NEURAL NETWORK TECHNOLOGY APPLIED TO REFINERY INFERENTIAL ANALYZER PROBLEMS

Nunzio Bonavita

ABB Elsag Bailey Hartmann & Braun S.p.A
Via Hermada, 6 - 16154 Genova Italy

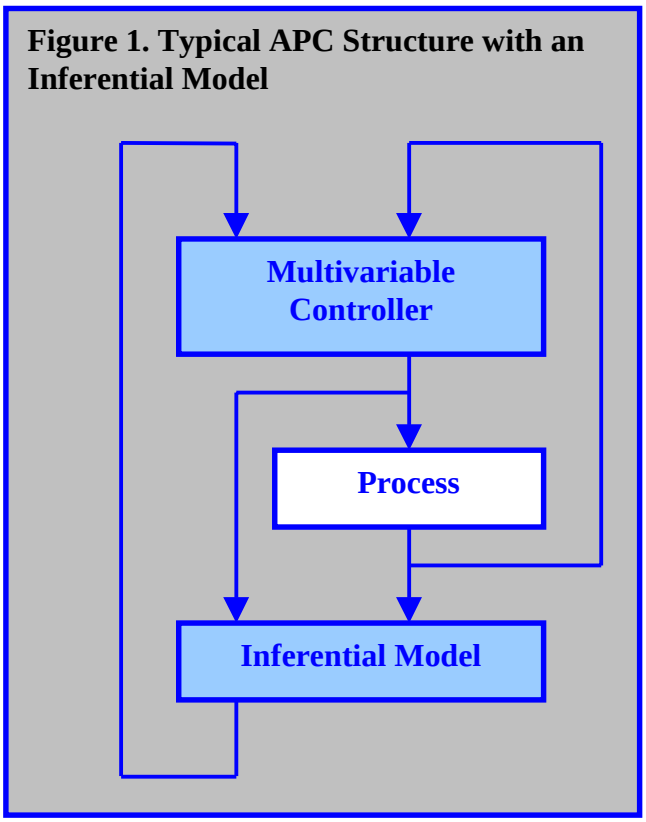
Ted Matsko

ABB Automation Inc.
29801 Euclid Ave.Wickliffe OH USA

1. Introduction

In the refining industry, process control engineers have applied computer based advanced process control (APC) for 30 years and multivariable control for 15 years. Throughout this period, these techniques have relied on modelling product quality variables from laboratory data and process measurements to provide on line feedback variables, Figure 1. Model development has become more difficult as process designers have built more complicated plants and as operating procedures have run units against their limits enlarging the working envelope the models must be accurate over.

Figure 1. Typical APC Structure with an Inferential Model



There are several factors that contribute the difficulties in process modelling. These models must accommodate deadtimes that occur in process measurements and analyzers; and they must also handle the nonlinear nature of chemical and physical relationships that govern plant behavior. Current trends in process design lead to complicated energy integration schemes, making the model development procedure still more involved.

The lower cost of distributed control systems (DCS) have allowed refiners to increase the number of measurements and actuators on the process. This allows for better models, but it also complicates the process of creating the best inferential model by allowing a multiplicity of possible variable pairings.

The integration of historians with the DCS provides a wealth of process data, but a wealth of data is not necessarily a wealth of information [1]. The operators are another source of information regarding plant behavior, but as units have become more complex

their expertise has become more anecdotal. One modelling tool can work in this data rich environment to extract information about process behavior. Neural networks have arrived in online applications in the refining, chemical, energy and paper industries.

In the applications discussed in this paper, the neural networks were built using the Process Advisor toolset from AIWare [2] and were deployed in an Infi-90 DCS from ABB Automation [3].

2. Modelling Techniques

Modelling techniques are used by engineers to develop compact mathematical expressions that describe the behavior of a process or event. These equations take values for input variables and compute resultant values for the output variables. Depending on the nature of the model, this may or may not represent a causal relationship.

There are two poles in modelling technology, the theoretical and the empirical. A theoretical model is derived from scientific principles such as conservation of mass, energy and species, and the laws of thermodynamics. An empirical model is mathematically derived from collected process data. Valid theoretical models always provide a causal relationship, while an empirical model may not. The empirical model may just imply that the same driving forces move both the input and output variables, and that they are related by the underlying theoretical model. So the user must insure that the underlying process does not change behavior if an empirical model is used.
In practice, full theoretical models are very expensive to derive and only used in full scale optimization projects. Process control applications usually employ empirical models. There are several varieties in common use based on the well known techniques of linear and nonlinear regression. Steady state linear models are represented by the following equation,

Eq. 1 $$\mathbf{y_{pred}} = \underline{\mathbf{x}} * \underline{\mathbf{w}} + \mathbf{b}$$

where $\mathbf{y_{pred}}$ is the predicted value of the product quality
$\underline{\mathbf{x}}$ is the data input vector
$\underline{\mathbf{w}}$ is the model coefficient vector
$\mathbf{b}$ is the bias coefficient

