurons for building the multilayer feed-forward neural network to map the input–target relationship of the given data set is critical to the overall modeling exercise. The number of weights required to capture a reliable relationship between a given input and output is directly proportional to the number of neurons. ANN being a mimicry of the biological nervous system would be more appropriately optimized if the number of neurons were chosen in a manner that is patterned after natural evolutionary systems. Interestingly many naturally occurring networks (e.g. in forestry, microbiology and river systems—delta formation) containing optimum nodal points are known to be described by the Fibonacci sequence [15–17]. Consequently, it seems logical to derive the number of neurons in our ANN simulation of the optimal catalyst design using members of the Fibonacci series [18]. To the best of our knowledge, this is the first appropriation of the Fibonacci search strategy in ANN simulation of either chemically reactive systems or artificially contrived proceesses. The numerical procedure was carried out in MATLAB Neural Network Toolbox™ version 7.8.0.347 (R2009a) and may be summarized as:

Step 1: The MATLAB Neural Network Toolbox random data division function was used to partition the 34 data cases into 24, 5 and 5 for training, validation and testing of the networks respectively. A three-layer architecture (input, hidden and output layers) was considered with different artificial neural networks (ANNs) arising from variation in the number of neurons in the hidden layer. A tangent sigmoid transfer function was used for the neurons in the hidden layer. The number of neurons in the hidden layer for each ANN was chosen as terms of the Fibonacci series to introduce optimality in both the number of ANNs and computational effort. The total number of neurons required to obtain excellent model adequacy of an empirically fitted approximation and the ratio of number of neurons in the hidden layer to the number of data cases has to satisfy two opposing criteria detailed in Haykin [19] and Barron [20], namely:

(a) *Accuracy of best approximation*—the size of the hidden layer, N_{hl}, must be as large as possible in accordance with the universal approximation theorem and
(b) *Accuracy of the empirical fit to the approximation*—the ratio of the hidden layer size to number of training data cases, N_t, i.e. N_{hl}/N_t must be small.

In order to accommodate both criteria, we adopted the rule of thumb, $1 \leq N_{hl}/N_t \leq 10$ [9]. Given that N_t is 24 for this study, ANNs with 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 neurons in the hidden layer were chosen (total of 11 ANNs since the 2nd and 3rd ANNs with 1 neuron each, are mere repetitions while the 1st ANN with 0 neurons is physically inadmissible).

Levenberg–Marquardt algorithm was used to train each network based on the randomly selected 24 data cases. The random data division function was called 1000 times (cycles) to pick different combinations of the training, validation and testing data sets from the total data pool in each cycle. Thus, each of the 11 ANNs was trained 1000 times with different data set combinations resulting in a total of 11,000 trained, validated and tested ANNs. In this study after each cycle, i.e. for every ANN trained, after the entire training procedure was completed (R-square value ≥ 0.92 in nearly all cases), the weights and biases were used to simulate the network using all inputs (34 data cases) from the data set while the resulting output from the network after simulation and the targets from the data set were used to calculate the performance of the networks which was evaluated by different error indices, namely:

$$\text{Sum-of-squared error (SSE)} = \sum_{s=1}^{N} (y_s^o - t_s)^2 \tag{2}$$

$$\text{Mean squared error (MSE)} = \frac{1}{N} \sum_{s=1}^{N} (y_s^o - t_s)^2 \tag{3}$$

$$\text{Root-mean-squared error (RMSE)} = \sqrt{\frac{1}{N} \sum_{s=1}^{N} (y_s^o - t_s)^2} \tag{4}$$

$$\text{Mean absolute error (MAE)} = \frac{1}{N}\sum_{s=1}^{N}||y_s^o - t_s|| \quad (5)$$

$$\text{Mean absolute percentage error (MAPE)} = \frac{1}{N}\sum_{s=1}^{N}\left(\left\|\frac{y_s^o - t_s}{t_s}\right\| \times 100\right)\% \quad (6)$$

where N is the number of data cases in the data set; y_s^o is the output of the network for sth data case; t_s is the target of the sth data case.

