Journal of Engineering and Applied Sciences 3 (5): 401-415, 2008
ISSN: 1816-949X

Fault-Recovery Strategy on Critic Fault Tolerant Control Systems under FF Technology

G.R. Ferreiro and C.J.L. Rolle
Department of Industrial Engineering, University of A Coruña, ETSNyM,
Paseo de Ronda 51, 15011, A Coruña Spain

Abstract: The aim of the study deals with some aspects of the functional and hardware redundancy in fault detection, fault isolation, decision making and system recovery to solve the problem of supplying wrong information to severe or critic control systems, achieving a fault tolerant control system. To get such objectives, back-propagation neural networks are used as universal functional approximation devices which are used as residuals generators. Residuals will be evaluated by means of rule based novelty strategies in a decision-making task. Implementation procedure is carried out with the facilities supplied by a FOUNDATION™ Fieldbus compliant tool, which manage databases, neural network structures and training algorithms under an standard object oriented environment. Experimental results on a heat exchanger pilot plant are satisfactory.

Key words: Neural networks, back propagation, residual generation, fault detection, fault isolation, functional redundancy, hardware redundancy

INTRODUCTION

Products quality specifications associated to the complexity of process operation are exponentially increasing. In order to alleviate the operating requirements associated with these demands, plant health is being relayed upon the ultimate state of the art automation technology. In order to achieve required performance specifications, processes must tolerate instrumentation faults to operate fault free or safely.

Process supervision is the task responsible for correct operation by means of process monitoring tasks. The types of faults encountered in industrial applications are commonly classified into some of the following groups:

- Process parameter changes.
- Disturbance parameter changes.
- Actuator malfunctions.
- Sensor malfunctions.

The sequence of subtasks to be carried out to ensure the right process operation is the main body of process supervision usually referred to as the process monitoring tasks, which include:

- Faults detection.
- Fault identification.
- Fault diagnosis.
- Fault removing by process intervention, process recovery or process reconfiguration.

Process monitoring is based in data acquisition and data processing procedures. Process monitoring tasks can be classified into one or several following approaches:

- Data-driven.
- Analytical.
- Knowledge-based.

Data-driven: The proficiency of the data-driven, analytical and knowledge-based approaches depends on the quality and type of available models and on the quantity and quality of data available.

Principal Component Analysis (PCA) is the most widely used data-driven technique. PCA is an optimal dimensionality reduction technique in terms of capturing the variance of the data and it accounts for correlations among variables (Jackson, 1956, 1959). The structure abstracted by PCA can be useful in identifying either the variables responsible for the fault and/or the variables most affected by the fault.

Fisher Discriminant Analysis (FDA) is a dimensionality reduction technique developed and studied within

Corresponding Author: G.R Ferreiro, Department of Industrial Engineering, University of A Coruña, ETSNyM, Paseo de Ronda 51, 15011, A Coruña Spain

the pattern classification community (Duda And Hart, 1973). FDA determines the portion of observation space that is most effective in discriminating amongst several data classes. Discriminant analysis is applied to this portion of the observation space for fault diagnosis

Partial Least Squares (PLS) are data decomposition methods for maximising covariance between predictor block and predicted block for each component (Wise and Gallagher, 1996; MacGregor, 1994; Piovoso and Kosanovich, 1994; Piovoso and Kosanovich, 1992).

Analytical: Analytical methods that use residuals as features are commonly referred to as analytical redundancy methods. The residuals are the outcomes of consistency checks between the plant observations and its math-model. The residuals will be sufficiently large values under presence of faults and small or negligible in the presence of disturbances, noise and or modelling errors (Frank, 1993; Gertler, 1998; Hodouin and Makni, 1996). There are three main ways commonly used to generate residuals:

- Parameter estimation.
- Observers.
- Parity relations.

In the case of parameter estimation, the residuals are the difference between the nominal model parameters and the estimated model parameters. Deviations in the model parameters is an indication used as the basis for detecting and isolating faults (Bakiotis *et al*., 1979; Isserman, 1998, 1993; Mehra and Peschon, 1971).