Linear regression algorithms solve the following error minimization problem:

Eq. 2 $$\min_{\underline{\mathbf{w}},\mathbf{b}} \left(\underline{\mathbf{y}} - \underline{\mathbf{X}} * \underline{\mathbf{w}} + \underline{\mathbf{1}} * \mathbf{b} \right)^2$$

where $\underline{\mathbf{y}}$ is the product quality data vector
$\underline{\mathbf{X}}$ is corresponding matrix of process data, each row is a data set
$\underline{\mathbf{w}}$ is the model coefficient vector
$\mathbf{b}$ is the bias coefficient

General steady state nonlinear models have the form

Eq. 3 $\mathbf{y_{pred}} = f(\underline{\mathbf{x}}, \underline{\mathbf{w}}, \mathbf{b})$

where f () is a nonlinear function with coefficients $\underline{\mathbf{w}}$ and $\mathbf{b}$
$\underline{\mathbf{x}}$ is the data input vector
$\mathbf{y_{pred}}$ is the predicted value of the product quality
$\underline{\mathbf{w}}$ is the model coefficient vector
$\mathbf{b}$ is the bias coefficient

Nonlinear regression algorithms solve the following problem:

Eq. 4 $\underset{\underline{\mathbf{w}},\mathbf{b}}{\text{Min}} \; \Sigma_i \left(\underline{\mathbf{y}}(i) - f(\underline{\mathbf{X}}(i,*), \underline{\mathbf{w}}, \mathbf{b}) \right)^2$

where $\underline{\mathbf{y}}$ is the product quality data vector
f () is a nonlinear function with coefficients $\underline{\mathbf{w}}$ and $\mathbf{b}$
$\underline{\mathbf{X}}(i,*)$ is the ith row of the process data matrix
$\underline{\mathbf{w}}$ is the model coefficient vector
$\mathbf{b}$ is the bias coefficient

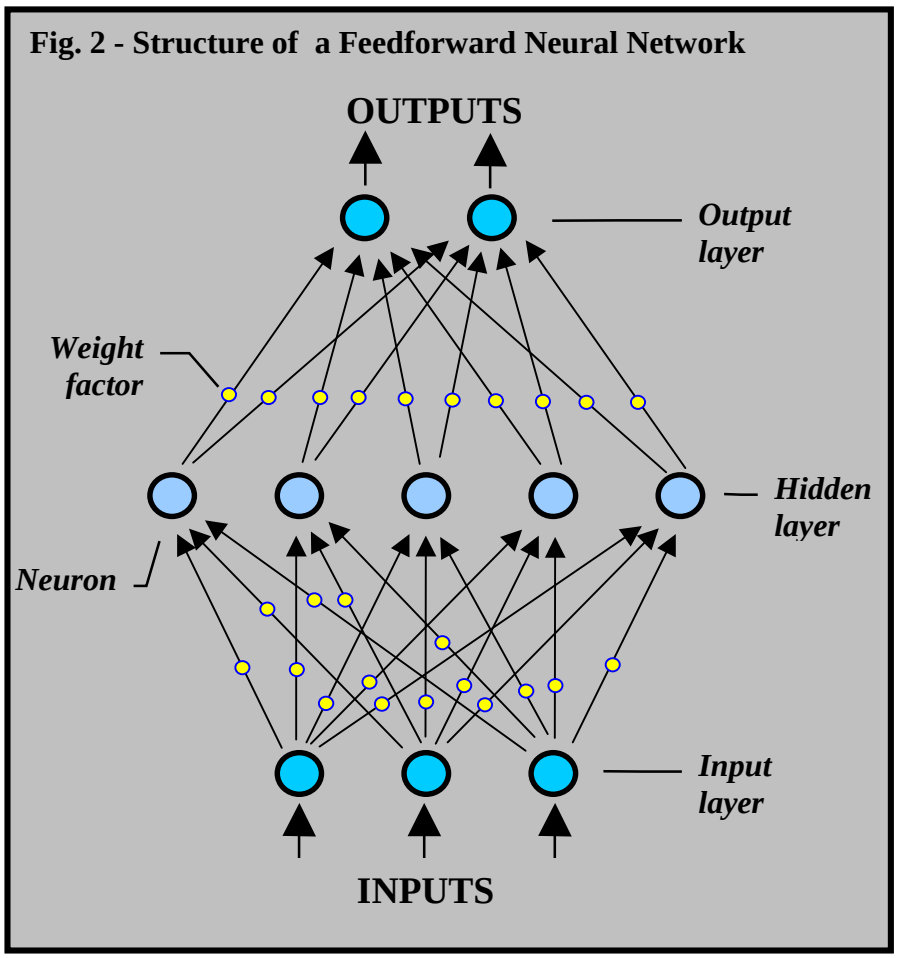
In both theoretical and regression models, the user specifies the form of the equation. For linear regression the engineer picks the variables (columns of $\underline{\mathbf{X}}$), including any transforms such as squares, cubes or logarithms of the measured variables. With nonlinear regression, the user must determine the form of f(). Neural networks, which also fall into the category of empirical models, offer some relief in this area.