Since the simulation errors namely: SSE (cumulative variance), MSE (indicator of training process adequacy) and RMSE (index for model predictive capability) are capable of explaining the variance in the errors while the MAE & MAPE explain the linearity of the model. Furthermore, to asses which error is significant in explaining the variance in the data with increasing number of neurons, the results were subjected to a 2-way ANOVA (error indices-vs-number of neurons) to determine the most important error indices that account for the variation due to different number of neurons in the ANN. The ANOVA computations were performed using software, R (version 2.9.0) with decisions made at 95% confidence level. All ANNs were then ranked for each of the statistically significant error indices (using the average of individual error index over 1000 training cycles). '1' was assigned to the ANN with the lowest error index average value and '11' was given to the ANN with the highest error index average value. The sum of the rank of each ANN across the statistically important error indices was then obtained (termed, gross rank). Subsequently the ANN with the lowest gross rank was given 'position 1' and accepted as the 'best' ANN from the 11 member-set.Since the difference between successive Fibonacci numbers is generally greater than 1 (with exception of the first four terms of the series), the number of neurons in the 'best' ANN from the above procedure, may require further refinement as detailed in step 2 to obtain optimum number of neurons to the nearest integer.

Step 2: Given that N is the number of neurons in the 'best' ANN from step 1, a new set of networks would be constructed such that the number of neurons is $x, x+1, x+2, \ldots N, N+1, N+2, \ldots z$ where x and z are the adjacent (before and after respectively) Fibonacci numbers to N. Training and performance evaluation of the ANNs were also carried out as explained in step 1 (with 100 cycles of training for each network), followed by the selection of an updated 'best' ANN.

Step 3: After deciding the optimum number of neurons (N^{opt}) with the aid of 'best' ANN from step 2 to describe the data, the choice of ANN architecture is an important step in the modeling exercise. The total number of ANN (S_{ANN}), for N^{opt} neurons distributed over h hidden layers is given by

$$\text{for } h = 1,\ S_{ANN} = 1$$
$$\text{for } h = 2,\ S_{ANN} = N^{opt} - 1$$
$$\text{for } h \geq 3,\ S_{ANN} = (N^{opt} + 1 - h) + \tfrac{1}{2}(N^{opt} + 1 - h)(N^{opt} - h)$$

and for instance the combinations of two and three hidden-layer networks are obtained as shown in Tables 1a and 1b respectively. While it is possible to use 1 ANN with a single hidden layer to secure a reliable relationship between the input–output variables, the possible reduction in various error indices that may arise from different ANN architecture with N^{opt} neurons from step 2 was further explored. This allows the ANN to capture higher-order statistics such as interaction between different hidden layers and other system nonlinearities [19,21]. We, however, note that the total number of optimum neurons between input-to-output would not vary with number of hidden layers in the ANN. Thus, the N^{opt} obtained in step 1 for the single hidden layer is still valid for the multi-hidden layer ANN with the same input–output characteristics [22]. Specifically, N^{opt} neurons were distributed over ANNs with different hidden layers and the new set of ANNs was then assessed with the data pool to secure members with the least error indices.

Table 1a
Two hidden layer ANNs ($h = 2$ for this architecture).

First hidden layer	Second hidden layer
1	$[N^{opt} - (h-1)]$
2	$[N^{opt} - (h)]$
3	$[N^{opt} - (h+1)]$
⋮	⋮
⋮	⋮
$[N^{opt} - (h+1)]$	3
$[N^{opt} - (h)]$	2
$[N^{opt} - (h-1)]$	1

Table 1b
Three hidden layer ANNs ($h = 3$ for this architecture).

First hidden layer	Second hidden layer	Third hidden layer
1	1	$[N^{opt} - (h-1)]$
1	2	$[N^{opt} - (h)]$
1	⋮	$[N^{opt} - (h+1)]$
1	⋮	$[N^{opt} - (h+2)]$
1	⋮	⋮
1	$[N^{opt} - (h+1)]$	3
1	$[N^{opt} - (h)]$	2
1	$[N^{opt} - (h-1)]$	1
2	1	$[N^{opt} - (h)]$
2	2	$[N^{opt} - (h+1)]$
2	⋮	$[N^{opt} - (h+2)]$
2	⋮	⋮
2	⋮	⋮
2	$[N^{opt} - (h+2)]$	3
2	$[N^{opt} - (h+1)]$	2
2	$[N^{opt} - (h)]$	1
3	1	$[N^{opt} - (h+1)]$
3	2	$[N^{opt} - (h+2)]$
3	⋮	$[N^{opt} - (h+3)]$
3	⋮	⋮
3	⋮	⋮
3	$[N^{opt} - (h+3)]$	3
3	$[N^{opt} - (h+2)]$	2
3	$[N^{opt} - (h+1)]$	1
⋮	⋮	⋮
⋮	⋮	⋮
⋮	⋮	⋮
$[N^{opt} - (h-1)]$	1	1