In the observer-based methods, system output is reconstructed from measurements or a subset of measurements with the help of observers. The differences between actual measured output and estimated output are the residuals (Frank, 1990; Clark *et al*., 1975; Ding and Guo, 1996).

Parity relations strategy checks the consistency of process math-model (the mathematical equation of the system) with real time measurements. The parity relations are subjected to a linear dynamic transformation with the transformed residuals used in detection and isolation tasks (Gertler, 1998; Mironovski, 1979, 1980). Mentioned and commented analytical approaches require error free mathematical models in order to be effective.

Knowledge-based: Knowledge-based methods, extensively applied on process monitoring tasks include the following:

- Causal analysis.
- Expert systems.
- Pattern recognition.

This techniques are based on qualitative models, which can be obtained through causal modelling of the systems, expert knowledge, a detailed model describing the system, or fault-symptom based cases.

Causal analysis techniques are based on the causal modelling of fault-symptom relationships. Causal analysis techniques including signed directed graphs and the symptom tree are primarily used in fault diagnosis (Lee *et al*., 1999; Mo *et al*., 1997, 1998).

Expert systems are used under a human reasoning scheme (shallow-knowledge expert system). Domain experts experience can be formulated in terms of knowledge stored into a rule base, combined with first principles knowledge and applied successfully on fault diagnosis (Kramer and Finch, 1988; Li, 1989). In contrast to shallow-knowledge expert systems, deep-knowledge expert systems are based on a model such as engineering fundamentals, a structural description of the system, or a complete behavioural description of its components in faulty and normal operation conditions (Kramer and Palowitch, 1987; Kramer and Finch, 1988). More advanced expert systems using machine learning techniques (David *et al*., 2003), are advantageously used to shallow and deep knowledge expert systems, in which neural network based learning algorithms are extensively used (Bakshi and Stephanopoulos, 1994; Gao and Ovaska, 2001; Frosini and Petrecca, 2001).

Pattern recognition techniques use association between data patterns and faults classes without an explicit modelling of internal process states or structure (Xu *et al*., 2003). Artificial neural networks and sef-organizing maps based in the unsupervising learning known as Kohonen self-organising map are the main tools (Doyle *et al*., 1993).

An extensively used technique for process diagnosis based in neural networks apply the back-propagation neural network scheme (Nekovie and Sun, 1995). In this work, back propagation neural networks will be used as the main tool associated to rule based decision making strategies (Zilouchian and Bawazeer, 2001; Demuth and Beale, 1998).

None of the mentioned methods are affective (alone or individually used) in large scale systems supervision without being combined between them. Usually the best process monitoring schemes include the use of multiple methods for fault detection, identification and diagnosis (Stephen and Singh, 2003).

The aim of this work is focused on the description of the controlled system recovery methodology by using fault finding, isolation and reconfiguration tasks as part of the plant supervision, including decision making procedures according rule based techniques. To carry out proposed task, the implementation of massive neural network based models using back propagation algorithm based on collection of real-time data for a steady state operation conditions is presented (Anonymous, 2001).

The main relevant topic of the contribution in this work, is the combination of a plant recovery strategy with the utilisation of Artificial Neural Networks (ANN) technology for the inferential analysis of instrumentation performance in a wide range of industrial controlled plants. The proposed neural networks architectures can accurately predict various properties associated with plant performance behaviour. The Back-Propagation Neural Network (BPNN) is the most popular feedforward predictive network deployed in process industries. The back-propagation network assumes that all processing elements and connections are somewhat responsible for the difference of expected output and the actual output. The training algorithm is a modified iterative gradient descent algorithm designed to minimise the mean square error (RMS) between the actual output and the desired output, requiring a continuous differentiable non-linear search space called conjugate gradient method.

