

# Prediction of Distillate Yields in a Visbreaking Process by Neural Networks

O. Foddi, M. Grosso, S. Tronci\*, R. Baratti

*Dipartimento di Ingegneria Chimica e Materiali, Università degli Studi di Cagliari,  
Piazza d'Armi, I-09123, Cagliari, Italy*

*\*e-mail: tronci@dicm.unica.it*

## Abstract

*An estimator which infers the distillate yields of an industrial visbreaking plant operating at the SARAS refinery is here presented. The estimator is based on a feed forward neural network, and the input data are pretreated with the Principal Component Analysis, which allows identifying the significant features in the available data. The neural model is trained, validated, and tested off-line by using plant data spanning an operating window of nine months. Results show a rather good agreement between the inferred yields and experimental measurements.*

## 1. Introduction

Visbreaking is a mild liquid phase pyrolysis of atmospheric or vacuum distillation residues of crude oils. The aim of this process is to reduce the viscosity of the residues and to significantly increase the production of lighter distillates like gas, gasoline, kerosene.

First principle modelling of visbreaking processes is very complex, time consuming and difficult in view of the many operating variables affecting the product quality properties. First principle models are actually available in literature [1-4], but they could not be adequate to use for on-line applications, where simple and compact models are preferred because, generally, they have to be implemented on DCS. Within this regard, Artificial Neural Networks (ANNs) may constitute a powerful approach to develop estimators that could be used for on-line applications. ANNs are data driven models, widely applied in process modelling and control [5-10], and they have demonstrated to be capable of successfully modelling non linear processes.

In this work, we describe the development of an ANN estimator to infer the conversion of the visbreaking unit operating at SARAS refinery located

in Sarroch (Cagliari, Italy). The goal is to estimate the distillate yields as a function of operating conditions and feed characteristics. The estimated yields may be used both to obtain the unit performances for a feedstock change, and support the operator to take a proper and prompt action when the yield quality is not at the desired value. The development of the estimator is not a trivial task, also because the considered plant is characterised by a high variability of the quality of the feedstock, which is formed by a mixture of heavy vacuum oils containing high concentration of sulphur compounds. Within this scenario, a strict control on the severity of the process is necessary in order to have the maximum recovery of distillates as the feedstock quality varies, without causing destabilizing effects on the residue (TAR). Furthermore, a correct management of the plant should assure an optimal period of life cycle maintenance. In fact, this factor often varies as a function of the production policy and of the scheduled stops of the other plants integrated with the visbreaking one.

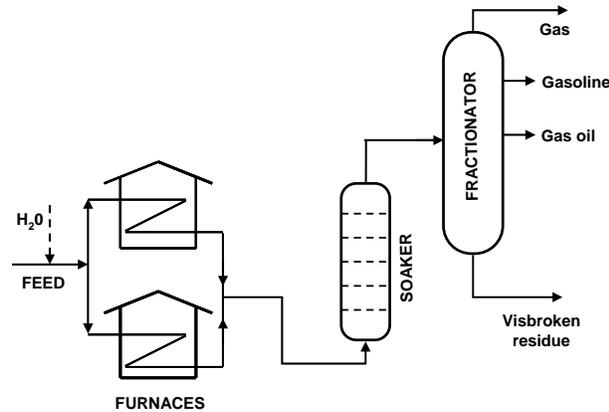
## 2. The industrial process and the experimental data

A schematic representation of the visbreaking plant operating at SARAS Refinery is schematically reported in Figure 1.

In this scheme, the mixture of vacuum oils, heated in two furnaces, is fed to an adiabatic reactor (soaker) where, mainly, the visbreaking process takes place. Each furnace is crossed by two coils that go through a convective and a radiant section. The soaker is equipped with perforated plates in order to reduce backflow and backmixing phenomena. The main purpose of the configuration "furnaces and soaker" is to increase the residence time and, consequently, reduce the average temperature of the process. In this way it is possible to decrease the fouling inside the coils and to increase the run time of the plant. The

soaker is followed by a fractionating tower where the lighter fractions are separated from TAR.

Two different streams constitute the feed to the plant: the principal one directly comes at high temperature from a vacuum unit (not reported in the Figure); the second is a cold stream, produced by another vacuum unit, which comes from stocking vessels. The hot visbroken residue is usually fed to a power plant.