2.3. Relative importance of input variables

The weights of the optimal ANN model can be used to derive the relative importance of propane steam reforming and CO_2 carbon gasification with respect to CH_4, H_2 and CO yield. Garson [23] proposed a method which has been widely used [22,24,25] for evaluating the relative importance of each input variable in a single hidden layer ANN via the connection weight for the ith variable to

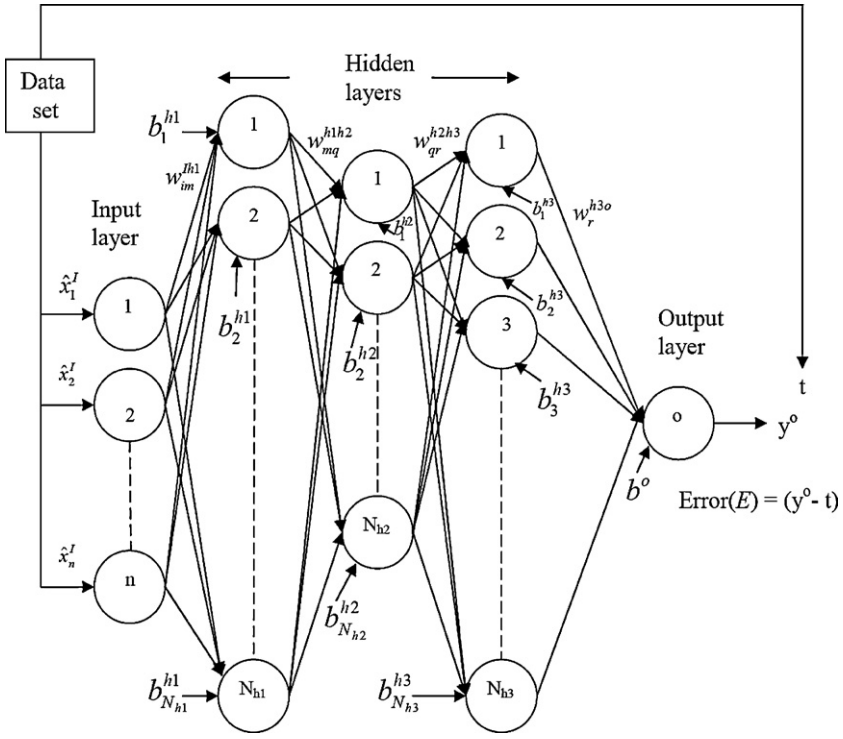

Fig. 2. A multilayer feedforward neural network with three hidden layers.

the output is given as:

$$cw_{is}^{Io} = \frac{\sum_{m=1}^{N_{h1}}(|w_{im}^{Ih1}||w_{ms}^{h1o}|)/\sum_{i=1}^{n}|w_{im}^{Ih1}|}{\sum_{i=1}^{n}\sum_{m=1}^{N_{h1}}(|w_{im}^{Ih1}||w_{ms}^{h1o}|)/\sum_{i=1}^{n}|w_{im}^{Ih1}|} \quad (7)$$

where I, $h1$ and o represents input, hidden, and output layers respectively and $i = 1, 2, \ldots, n$ (number of neurons in the input layer, i.e. number of variables), $m = 1, 2, \ldots, N_{h1}$ (number of neurons in hidden layer), $s = 1, 2, \ldots, N_o$ (number of neurons in output layer) and w is the raw weights between layers.