3. Neural Network Technology

Neural networks are a technology that has entered the mainstream of data modelling in the last ten years. The applications are numerous and are in wide ranging fields, from economics, to pattern recognition, to chemistry, to signal processing, to fault detection. In the refining and chemical industries the following applications of interest are in operation (for examples, see [4], [5], [6] and [7]):

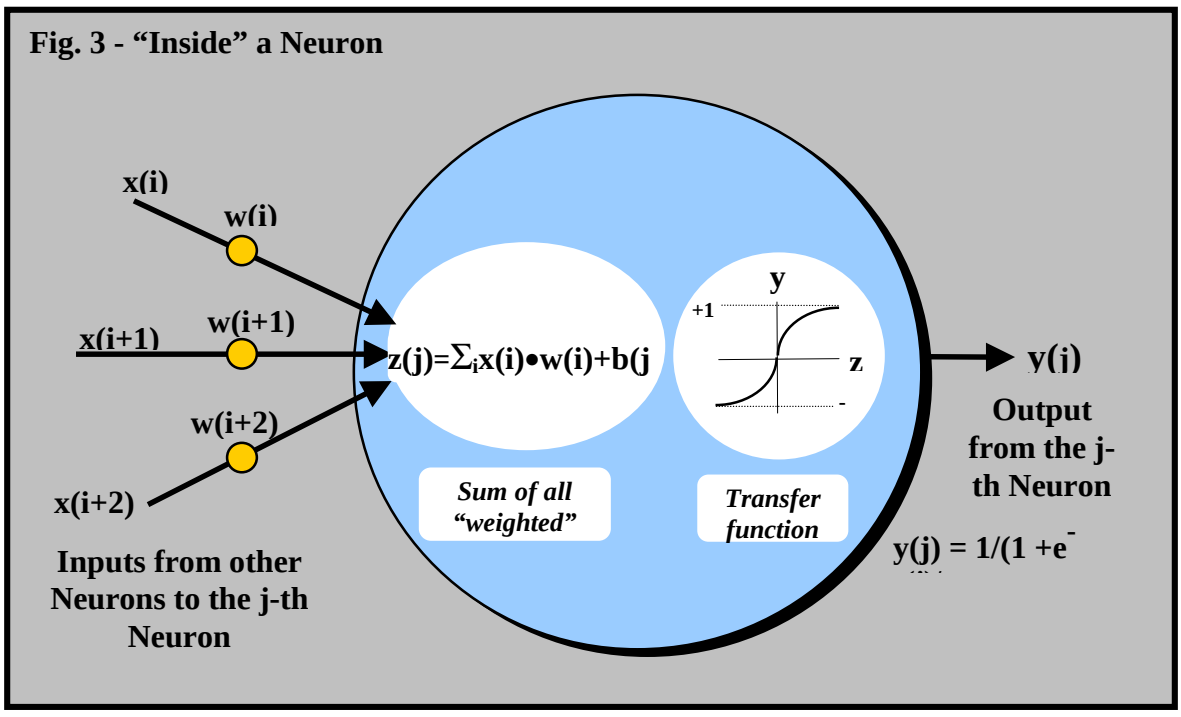
- product formulation (rubber, plastic, polymers, adhesives, gasoline, etc.)
- interpretation of complex measures (NIR spectra, etc.)
- emissions monitoring
- online process control and optimization

Neural networks are a special branch of nonlinear regression modelling. As the name implies, they are related to the human nervous system. The flow of data through the equations that make up a neural network is best described through an illustration, see Figure 2. Input data enters at the bottom layer. Each input variable is mapped to each central or “hidden” layer node, where the most important calculations take place. More than one hidden layer may be used. The outputs are then formed by combining the outputs of the last hidden layer.



The calculations for each node are defined in Figure 3. All inputs to the hidden nodes are combined in a weighted sum. Each node in each layer has a different set of input weights. These weights are analogous to the model coefficients in a regression model. The weighted sum is then used as the input to the S shaped sigmoidal function illustrated in Figure 3.

Combining all the node equations for the structure in Figure 2, we can derive a series of summation equations that relate the predicted output to the inputs. For example, take a standard three layer network with 2 inputs, 3 nodes in the hidden layer and 1 output.