**Figure 1: Scheme of the visbreaking plant**

The analysis of the process was started with a comprehensive study of the visbreaking plant, either in terms of process conditions, including the configuration and location of the available measurements, and of stream characterization, including feed and products. The analysis was aimed to find the potential variables to be used to develop the distillate yield estimator. Finally, fifteen process variables were identified as possible inputs to the soft sensor, and they are essentially of two types: 6 variables are related to the process operating conditions (temperatures, flowrates, etc.), while the remaining 9 variables concern the feed characteristics (e.g. carbon and hydrogen content). The selected variables were then collected with daily frequency for about nine months of operation. Because of a confidential agreement, no detailed indications on the variables made available for the current analysis can be reported in the present paper, and they will be denoted as  $OC_i$  ( $i = 1, \dots, 6$ ) variables when referred to the operating conditions, and  $FC_i$  ( $i = 1, \dots, 9$ ) variables when referred to the feed characteristics.

It should be noted that a reliable feedstock characterization is not available and this fact is one of the main difficulties when modelling the present process. Within this regard, these variables were derived by daily global mass balances combined with

analysis (True Boiling Point, TBP, curves) of the crude oils processed in the refinery.

The selected inputs are the candidates to infer the distillate yield, which is one of the most important indexes of the plant performance. This variable is not directly measured in the plant, but it is attained through the measurements of the volumetric flowrate of the products (gas,  $Q_G$ ; gasoline,  $Q_{GL}$ ; and gasoil,  $Q_{GO}$ ) and the feedstock flowrate,  $Q_F$ , according to the following relation:

$$Y = \frac{(Q_G + Q_{GL} + Q_{GO})}{Q_F} \quad 1.$$

Incidentally, it is worth noting that flowrate measurements are affected by an error which depends on the instrument precision. At the SARAS visbreaking plant, flowrates are measured by obstruction flow-meters. Generally this kind of sensors is characterized by small rangeability (3:1) and limited accuracy (3 - 5%). The measurement error was thus calculated by considering that each flow rate is affected by an estimated relative error  $\delta Q/Q$  equal to 4% circa. Resorting to consolidated formulas for the error propagation, one ends up with an error estimation on the experimental measure for the distillate yield equal to  $\delta Y/Y = 8\%$ .

### 3. The neural estimator

The distillate yield estimator is based on a feedforward fully connected neural network. The aim is to estimate the product yields through indirect measurements easily accessible. Soft sensors are required to be as simple and compact as possible, since they are devoted to on-line applications, and they usually have to be implemented on DCS.

The first issue of the model development is the selection of the inputs to the neural network. This is not a trivial task, but some of the following indications may help the input choice. For example, measurements with high level of noise or with significant time delays should be not considered. Moreover, a proper selection of the inputs variables will allow reducing the number of inputs and will help the soft sensor design. Within this regard, one should select out, among all the potential variables, the actual variables carrying out enough information to detect variations of the process status. Redundancy should be avoided and, quite often, lumping (grouping) more variables may improve the neural model performance. In this work, this step of the network design was carried out by applying the Principal Component Analysis (PCA) to the fifteen

candidate data inputs. This technique implies the reduction of the dimensionality of the input space, makes the data linearly independent and less affected by the experimental noise.

Another important issue, when developing neural models, is the selection of the number of hidden layers and neurons in each hidden layer. In this case, a network structure with one hidden layer was selected, while the number of hidden neurons was found with a trial and error procedure, with the final goal to obtain the simplest structure as possible.

The neurons for each layer were activated by means of a sigmoidal function, and the resulting neural model is the following

$$z_3(i) = \frac{1}{1 + \exp \left[ - \left( \sum_{j=1}^{n_2} w_2(j,i) z_2(j) + w_2(n_2+1,i) b \right) \right]} \quad 2.$$

$$z_2(i) = \frac{1}{1 + \exp \left[ - \left( \sum_{j=1}^{n_1} w_1(j,i) z_1(j) + w_1(n_1+1,i) b \right) \right]}$$

$i = 1, \dots, n_2$

where  $z_3(i)$  is the network output,  $z_1(j)$  are the neural inputs,  $b$  is the bias term set equal to +1,  $n_1$  and  $n_2$  are the number of the input and the hidden layer, respectively,  $w_1$  and  $w_2$  are the weights of the input and hidden layer of the network, respectively.