However, the direction (i.e. detrimental or enhancement effect) of the input variable's influence on the output is not explicitly accommodated for. Moreover, it tends to assign more prominent role to the connection weights between the hidden to output layer than between the input to hidden layer. Alternatives to the evaluation of the importance of input variables have been proposed [10,26,27] in order to minimize these drawbacks and improve the learning rate of the neural network. Olden et al. [27] proposed a connection weight method (CWM) and compared it against other evaluative techniques such as the Garson, partial derivatives, input perturbation, sensitivity analysis, forward stepwise addition, backward stepwise elimination and improved stepwise selection methods and concluded that the CWM is significantly more accurate in procuring the relative importance between input-to-output than all others. An advantage of the CWM is that the results indicate whether an input variable has a detrimental or favorable influence with respect to the output. The connection weights (CW) for a typical three hidden layer model (as shown in Fig. 2) are given by equations 8, 9 and 10 for; inputs and 2nd hidden layer, inputs and 3rd hidden layer and over all connection weights between the inputs and output respectively:

$$cw_{iq}^{Ih2} = \sum_{m=1}^{N_{h1}} w_{im}^{Ih1} w_{mq}^{h1h2} \quad (8)$$

$$cw_{ir}^{Ih3} = \sum_{q=1}^{N_{h2}} cw_{iq}^{Ih2} w_{qr}^{h2h3} \quad (9)$$

$$cw_{is}^{Io} = \sum_{r=1}^{N_{h3}} cw_{ir}^{Ih3} w_{rs}^{h3o} \quad (10)$$

where I, $h1$, $h2$, $h3$ and o represents input, 1st hidden, 2nd hidden, 3rd hidden and output layers respectively and $i = 1, 2, \ldots, n$ (number of neurons in the input layer which is same as the number of variables); $m = 1, 2, \ldots, N_{h1}$ (number of neurons in 1st hidden layer); $q = 1, 2, \ldots, N_{h2}$ (number of neurons in 2nd hidden layer); $r = 1, 2, \ldots, N_{h3}$ (number of neurons in 3rd hidden layer); $s = 1, 2, \ldots, N_o$ (number of neurons in output layer).

3. Results and discussion

3.1. Selection of optimum number of neurons

The 34 data cases (cf. Table A1 of Appendix A) were analyzed via ANN implementation for forced cycling operation between propane steam reforming and CO_2 carbon gasifier. The optimum number of neurons required in the hidden layer(s) of the ANN model that can achieve significant model accuracy is carried out by training 11 single hidden layer feed-forward ANNs. These networks were distinguishable by the number of neurons in the hidden layer namely: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 and 144 which belongs to the Fibonacci series. The average SSE, MSE, RMSE, MAE and MAPE for each network with varying number of neurons was calculated following the procedure detailed in step 1–3 of Section 2.2.

The analysis of variance of SSE, MSE, RMSE, MAE and MAPE against different networks (i.e. against total number of neurons in the hidden layer) revealed that only SSE and MAE are significant (as shown in Table 2a). The network model with 5 and 8 neurons in the hidden layer ranked first based on the significant errors as shown in Table 2b and the corresponding parity plots for these models are

Table 2a
2-way ANOVA of error indices against neurons.

Error index	Degrees of freedom	Sum of squares	Mean square	*F*-value	Probability (%)
SSE	1	19274.1	19274.1	202.5414	7.524×10^{-04}
MSE	1	NA	NA	NA	>100
RMSE	1	51.2	51.2	0.5384	49.075
MAE	1	869.7	869.7	9.1390	2.330
MAPE	1	0.9	0.9	0.0096	92.533
Residuals	6	571.0	95.2		

NA—The values are not available for MSE because the probability value is greater than 100% which is physically inadmissible.

Table 2b
Ranking of networks based on 2-way ANOVA.

ANN	Error indices		Error index based ranking		Gross rank	Position of gross rank
Number of neurons in hidden layer	SSE	MAE	SSE	MAE		
1	2.762103	0.132121	3	5	8	3
2	2.905645	0.129348	4	4	8	3
3	3.048169	0.135167	5	6	11	6
5	2.410055	0.090595	1	2	3	1
8	2.562908	0.090213	2	1	3	1
13	4.432908	0.110623	6	3	9	5
21	10.12715	0.170215	7	7	14	7
34	20.35969	0.238841	8	8	16	8
55	36.7802	0.30835	9	9	18	9
89	45.60238	0.335051	10	10	20	10
144	54.28943	0.361181	11	11	22	11

shown in Fig. 3a and b respectively. In fact the errors were on a runaway trajectory from 8 neurons (cf. in Table 2b) supports the theory stated by Barron [20]. As a result, $N = 8$ with $x = 5$ and $z = 13$ were used in further refinement of the selection of N^{opt} as previously discussed.