PROBLEM BACKGROUND AND NEURAL NETWORK BASED MODELLING

It is not common to operate with linear processes because a system is linear if all of its elements are linear and non-linear if any element is non-linear. Due to such reason industrial processes are usually non-linear. On the other hand, real lumped parameter systems doesn't exists. Process parameters usually encountered in industrial systems are generally distributed instead of lumped and finally, such systems are non-stationary, which means that its parameters are time-variant. Under this scenario any attempt to model an industrial system by analytical means could not succeed unless it will be assumed a considerable modelling error. Mentioned drawbacks could be minimised or at least slowed down by applying an alternative modelling approach under functional approximation. Functional approximation has been extensively applied in many industrial applications where it can be pointed out some recent works due to (Bawazeer, 1996; Bawazeer and Zilouchian, 1997), among other authors. Nevertheless, in this work, functional approximation is being applied exploiting its maximum modelling power to describe real time applications: Here varying time parameters of time variant systems are considered as system variables from a modelization point of view. Such modelling concept is carried out by means of conveniently trained BPNN. Under such assumption a process can be described by a set of variables classified as command inputs, disturbances, controlled outputs, internal process variables, variable parameters, constant parameters and in general all variables and parameters related by any functional dependence between them and stored into a database under some restrictive conditions.

Causal processes can be modelled by means of universal functional approximation devices. A modelling property of causality is used in this work, to predict not only steady state process input-output relationships but transient state dynamics also.

In order to reaffirm the concept of neural network based modelling (NNBM), let us consider a causal process being described by a functional approximation procedure where V_1 is the output variable, V_2, V_3, ... V_N are input variables including its derivatives and P_1, P_2, ... P_J are process parameters. Under such notation, the following transient state inputs/output relationship may be expressed for every sample cycle as:

$$V_1 = f(V_2, V_3, \cdots V_N, P_1, P_2, \cdots P_M) \quad (1)$$

Given a database containing causal data supplied from the process defined by Eq. (1), following relationships can be stated as output predictions according the following expressions:

$$\begin{aligned} V_1 &= f(V_2, V_3, \cdots V_N, P_1, P_2, \cdots P_J), \\ V_2 &= f(V_1, V_3, \cdots V_N, P_1, P_2, \cdots P_J), \\ P_1 &= f(V_1, V_2, \cdots V_N, P_2, P_J), \\ P_J &= f(V_1, V_2, \cdots V_N, P_1, P_2) \end{aligned} \quad (2)$$

where, $V_1 = f(V_2, V_3, V_N, P_1, P_2, \ldots P_J)$ in (2) is a Direct Model Predictor (DMP) and any other functional relation in (2) are Inverse Model Predictors (IMP).

For the common case of constant parameters Eq. (2) yields:

$$\begin{aligned} V_1 &= f(V_2, V_3, \cdots V_N), \\ V_2 &= f(V_1, V_3, \cdots V_N), \\ &\vdots \\ V_N &= f(V_1, V_2, \cdots V_{N-1}) \end{aligned} \quad (3)$$

Neural networks will not be an accurate predictor, if operating input/output data are outside their training data range. Therefore, the training data set should possess sufficient operational range including the maximum and minimum values for both inputs/output variables.

Every sampled data set, in order to be acquired and stored into a database, must satisfy the condition of functional dependency representing the real-time dynamic behaviour. For the case of deficient information in quantity or quality, there are some alternatives (Roelof and Pedrycz, 2003), not considered in this work. In order to ensure such condition a signal conditioning task by proper filtering is to be carried out. Such signal-conditioning task requires that every variable would be enabled to enter the database when all inputs/output variables satisfy the condition of being acquired into the same sample cycle. If one and only one data point fail entering the database, then all data set is eliminated.