The equations to compute the predicted output are

Eq. 5 $\mathbf{y}_{\mathbf{pred}} = \mathbf{y}(1,3) = \mathbf{w}(1,1,3)*\mathbf{y}(1,2) + \mathbf{w}(2,1,3)*\mathbf{y}(2,2) + \mathbf{w}(3,1,3)*\mathbf{y}(3,2)$

where $\mathbf{y}(j,k)$ is the output of the jth node in layer k
$\mathbf{w}(i,j,k)$ is the weighting coefficient for the ith input to node j in layer k

For the second or middle node in layer 2,

Eq. 6 $\mathbf{y}(2,2) = 1/(1 + e^{-\mathbf{z}(2,2)/s})$

Eq. 7 $\mathbf{z}(2,2) = \mathbf{w}(1,2,2) * \mathbf{x}(1) + \mathbf{w}(2,2,2)*\mathbf{x}(2) + \mathbf{b}(2,2)$

where $\mathbf{z}(j,k)$ is the summation term of the jth node in layer k
$\mathbf{b}(j,k)$ is the bias term for the jth node in layer k
$\mathbf{x}(j)$ is the jth input node (= $\mathbf{y}(j,1)$)
s is the sigmoidal slope

For a small network, one can compute the prediction with a calculator. For larger networks, it is very easy to covert these equations, with a few simple loops, to a computer program.

Before we can calculate the predictions, we must determine the values of the coefficients. This is done by constructing an optimization problem that is equivalent to Eq. 4.

Eq. 8 $\underset{\underline{\mathbf{w}},\underline{\mathbf{b}}}{\text{Min}} \quad \Sigma_i \left(\underline{\mathbf{y}}(i) - \mathbf{y}_{\mathbf{pred}} \left(\underline{\mathbf{X}}(i,*), \underline{\mathbf{w}}, \underline{\mathbf{b}} \right) \right)^2$

where $\underline{\mathbf{y}}$ is the product quality data vector
$\mathbf{y}_{\mathbf{pred}}(\,)$ is the predicted value of the product quality
$\underline{\mathbf{X}}(i,*)$ is the ith row of the process data matrix
$\underline{\mathbf{w}}$ is the model coefficient vector
$\underline{\mathbf{b}}$ is the bias coefficient vector

The solution to this optimization is commonly called “training” for a neural network. While nonlinear regression typically uses gradient algorithms to find the coefficients that generate the minimum error, neural network training typically computes $\underline{\mathbf{w}}$ using a family of numerical techniques known as genetic algorithms. For the given structure of neural networks, the genetic algorithms are very efficient at finding the optimum.

The structure of the neural network gives the user some advantages over the previously mentioned regression techniques. The first comes from the sigmoidal function used in each node. The shape of this function allows the network to adapt the model to the shape of the data set without explicit selection of polynomial order or logarithmic terms by the user. The second advantage is a matter of convenience. The structure of the network allows for cross terms to be examined automatically, as each node in the hidden

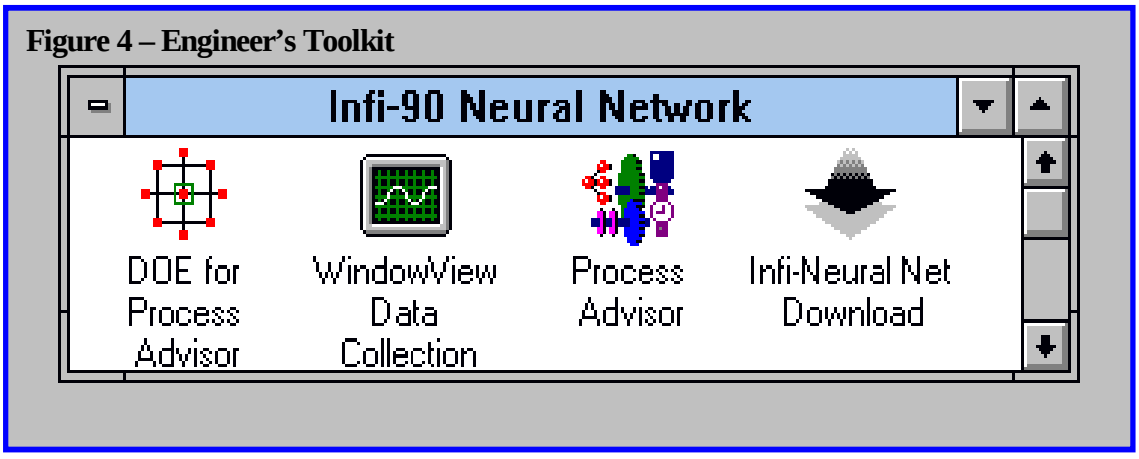
layers is connected to all input nodes in the preceding layer. With this capability for fitting data, the user must be cognizant of the potential to create a network with too many degrees of freedom (coefficients), allowing the network to overfit the data. Good neural network packages come with diagnostics to evaluate the quantitative and the qualitative measures of the model.