The implementation of step 2, requires that $x = 5$ and $z = 13$, resulting in 9 new ANNs with different number of neurons, in the single hidden layer, viz; 5, 6, 7, 8, 9, 10, 11, 12, and 13. A 2-way ANOVA of the ANNs (9) vs error index (5) (cf. in Table 3a) revealed that SSE, RMSE and MAPE were the only statistically significant error indices. Further treatment confirmed that the ANN with 5 neurons would be the 'best' choice (as shown in Table 3b). The relevant parity plot is illustrated in Fig. 4.

3.2. Optimization of the ANN architecture

With $N^{opt} = 5$ neurons, a new set of ANNs of 1-, 2-, 3- and 4-hidden layers were derived as detailed in step 3 resulting in 15 ANNs distributed as 1 single-hidden layer ANN, 4 two-hidden layer ANNs, 6 three-hidden layer ANNs and 4 four-hidden layer ANNs. The optimal ANN model with 3 and 2 neurons in the 1- and 2-hidden layer respectively had the minimum performance index for all the errors (SSE = 0.6589, MSE = 0.0064, RMSE = 0.0803, MAE = 0.0585, MAPE = 24.47). The parity plot for this model is also shown in Fig. 5. Individual parity plots for the optimum ANN model for H_2, CH_4 and CO yield are depicted in Figs. 6–8 respectively. It may be seen from Figs. 5–8 that the optimum ANN has learned the forced cycling process sufficiently well and can be used as predictive model for further optimisation of the process.

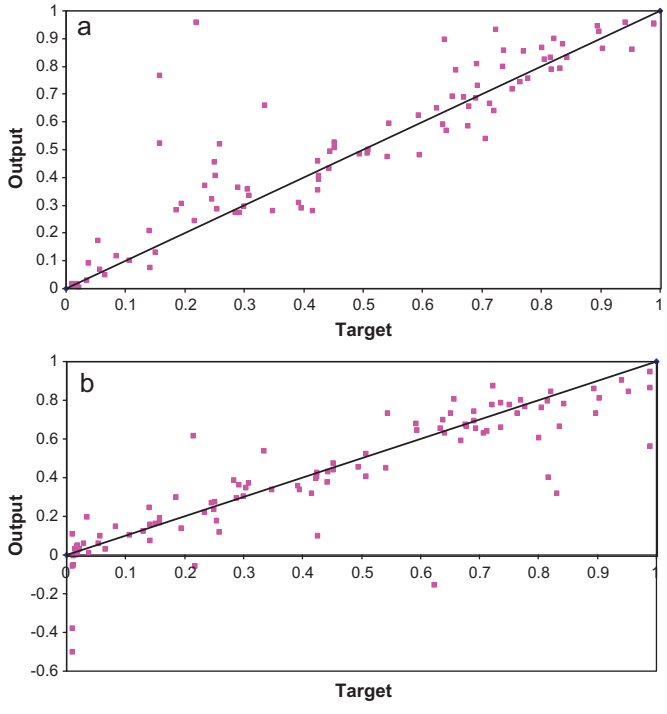

Fig. 3. (a) Predicted vs experimental values for single-hidden layer (5 neurons) ANN model. (b) Predicted vs experimental values for single-hidden layer (8 neurons) ANN model.

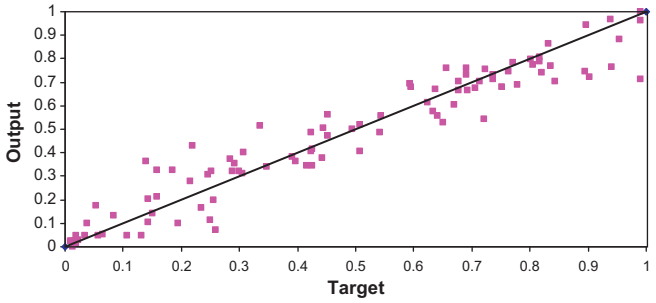

Fig. 4. Predicted vs experimental values for single-hidden layer (5 neurons) ANN model.

Table 3a
2-way ANOVA of error indices against neurons.

Error index	Degrees of freedom	Sum of squares	Mean square	*F*-value	Probability (%)
SSE	1	4.481	4.481	6.8173	5.9358
MSE	1	NA	NA	NA	>100
RMSE	1	44.394	44.394	67.5340	0.1195
MAE	1	3.924	3.924	5.9689	7.0969
MAPE	1	4.571	4.571	6.9539	5.7760
Residuals	4	2.629	0.657		

NA—The values are not available for MSE because the probability value is greater than 100% which is physically inadmissible.