The subsequent calibration of the ANN model was accomplished in two different steps: model training and off-line validation. In particular, the total available data (233 daily averaged values) were split into a training and validation data set (170 points for training and 30 points for validation) and a test data set (the remaining 33 points).

The model calibration was accomplished by searching the minimum of the cost function mean square error giving the distance among the experimental values  $y_i$  and the predicted ones  $y_{pi}$ . The minimization of the cost function is performed using the Levenberg-Marquardt algorithm, while overtraining was avoided by means of the cross-validation. The capability of the neural model to reconstruct the distillate yields is evaluated by considering as performance indexes the determination coefficient  $R^2$  and the root mean square error  $MSE$  as defined below:

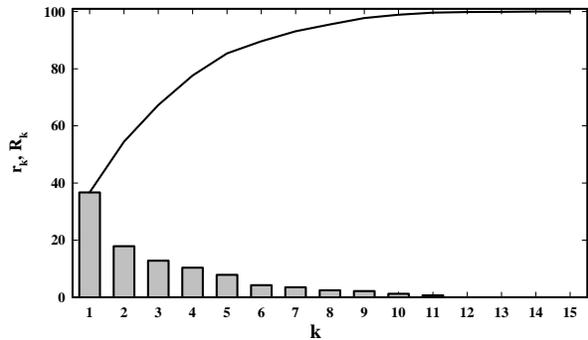
$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - y_{pi})^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad 3.$$

$$MSE = \frac{\sum_{i=1}^N (y_i - y_{pi})^2}{N} \quad 4.$$

where  $N$  is the number of experimental data and  $\bar{y}$  is their average value.

## 4. Preprocessing of data

As discussed in the previous section, the data were preprocessed by performing the Principal Component Analysis on the available measurements of the fifteen input candidates. The PCA technique is a well consolidated mathematical procedure that transforms a number of correlated variables into a smaller number of uncorrelated ones, which are the projections of the original variables onto the principal components [11]. These (linearly) independent variables are often referred as latent variables.



**Figure 2: Bars: Eigenvalues (in decreasing order) of the covariance matrix. The y-axis is the relative percentage. Solid line: Cumulative percentage of the principal components**

Let  $\mathbf{X}$  be a  $(n \times p)$  data matrix of  $n$  observations on  $p$  variables  $x_1, x_2, \dots, x_p$ . The correlation among the variables can be detected by inspection of the  $(p \times p)$  covariance matrix  $\mathbf{S}$  of the vector data. The principal components are then obtained as the eigenvectors of the covariance matrix. With this procedure one can transform interdependent coordinates into significant and independent ones. The eigenvalue associated to every principal component gives a measure of the variance explained by the single component. The principal components are then sorted in such a way that they account for the variability of the data with a decreasing order. Thus, the first principal component takes into account the largest variability of the data,

whereas the smallest principal components can be regarded as redundant information that can be dropped. The final objective is to discover common trends and to reduce the dimensionality of the observed data.

Results of the PCA procedure for the current case are reported below. In particular, Figure 2 shows the percentage of total variance explained by the single principal component (bars) and the cumulative percentage (solid line). It was found that the first 11 latent variables collect the 99.6% of the total variance of data. Thus, projecting the data onto the linear space spanned by these principal components, one can end up with 11 linearly independent variables which can be

used instead of the original data. These projected variables are usually called latent variables, and they will be denoted with  $b^j$  ( $j=1, \dots, 11$ ). In this way, one can address a reduction of the size of the data to be investigated without loss of information.

In Table 1, the loadings between the single process variable (in the  $i$ -th line) and the principal component (in the  $j$ -th column) are reported. High values imply that the  $j$ -th principal component is dominated by  $i$ -th variable behaviour. The last rows in Table are, respectively, the eigenvalues and the variance explained by the  $j$ -th principal component.