4. Neural Net Implementation Environments

The implementation of a neural network involves both an on-line and an off-line environment. The on-line environment is related to the equipment and interface provided for the operator. The off-line environment is related to the tool set supplied to the engineer.

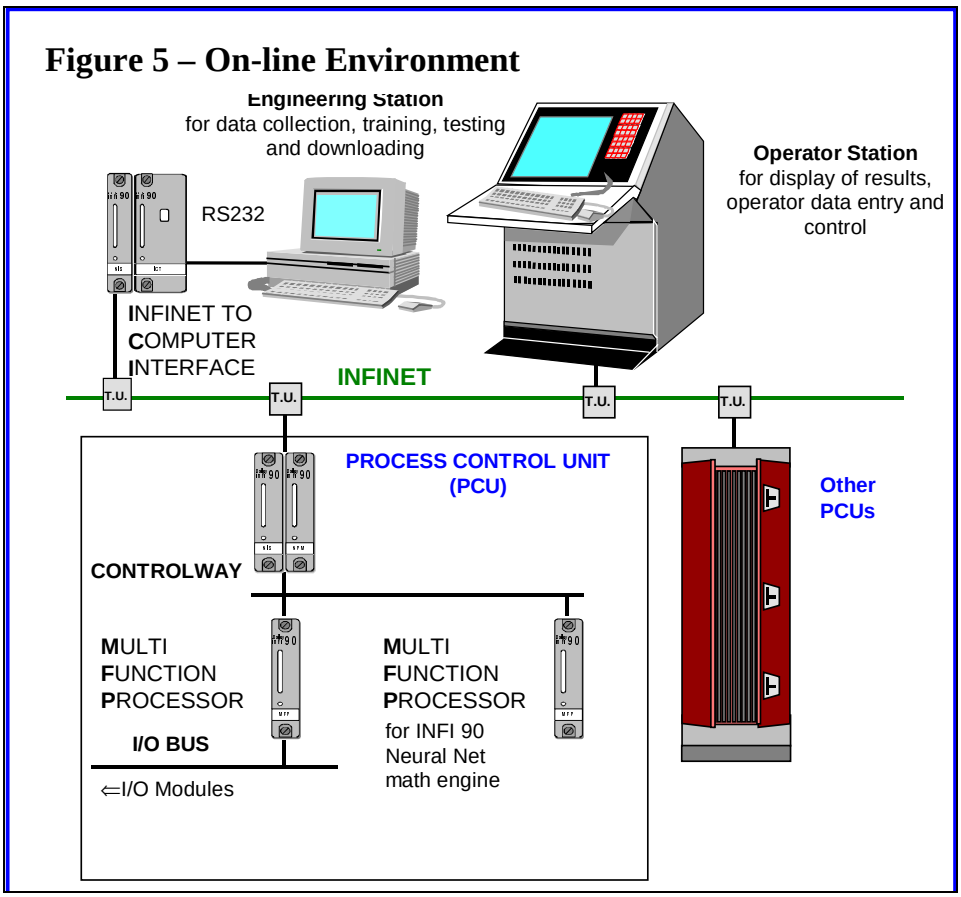
A typical set of tools is shown in Figure 4. This set consists of four programs that parallel the engineering project work. The first step is to determine how data will be collected. This may require an experimental design to maximize the utility of the data collected in the limited time the engineer has access to the plant for testing. The second tool allows the engineer to collect DCS data directly on his PC, usually a laptop carried to the plant site. The third program then imports the data and performs the data filtering, network setup, optimization and analysis to find the neural network coefficients. This is the training step, previously described. The resulting coefficients are stored in a model file, available for use by the fourth tool. The final step is loading the coefficient file, a tag database and the actual generic neural network code into the DCS. One objective of the toolset is to remove all programming requirements from the user, while providing a standard windows based GUI. The engineer concentrates on building a good model, not on writing any custom code [8].

Figure 4 – Engineer's Toolkit



The objective of the online environment is to provide a reliable, secure, well integrated hardware platform with sufficient computing power [9]. The online environment for this neural network package is the DCS, as illustrated in Figure 5. The on-line neural network engine runs in a standard DCS controller, an MFP, providing excellent integration with base and advanced controls (the multivariable controls also run in the DCS). A single engine can run multiple neural network models. Typical cycle times for the calculations vary between 5 seconds to 1 minute, depending on process dynamics. An advantage of this architecture is the that the plant operator can view the