Table 3b
Ranking of networks based on 2-way ANOVA.

ANN	Error indices			Error index based ranking			Gross rank	Position of gross rank
Number of neurons in hidden layer	SSE	RMSE	MAPE	SSE	RMSE	MAPE		
5	1.809411	0.133189	94.68261	1	1	1	3	1
6	1.88605	0.13598	95.76902	2	2	3	7	2
7	32.43063	0.563868	95.01077	9	9	2	20	7
8	3.803123	0.193095	113.7054	6	6	5	17	6
9	3.456277	0.184079	116.2629	5	5	6	16	5
10	2.385958	0.152944	112.9745	3	3	4	10	3
11	2.590865	0.159376	117.404	4	4	7	15	4
12	4.093159	0.200322	138.2391	8	8	8	24	9
13	4.061389	0.199543	142.0007	7	7	9	23	8

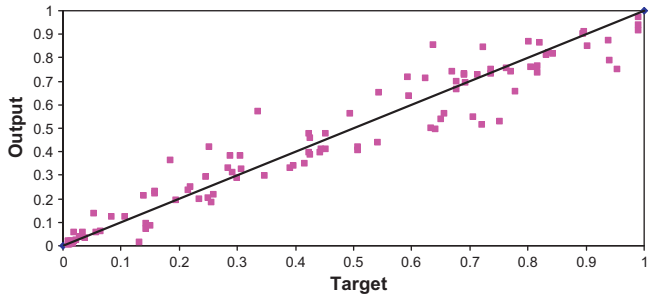

Fig. 5. Predicted vs experimental values for two-hidden layer (5 neurons) ANN model.

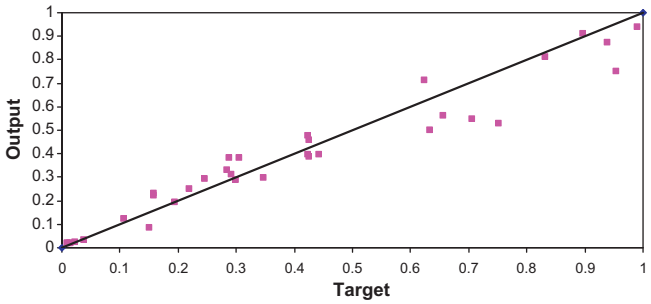

Fig. 7. Predicted (output) vs experimental (target) CH_4 yield for the optimum ANN model.

3.3. Interpretation of optimum ANN model

The weights from the optimum model (1-hidden layer: 3 neurons and 2-hidden layer: 2 neurons) were used to obtain a relationship between propane steam reforming period and CO_2 gasification period to H_2, CH_4 and CO yield by employing connection weights method (CWM) and the results are schematically represented in Fig. 9. It is evident from Fig. 9 that cycle split (≈+85%) played a more important and positive role than cycle period (≈−15%) in both H_2 and CO production. However, they have almost equal but opposite roles in

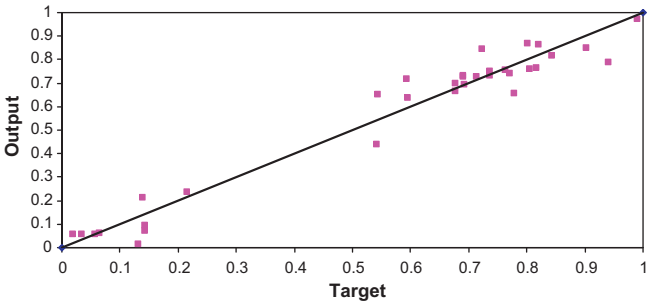

Fig. 8. Predicted (output) vs experimental (target) CO yield for the optimum ANN model.

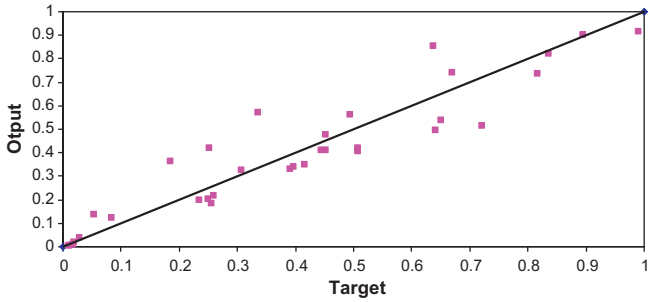

Fig. 6. Predicted (output) vs experimental (target) H_2 yield for the optimum ANN model.