**Table 1: The Principal Component table for the process variables**

	PC <sub>1</sub>	PC <sub>2</sub>	PC <sub>3</sub>	PC <sub>4</sub>	PC <sub>5</sub>	PC <sub>6</sub>	PC <sub>7</sub>	PC <sub>8</sub>	PC <sub>9</sub>	PC <sub>10</sub>	PC <sub>11</sub>
OC <sub>1</sub>	0.2684	-0.1875	-0.3254	0.0066	-0.0802	0.4441	-0.1924	0.1434	0.6516	0.0725	-0.2994
OC <sub>2</sub>	0.2210	0.1621	0.2956	-0.1921	-0.5031	-0.1608	0.1041	-0.3017	0.3422	0.4543	0.2971
OC <sub>3</sub>	-0.0214	-0.3472	-0.1434	0.2292	0.5867	-0.0497	0.3491	-0.3620	0.2180	0.2847	0.2734
OC <sub>4</sub>	0.2100	-0.3952	-0.2774	-0.0290	-0.2781	0.1361	0.0296	0.2553	-0.1588	-0.2303	0.6827
OC <sub>5</sub>	-0.1624	0.3753	0.0133	0.0488	-0.0728	0.6197	0.6439	0.1244	-0.0700	0.0536	0.0439
OC <sub>6</sub>	0.0646	-0.3635	-0.3309	0.0907	-0.4394	-0.1916	0.4020	-0.2246	-0.2807	0.0809	-0.4645
FC <sub>1</sub>	0.3420	0.1827	-0.2036	-0.2727	0.1441	0.0374	-0.0245	-0.0470	-0.2516	0.1582	0.1100
FC <sub>2</sub>	0.2564	0.3335	-0.3597	-0.1295	0.1272	-0.2175	0.0994	0.0743	0.0244	-0.0477	-0.0187
FC <sub>3</sub>	-0.1783	-0.1376	-0.0595	-0.6855	0.1258	-0.0526	0.1417	0.0444	0.0946	0.0336	-0.0496
FC <sub>4</sub>	-0.2634	-0.2189	0.0202	-0.5382	0.0277	0.1825	-0.0253	-0.1399	-0.0616	-0.0566	-0.0470
FC <sub>5</sub>	-0.3315	0.1602	-0.2034	-0.0212	-0.1217	-0.3656	0.2509	0.0171	0.4540	-0.4707	0.1204
FC <sub>6</sub>	-0.3785	-0.0353	-0.2204	-0.0357	-0.0353	-0.0654	-0.1483	0.3370	-0.0772	0.5424	0.0457
FC <sub>7</sub>	0.3317	-0.1619	0.3370	-0.1715	0.1081	0.1343	0.1066	-0.1855	-0.0214	-0.2919	-0.0998
FC <sub>8</sub>	0.3113	0.2849	-0.3011	-0.1272	0.1547	-0.1322	0.0507	0.0011	-0.0639	0.0217	-0.0577
FC <sub>9</sub>	-0.2449	0.2078	-0.3613	0.0322	-0.1240	0.2776	-0.3509	-0.6699	-0.0937	-0.1085	0.1287
Eigen values	5.4997	2.6732	1.9186	1.5488	1.1704	0.6285	0.5279	0.3655	0.3252	0.1831	0.0989
% variance	36.66	17.82	12.79	10.33	7.80	4.19	3.52	2.44	2.17	1.22	0.66

## 5. Results

Preprocessing of the data with the PCA procedure allowed reducing the input set variables to 11 independent latent variables which are formally equivalent to the 15 original ones. A subsequent step in the neural modelling would be a proper selection of the variables relevant for the output. In fact, although the PCA procedure guarantees that the 11 latent variables collect almost the total variance of the input

data, it does not take into account the relationship between the output and the input variables. Introducing irrelevant variables in the input sets may eventually have detrimental effect on the prediction capabilities of the neural network.

To this end, a forward/backward selection of the latent variables was implemented. The variables were inserted in turn one at a time, computing 100 ANNs, and choosing each time the one leading to the most efficient ANN. The basic procedure is as follows.