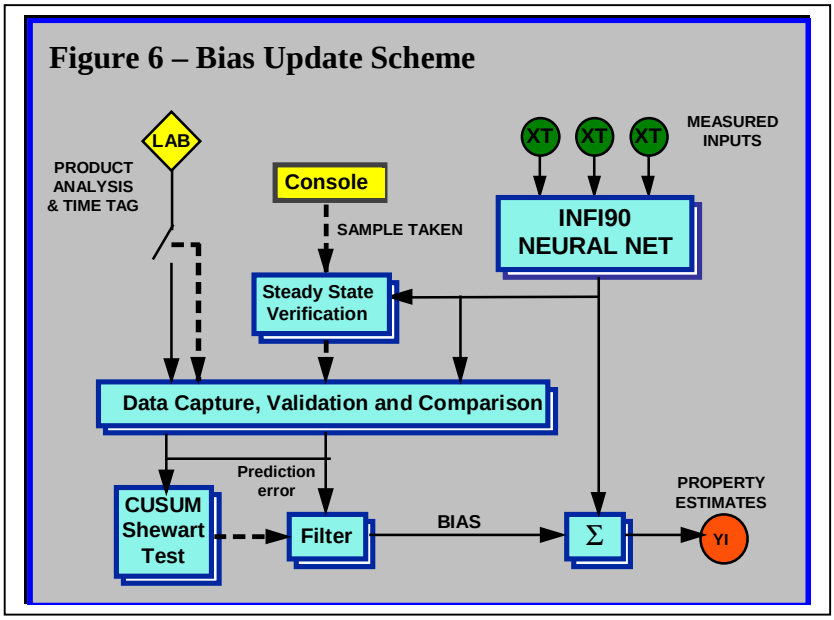
results of and interact with the inferential calculations and the advanced control algorithm at his normal console.



5. Additional On-line Considerations

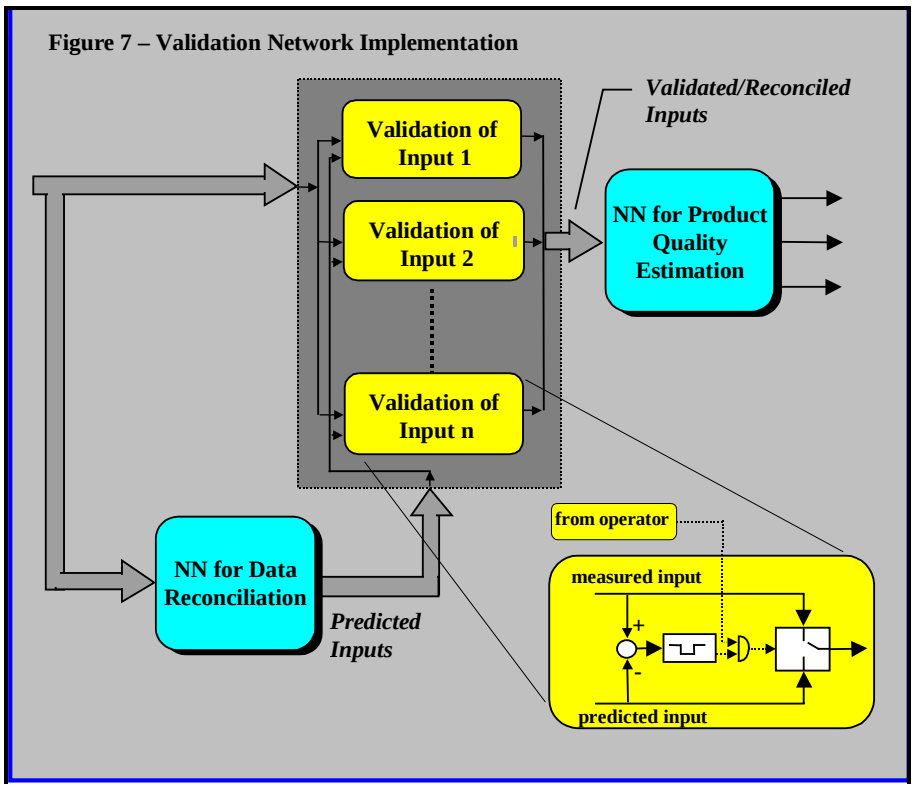
The neural network application can benefit from two additional features that are built into the on-line application, but are not part of the base inferential model.

The first is a package designed to keep the model aligned with the laboratory data as the process drifts away from the model or as unmeasured disturbance variables affect the process. This application involves feedback of the laboratory data to update the model bias. This is illustrated in Figure 6. Because lab data is a significant source of noise, we want to take precautions when injecting these values into a feedback control strategy. The simplest form of a bias update is taking the difference between the lab value and the model value as the bias, and then adding that to the model value to form the final inferential variable. To reduce unnecessary control action based on a noisy lab procedure, a direct bias update is almost never done. Instead in practice, the delta value is only accepted if the process is stable, then it is tested for statistical significance, and finally it is filtered with the old value of the bias before a change is made to the inferential value.



A second consideration for on-line operation is management of an input sensor failure. The inferential estimator will not function without all of the input data. To make the application more robust, a second neural network acts as a backup system for the input data. This auxiliary model is called the validation/reconciliation network.