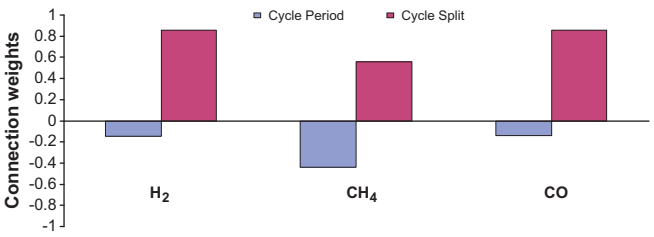

Fig. 9. Influence of cycle period and split on H_2, CH_4 and CO yield.

the determination of CH_4 yield. In particular, cycle period has only a mild detrimental effect on H_2 and CO yield. During the propane steam reforming section of the period, carbonaceous deposition simultaneously took place with H_2 and CO production via:

$$C_3H_8 + xH_2O \rightarrow xCO + \left(\frac{x^2 + 12x - 3}{2x}\right) H_2 + \left(\frac{3-x}{x}\right) C_xH_{1-x} \quad (11)$$

where the composition of the carbon residue, C_xH_{1-x}, $(0 < x < 1)$ on the catalyst surface depends on the steam:propane ratio used. However, CH_4, H_2 and CO are all produced in the CO_2-rich section of the period due to the gasification of the carbonaceous (C_xH_{1-x}) surface layer, in accordance with:

$$C_xH_{1-x} + \left(\frac{9x-1}{8}\right) CO_2 \rightarrow \left(\frac{9x-1}{4}\right) CO + \left(\frac{1-x}{4}\right) H_2 + \left(\frac{1-x}{8}\right) CH_4 \quad (12)$$

Thus, the cycle split, *s*, will have a positive influence on the synthesis rates of both H_2 and CO since they are produced in both sections of the cycle period, whereas an increase in the cycle period will increase the amount of carbon deposition during propane steam reforming and the propensity for further dehydropolycondesation of the C_xH_{1-x} species to less reactive graphitic carbon (by aging) will reduce active metal site density and hence catalyst activity. This will result in the overall decrease in CH_4, H_2 and CO yield (cf. Eq. (12)). The findings in Fig. 9 are consistent with these considerations. In particular, the ANN model may be used to predict the optimum cycle split and period by varying the numerical values of the appropriate connection weights for the individual species to achieve maximum H_2 and CO with minimum CH_4. This is the inverse-problem to the development of the neural network model and the objective of a forthcoming paper on the ANN-based optimisation of the forced cycling propane reforming reactor.

4. Conclusions

In this study a systematic approach of selecting optimum number of neurons and number of hidden layers was employed which resulted in an ANN ensemble of significant model accuracy. This study shows that even for a highly nonlinear process like propane steam reforming using periodic operation, an artificial neural network can be used for the input–output mapping to determine the underlying relationship between the parameters and the yield. This neural network learning experience can be used further to predict targets for entirely unknown data and the analysis of these targets can result in the optimization of the process.

Acknowledgements

Financial support of the Australian Research Council is gratefully acknowledged. FA thanks the KACST, Riyadh, Saudi Arabia for provision of a scholarship and study leave.

Appendix A.

See Table A1.

Table A1
Experimental data used for ANN modeling.