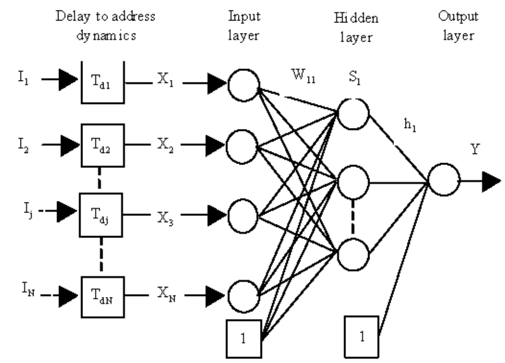

Fig. 1: Three-layer feedforward NN

NN based functional approximation: Neural Networks (NN) are essentially nonlinear function approximators that utilize process inputs to predict process output. The technical promise of neural-network technology comes from the fact that universal approximators are created using a multi-layer network with a single hidden layer that can approximate any continuous function to any desired degree of accuracy.

Soft sensors that utilize NNs must be adapted to the special requirements of the process industry. In particular, it is necessary to compensate for the delay in the process output for changes in upstream conditions (Mehta *et al.*, 2001; Tzovla and Mehta, 2001; Ganesamoorthi *et al.*, 2000). Thus, a NN typically has one output (the predicted variable) and any number of upstream measurements as inputs with compensation of process delay. Figure 1 shows a 3-layer feedforward NN.

In this example, the NN supports n inputs, a single layer of hidden neurons, bias nodes at the input and hidden layers and one output. Each input is delayed by a certain amount to allow the value used in the network to be time coincident. The weight w_{ij} connects the ith node in the previous layer to the j^{th} node in the next layer. Weighted values are summed at the node before being passed through an activation function.

For a sigmoidal hidden neuron, the summed node input S_j and the output h_j, are given by as:

$$h_j = \frac{1-e^{-S_j}}{1+e^{-S_j}} \text{ where, } S_j = \sum_i w_{ij} \cdot x_i \qquad (4)$$

Typically the output layer has a linear activation function, that is, a summation of the inputs to the output layer.

The structure of a NN soft sensor is relatively simple and is implemented in real-time controllers with little processor loading (Bawazeer and Zilouchian, 1997; Terrence *et al.*, 2003). The real challenges in the design and implementation of tools used to create a working NN may be summarized as: Collecting historic data on the process inputs and the process output measurement for screening of the potential inputs to the NN. Also, this data is needed to determine the NN structure and value of parameters used in the NN:

- Identifying the delay between each input and its impact on the process output predicted by the NN.
- Determining which of the process inputs significantly affects the process output through a calculation of input sensitivity.
- Determining the weighting factors and number of neurons included in the hidden layers for best results.
- Validation of the network.

Data acquisition: The data collection is by far one of the most critical steps in the development of a NN. When collecting data to be used to train the NN, it is important that the inputs and outputs vary over their normal operating range. If the process output is available only as a lab analysis, then this data must be merged with historical data on the input measurements to allow further analysis. Some simple rules that should be observed in the data collection are:

- Only if the inputs change during the time that data is collected will it have an impact on the process output and thus be identified in training. Where possible, a uniform number of sample values should be collected over the operating range of that input.
- Provisions should be made to automatically exclude values that fall outside this normal range of operation. Statistical techniques often prove useful in defining the outlier boundaries. A good rule of thumb for valid data limits is Mean +/-3.5 * Standard deviation, which includes approx. 99.9% of the data in the given region.

Also, it is important that the user be able to flag regions of the historic data to indicate periods of abnormal operation so this data, for all inputs and outputs, will not be used in the development of the network.

Determining input delay: A change in an upstream measurement may not immediately be reflected in a process output. Identification of this delay is the first step in the creation of the NN. The overall objective is to determine how much each process input needs to be delayed to allow best alignment with the output response. This is done automatically by calculating the cross correlation between each upstream measurement value selected as NN inputs and the process output value selected as the neural-network output. The cross-correlation, is calculated for a process input and a process output.

The time shift, K, between the input and the output that produces the maximum cross-correlation coefficient is used as the input delay that should be introduced into the input processing of the NN.

The cross-correlation value indicates the magnitude (and sign) of the effect of the input on the output. For example, for a simple first-order process, input at delay that equals approximately (dead time + time constant/2) has most relevance, as the maximum correlation value occurs at that delay. Once the most significant delay is known, the input data is shifted by that delay.