The inputs to this network are the same inputs that the inferential model uses. Additional measurements that were not used in the inferential model may be added to the validation network. These additional inputs are especially useful if they are highly correlated with the regular inputs (a factor that would lead to their elimination as inputs to the inferential model). The outputs are estimates of the measured inputs to the inferential model. Inputs are not allowed to be mapped to themselves as part of their own models.



In practice, we let the plant operator use his experience and expertise as part of the system, as illustrated in Figure 7. The predicted input is compared to the actual input. A deviation results in an operator alarm. The operator may then 1.) allow substitution of the predicted value, 2.) supply a manual substitution for the value 3.) shut

the inferential calculation off 4.) allow the measurement to be used if he believes it is still valid.

6. Example Applications

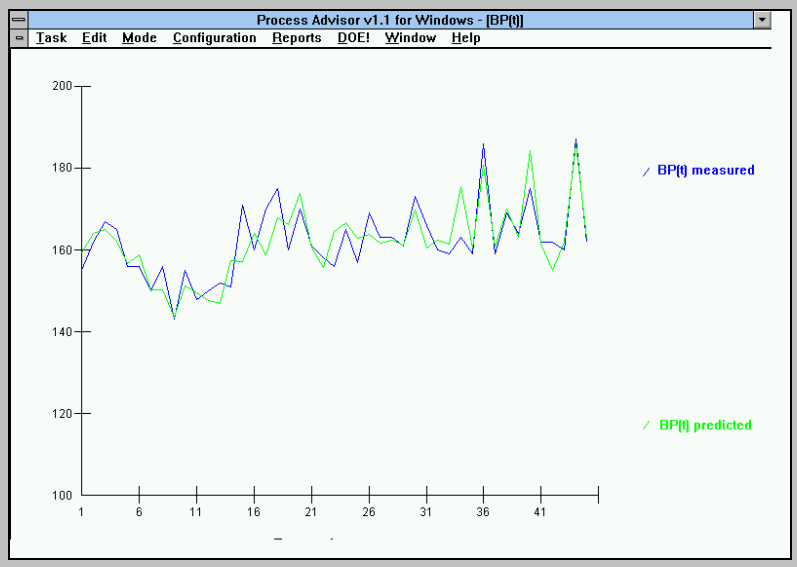
Delayed Coking

Delayed cokers provide a means of upgrading heavy crude residuals to lighter products and a solid by-product used as fuel or in chemical applications. This application involves a refinery in Italy, where the unit received feed from one crude unit's vacuum tower bottoms and a second crude unit's atmospheric bottoms [10]. The coker has a single four pass heater with two coke drums and a fractionator with overhead naphtha and two side draws for LGO and HGO. Prior to the advent of advanced controls, the operators controlled the product quality by monitoring the process temperatures of a few specific trays in the fractionator, which were augmented by laboratory feedback once per day. The economic operating objectives are to keep product quality on specification for the naphtha, LGO and HGO and to maximize the product yields by operating as close to the specifications as possible.

The closed loop advanced control of a coker is made difficult by the periodic behavior caused by switching drums. By modelling the effect of the drum switch on process variables and having a good inferential model of the product properties, the advanced control package is capable of running during the major upset caused by a drum switch. This maximizes the effectiveness of the application, creating the best economic return.

Neural network models were built for naphtha D86 95% point, LGO D86 95% point and HGO density. The models were based on 300 data records containing lab data and process data. The process data was collected as one hour averages from the data historian, synchonized to the lab samples collection time. Forty input variables were available for use. The actual number used was reduced to a minimum of 13 for the naphtha D86 95% network and a maximum of 16 for the HGO density network. The input set is reduced by checking the variables correlation with the lab data and also checking the affect on error residual.

Figure 8 – Comparison of Test Data Set to Model



We also tried to help the network by creating some computed inputs that were based on the raw process variables. The values computed were typical values traditionally used in linear regressions for fractionator inferential variables, such as flow ratios, heat exchanger duties and internal reflux. These values did not improve the models significantly, therefore the final models were built on simple inputs.