First, an ANN  $Y = G(b^i)$  with only one latent variable was implemented by varying  $i$  ( $i = 1, \dots, 11$ ), and the neural model which works better in terms of performance indexes was selected. Next, a second latent variable  $b^j$  (with  $j \neq i$ ) was selected and one looks for the neural model  $Y = G(b^i, b^j)$  ( $i$  fixed and evaluated in the previous step,  $j = 1, \dots, 11, j \neq i$ ) showing the best performance. The procedure was iterated by adding further latent variables until the  $MSE$  (or, equivalently,  $R^2$ ) saturates. The procedure involves the re-examination at every stage of the influence of the inputs previously incorporated into the neural model. In fact, an input variable which may have been the best single variable to enter at an early stage may, at a later stage, be superfluous in view of the (nonlinear) relationships between it and the other variables next added as inputs to the neural model. This check was performed by using a backward procedure. However, for the current study, backward procedure evidenced that no latent variables previously included in the model revealed to be negligible in the next iterations. The performance index was always evaluated on the validation data set. It is worth noting that for each input structure, the number of hidden neurons was varied up to a maximum of two neurons. This limit depends on the small amount of available data, which required the development of a parsimonious model.

The results of the procedure are summarized in Table 2, where the  $R^2$  and  $MSE$  evaluated for the best neural model obtained at any iteration are reported. The latent variables selected as inputs to the network are also shown, and for every model the number of hidden neurons was found to be always equal to two.

The procedure applied for the selection of the neural inputs evidences that saturation of the performance indexes is reached with five inputs.

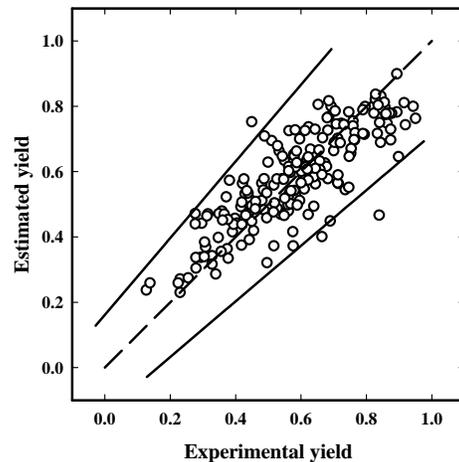
A combined analysis of Table 1 and Table 2 can give useful insights on the rule of the different variables on the performance of the neural model. In fact, the first latent variables to be selected (in more detail, the 5<sup>th</sup>, the 9<sup>th</sup> and the 2<sup>nd</sup> ones) are mainly related to the operating conditions, whereas the loadings of the feed characteristics in such principal components have a minor impact. Only at the 4<sup>th</sup> iteration the information related to the feed characteristics are taken into account by means of the first principal component. One should remark that although the first principal component collects almost 37% of the total variance of data, such information appears to be less significant for the modelling of the distillate yield. In conclusion, this result clearly indicates the significant rule played by the process

conditions and the minor importance of the feedstock characterization data.

**Table 2: Summary of the performance indexes calculated for the best neural model (validation data set) at any step and the corresponding selected latent variables.**

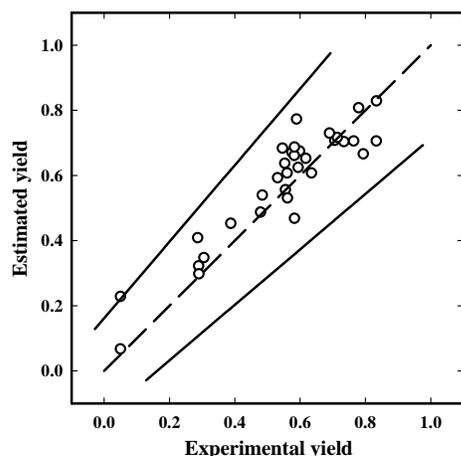
	<i>Input variables</i>	$R^2$	$MSE$
1	5	0.37	1.8e-2
2	5-9	0.56	1.6e-2
3	5-9-2	0.61	1.1e-2
4	5-9-2-1	0.67	1.0e-2
5	5-9-2-1-11	0.72	9.0e-3
6	5-9-2-1-11-4	0.74	8.7e-3
7	5-9-2-1-11-4-8	0.75	8.6e-3

In the following, the prediction capabilities of the model are illustrated. For confidential reasons, all the following figures do not report the absolute values of distillate yield, but the normalized ones. Figure 3 shows the comparison of network predicted distillate yields with the measured yields for the training and validation data set, obtained by selecting five inputs for the neural network (structure #5 in Table 2). In this case, the determination coefficient is  $R^2 = 0.70$  and the mean square error is  $MSE = 0.010$ .