Estimation of input sensitivity: In the initial definition of the NN, it may not be possible to know which of the upstream measurements influence the process output to be predicted by the soft sensor. Only those that have a significant impact should be included in the network. The sensitivity is defined as the change in dependent variable (output) y for a unit change in independent variable (input) x, or, mathematically:

$$S_x^{\,y} = \frac{\Delta y / y}{\Delta x / x} \tag{5}$$

The initial sensitivity estimate is calculated from a simplified linear model, prior to developing the NN model. A linear model for computing the sensitivities may be obtained using the standard PLS algorithm, for example. The delayed input values and process output are used in the development of this model. Using this model, the input sensitivity is calculated by changing the input by unit while all other inputs are kept constant and determining the output change.

In a multivariable system, a higher value of sensitivity indicates that change in that input has higher influence on the output. The sum total of all sensitivities is normalized to 1. Information on the sensitivity of the inputs to the output indicates their relative importance. The sensitivity value at the delay identified in the previous step is used to exclude inputs that the output shows little or no dependence on. One technique is to exclude inputs whose individual sensitivities are small compared to the average sensitivity.

Achieving input weights: Having determined the inputs and delays to be used in the NN, it is now possible to determine the weighting factors and number of neurons included in the hidden layers to provide the best results. The weights in the network are initialized with random small non-zero values. Randomness insures lack of bias, while small values give more freedom for modifying the weights to avoid saturation. For a given number of neurons in the hidden node, the squared error between the calculated soft sensor output and the actual output measurement for one point in time is expressed as:

$$E_p = (y_p^{\,pred} - y_p)^2 \tag{6}$$

The cumulative error for set of data is calculated as follows:

$$E = \sum_{p=1}^{data} E_p \tag{7}$$

By incrementing the weighing factor W_{ij} associated with input i and node j on the value ó ∂W_{ij} and observing change in cumulative error E yields the gradient as:

$$\frac{\partial E}{\partial W_{ij}}$$

or in final differences

$$\frac{\Delta E}{\Delta W_{ij}}$$

Gradient defines change on the output for a unit change of the weight Wij. The back propagation algorithm (8) is used to calculate new weights that minimize the cumulative error in the direction of negative gradient (steepest descents) for each pass through the data set (an epoch):

$$W_{ij}^{\,new} = W_{ij}^{\,old} - \alpha \frac{\partial E}{\partial W_{ij}} \tag{8}$$

where, α defines the step size of change in the gradient direction, more popularly known as the learning rate. Instead of using the gradient descent method for error back propagation, which suffers from a number of problems, such as slow convergence and fixed learning rate DeltaV Neural uses a modified algorithm called conjugate gradient method. The speed of convergence is improved by modifying the back propagation algorithm to incorporate the conjugate gradient technique. Rather than use a fixed step size, the new direction is based on a component of the previous direction. To avoid settling on a local minima, the training algorithm is designed to automatically turn the direction remembrance on and off depending on whether the error is improving or not. Each time this is done, it results in starting with a brand-new direction. One complete pass through the data set is known as an epoch.

During the training of the NN, the cumulative error for the training set of data will decrease monotonically, approaching a constant value in an asymptotic manner. However, if the cumulative error is calculated using a validation data set not used in training, then at some point the error will begin lo increase. Past this point where the training and validation error begin to diverge, the neural nets start learning features specific to the training data set rather than the general process. The goal of training, though, is to learn to predict the output given real process inputs and not just to memorize the training set. This is known as generalization.

To detect over-training, a certain portion of the data set is kept aside for validation during the training phase. This is called the test set. At each epoch, while the weights are modified based on the error on the training set, the test set is used to detect when over fitting of the training set has occurred. At each epoch, the error on the test set (test error) for the new set of weights is calculated and compared with the best test error. In order lo prevent training from stopping at some random choice of weights for which the test error turns out to be small, the algorithm runs for at least a fixed number of epochs before establishing any minima. Also, the training error is added to the test error to define a stringent minimum total error condition. In this way, it is made sure that both the data sets have acceptable errors when the algorithm converges and the weights at the best epoch are picked at the end of training.