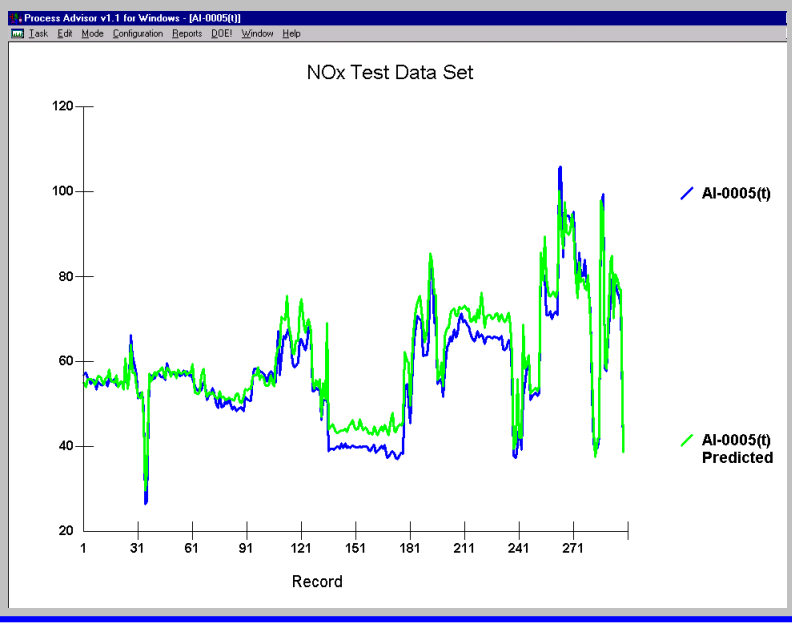
It is important to validate the model by reserving some of the data as a test set. The model is then run against this data and the residuals for the test set are compared to the residuals for the data the model is based on. If there is a significant increase, the neural network has not captured the fundamental relationship between the process data and the product quality data. Figure 8 compares the actual product quality value to the model prediction for the test data set for the naphtha 95% point. Based on the validation test, we expected good performance from this model.

The inferential analyzers have been in service for one year of successful operation. The overall application has a service factor of 95%, including furnace decoking. For naphtha, the most important quality variable, the inferential analyzer is normally within 2 to 3 °C of the lab value. The inferential calculations have achieved a high degree of trust at the refinery, resulting in decreased sampling. This has reduced the operators workload as well as freeing up a substantial block of valuable laboratory time for other areas within the refinery.

Emissions Monitoring

Neural networks can be used in utility applications within the refinery, on process heaters and steam boilers [11]. Environmental regulations are requiring increased analysis of stack emissions, as well as better reliability of these analyzers. Neural network models have been built to act as backups to real analyzers and even to completely replace these analyzers. Rules defining applicability of inferential analyzers in environmental use vary by country and locally within a country.

Figure 9 – Comparison of Test Data Set to Model



A typical application is to model NOx. This is because NOx sensors are frequently not in place, are difficult to maintain and are now required by law in many locations. Very reliable models of NOx can be built from combustion temperatures, fuel and air rates, O2 levels and recirculation gas, if any.

Figures 9 show the fit of a neural network model against the test data set for a network built as a backup to an installed analyzer.

7. Conclusion

Neural networks are a nonlinear process modelling tool available to process engineers today for use in online monitoring, control and optimization applications. The procedure is driven from data collected at the plant site and aided by software tools that relieve the engineer of programming responsibilities. As with any statistical modelling package, engineering common sense is required in data selection and model building. Good neural network models are operating online today in conjunction with multivariable control applications providing economic benefits for the operating facilities.

References

1. Denmark, K., Farren, M., Hammack, B., (1993), "*Turning Production Data into SPC Gold*", *In*Tech, December 1993

2. AI Ware Inc. (1996), "*Process Advisor - User Manual*"

3. Elsag Bailey (1997), "*Infi-Neural Net*", LAINN01/02 Product Instruction

4. Hunt, K..J., Sbarbaro, D., Zbikowski, R., Gawthrop, P.J., (1992) "*Neural Networks for Control Systems – A Survey*", Automatica, pp. 1083-1112, Vol. 28, No. 6, 1992

5. Willis, M..J., Montague, G.A., Di Massimo, C., Tham, M.T., Morris, A.J. (1992) "*Artifical Neural Networks in Process Estimation and Control*", Automatica, pp. 1083-1112, Vol. 28, No. 6, 1992

6. Ramasamy, S., Deshpande, P.B., Paxton, G.E., Hajare, R.P., (1995) "*Consider Neural Networks for Process Identification*", Hydrocarbon Processing, pp. 59-62, June 1995

7. Ramchandran, S., Rhinehart, R.R., (1995), "*Do Neural Networks Offer Something for You?*", *In*Tech, pp.59-63, November 1995

8. Beaverstock, M.C., (1993), "*It Takes Knowledge to Apply Neural Networks for Control*", pp. 335-343, Proc of ISA, 1993

9. Bonavita, N., (1997), "*Reti Neurali e Sistemi di Controllo Distribuiti*" ("Neural Technology Integrated in a Distributed Control System"), Automazione Oggi, pp. 156-164, March 1997 {*in Italian*}

10. Bonavita, N., Rinaldi, G., Matsko, T., (1998), "Strategia di Controllo Avanzato per un Impianto di Coking Ritardato", *Automazione e Strumentazione*, April 1998, pp. 119-127 {*in Italian*}

11. Samdani, G.S., (1994), "*Software Takes On Air Monitoring*", Chemical Engineering, pp. 30-33, December 1994