**Figure 3: Comparison of the predicted distillate yields with the measured ones for the training and validation data set (structure 5 in Table 2)**

The capability of the network is then verified by processing the test data set and the results are shown in Figure 4, where the predicted yields are compared with the measured experimental ones. In this case, the performance indexes are  $R^2=0.83$  and  $MSE=0.007$ . The two solid lines in the figures indicate the error of  $\pm 8\%$  (estimated experimental error) on the measured values. It is evident that the prediction error is under the experimental one for the most of situations. Hence the results are surely satisfactory, and Figures 3 and 4 show a good agreement with an average error comparable to that of the measurements.



**Figure 4: Comparison of the predicted distillate yields with the measured ones for the test set (structure 5 in Table 2)**

## 6. Conclusions

A neural-based estimator has been developed to infer the distillate yield of an industrial visbreaking plant, owned by the SARAS refinery (Cagliari, Italy). The soft sensor was based on a feedforward fully connected neural network, with one hidden layer and sigmoidal activation functions. The data made available for this study consisted of fifteen daily averaged process variables (operating conditions and feed characteristics) collected during nine months of the plant operations. The data were preprocessed by PCA in order to reduce the redundancy in the data. The neural network inputs were then selected, among the first eleven latent variables, with a forward-backward procedure. This led to a parsimonious and efficient model constituted by five inputs, two hidden neurons and one output. The uncertainty of the estimation of the distillate yields is lower than the uncertainty assumed for the experimental measurements, hence the

results are promising. In addition, the adopted procedure evidenced a strong dependence of the distillate yield on the operating conditions, whilst the feed characteristics show a minor impact. These results suggest the possibility to use this methodology to obtain an on-line monitoring of the distillate yield, by means of real time data, and it gives also useful insights for the development of first-principle models of the investigated process.

## Acknowledgments

Authors acknowledge the personnel of SARAS refinery who made available the data used in this work.

## References:

- [1] M. Dente, G. Bozzano and G. Bussani, "The visbreaking process simulation: products amount and their properties prediction" *Comp. and Chem. Eng.*, 19, 1995, 205-S210
- [2] K.L. Kataria, R.P. Kulkarni, A.B. Pandit, J.B. Joshi and M. Kumar, "Kinetic Studies of Low Severity Visbreaking", *Ind. and Eng. Chem. Res.*, 43(6), 2004, 1373-1387
- [3] G. Bozzano, M. Dente, M. Sinatra and G. Conti, "An Advanced Model of Visbreaking Process", *Chem. Eng. Trans.* 7, 2005, 255-260
- [4] G. Bozzano, M. Dente and F. Carlucci, "The effect of naphthenic components in the visbreaking modeling", *Comp. and Chem. Eng.* 29(6), 2005, 1439-1446
- [5] E. Hernandez and Y. Arkun, "Study of the control-relevant properties of backpropagation neural network models of nonlinear dynamical systems", *Comp. and Chem. Eng.*, 16(4), 1992, 227-240
- [6] M.A. Hussain, "Review of the applications of neural networks in chemical process control - simulation and online implementation", *Artificial Intelligence in Engineering*, 13(1), 1999, 55-68
- [7] D.M. Himmelblau, "Applications of Artificial neural Networks in Chemical Engineering", *Korean J. Chem. Eng.*, 17, 2000, 373-392
- [8] S. Tronci, R. Baratti and A. Servida, "Monitoring Pollutant Emissions in a 4.8 MW power Plant through Neural Network", *Neurocomputing*, 43, 2002, 3-15
- [9] C. Kuroda and J. Kim, "Neural network modeling of temperature behavior in an exothermic polymerization process", *Neurocomputing*, 43, 2002, 77-89
- [10] G. Padmavathi, M.G. Mandan, S.P. Mitra. and K.K. Chaudhuri, "Neural modelling of Mooney viscosity of polybutadiene rubber", *Comp. and Chem. Eng.* 29(7), 2005, 1677-1685
- [11] I.T. Jolliffe, *Principal Component Analysis*, 1986, Springer, New York