Propane reforming time (min)	CO_2 regeneration time (min)	$\%H_2$	$\%CH_4$	%CO
9	1	0.00549	0.004998	0.018588
7	3	0.021067	0.02545	0.047441
5	5	0.049741	0.049611	0.073705
3	7	0.137114	0.125923	0.064275
1	9	0.186695	0.170641	0.067945
18	2	0.002548	0.00217	0.012141
14	6	0.012411	0.017563	0.047144
10	10	0.068403	0.074939	0.060529
6	14	0.092105	0.116938	0.059046
2	18	0.177772	0.167793	0.051889
27	3	0.002364	0.000833	0.005421
21	9	0.069346	0.03331	0.011938
15	15	0.141024	0.078164	0.062252
9	21	0.181516	0.134048	0.051759
3	27	0.277339	0.177188	0.05919
36	4	0.002003	0.000447	0.004709
28	12	0.063398	0.026837	0.012165
20	20	0.12514	0.075087	0.066714
12	28	0.201335	0.112915	0.060256
4	36	0.250274	0.160534	0.064235
45	5	0.000431	8.6E−05	0.002695
35	15	0.06794	0.02686	0.01115
25	25	0.122886	0.050432	0.070329
15	35	0.178591	0.074724	0.060296
5	45	0.233703	0.148514	0.069956
54	6	8.31E−05	7.61E−06	0.001387
42	18	0.070614	0.037756	0.000542
30	30	0.140905	0.053361	0.071263
18	42	0.125117	0.075202	0.067289
6	54	0.228405	0.110984	0.086538
1	1	0.08445	0.042612	0.082265
2	2	0.108088	0.052359	0.063117
3	3	0.109289	0.061124	0.071712
4	4	0.114856	0.051043	0.078919

References

[1] F. Joensen, J.R. Rostrup-Nielsen, J. Power Sources 105 (2002) 195.
[2] F. Alenazey, C.B. Dave, S.S.E.H. El-Nashaie, A.A. Susu, A.A. Adesina, Catal. Commun. 10 (2008) 406.
[3] F. Alenazey, C. Dave, S.S.E.H. El-Nashaie, A.A. Susu, A.A. Adesina, Chem. Prod. Process Model. 4 (5) (2009) 6.
[4] R. Mihail, R. Paul, Chem. Eng. Sci. 34 (1979) 1058–1061.
[5] A.K. Jain, R.R. Hudgins, P.L. Silveston, Chem. Eng. Sci. 36 (1981) 231–233.
[6] A.K. Jain, R.R. Hudgins, P.L. Silveston, Chem. Eng. Sci. 36 (1981) 233–234.
[7] P.L. Silveston, R.R. Hudgins, S. Bogdashev, N. Vernijakovskaja, Yu.Sh. Matros, Chem. Eng. Sci. 49 (1994) 335–341.
[8] A.B. Bulsari, Neural Networks for Chemical Engineers, Elsevier Science, Amsterdam, 1995, p. 26 (Chapters 1–7).
[9] D. Chesterfield, A.A. Adesina, J. Chem. Eng. Jpn. 42 (2009) s185.
[10] T. Hattori, S. Kito, Catal. Today 111 (2006) 328.
[11] A. Corma, J.M. Serra, E. Argente, V. Botti, S. Valero, Chem. Phys. Chem. 3 (2002) 939–945.
[12] M. Holĕna, M. Baerns, Catal. Today 81 (2003) 485–494.
[13] T. Umegaki, A. Masuda, K. Omata, M. Yamada, Appl. Catal. A 351 (2008) 210–216.
[14] A. Tompos, J.L. Margitfalvi, E. Tfirst, L. Végvári, Appl. Catal. A 303 (2006) 72–80.
[15] A. Bejan, S. Lorente, J. Lee, J. Theo. Biol. 254 (2008) 529–540.
[16] S.D. Holdsworth, J. Food Eng. 4 (1985) 89–116.
[17] I.M. Vardavas, Ecol. Model. 48 (1989) 65–81.
[18] I. Parberry, Circuit Complexity and Neural Networks, MIT Press, Massachusetts, 1994.
[19] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd ed., Prentice Hall, New Jersey, 1999, p. 4 (Chapter 1).
[20] A.R. Barron, IEE Trans. Inf. Theory 39 (1993) 930–945.
[21] M. Zhang, Artificial Higher Order Neural Networks for Computer Science and Engineering: Trends for Emerging Applications, 1st ed., Information Science Reference, Pennsylvania, 2010.
[22] V.K. Pareek, M.P. Brungs, A.A. Adesina, R. Sharma, J. Photochem. Photobiol. A: Chem. 149 (2002) 139.
[23] G.D. Garson, AI Expert 6 (1991) 47.
[24] J. Chen, W. Chen, Inf. Sci. 178 (2008) 4560.
[25] A. Aleboyeh, M.B. Kasiri, M.E. Olya, H. Aleboyeh, Dyes Pigments 77 (2008) 288.
[26] Y. Yoon, T. Guimaraes, G. Swales, Decis. Support Syst. 11 (1994) 497.
[27] J.D. Olden, M.K. Joy, R.G. Death, Ecol. Model. 178 (2004) 389.