SEARCHING FOR THE NUMBER OF NODES IN THE HIDDEN LAYER

The number of nodes in the hidden layer has a large impact on the accuracy of the function approximator or soft sensor. In general, a poor fit is achieved with a smaller number, while a larger number may lead to over fitting the training set. To determine the optimum number of hidden nodes, it is possible to train the network starting with one hidden node up to the maximum number of nodes, which is the number of inputs to the NN. For each increment, the minimum cumulative error is stored. If the difference between errors for different numbers of hidden nodes is within a tolerance level, the NN with a smaller number of hidden nodes is given preference. The algorithm then automatically picks the weights for the best combination of train/test error obtained. In this manner, the generalization of the network is maintained while exploring for the best possible network configuration.

Updating weights due to process changes: The property of a process output stream predicted using a NN and measured upstream conditions is automatically corrected for error introduced by unmeasured disturbances and measurement drift. This correction factor is calculated based on a continuous measurement or sampled measurement of the stream provided by an analyzer or lab analysis of a grab sample. An adaptive correction of the NN is created by incorporating this automatically generated correction factor as an integral part of the neural-network algorithm.

Two approaches are used to calculate the correction factor that must be applied to the NN prediction. Both are based on calculation of the prediction error using the time-coincident difference between the uncorrected predicted value and the corresponding measurement value. Depending on the source of the error, a bias or a gain change in the predicted value is appropriate. To avoid making corrections based on noise or short-term variations in the process, the calculated correction factor should be limited and heavily filtered; for example, equal to twice the response-time horizon for a change in a process input. During those times when a new process output measurement is not available, the last filtered correction factor is maintained.

An indication is provided if the correction factor is at the limit value. Also, a configurable filter on the corrected prediction value allows noise in the input measurements to be filtered. The basic implementation is shown in Fig. 2.

The fact of incorporate the adaptive correction as part of the NN, simplifies implementation and dramatically improves neural-network online performance.

State-of-the-art implementation: The following features are normally supported by a control systems based on NN functional approximation or soft sensors:

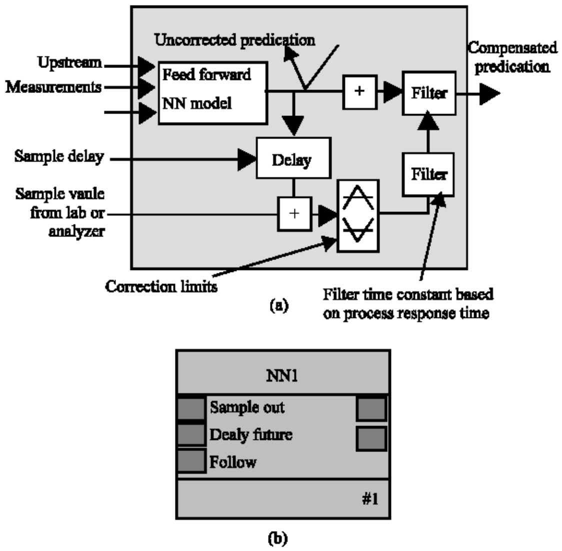

Fig. 2: (a) Adaptive NN Structure (b) NN Function block symbol

- Feedforward NN-based intelligent sensors that are tightly integrated with scalable process control system.
- The applied algorithms are based on the back-propagation technique with significant modifications for use in process industries.
- Statistical pre-processing techniques remove data unrepresentative of the general region of operation.
- Tools allow us to automatically realize a suitable input-output configuration for the data set.
- For all inputs, the input is time shifted through the use of cross-correlation values between the output and input, to account for dead time in the neural-network model.
- Sensitivity analysis determines the relative importance of the various inputs.
- Conjugate-gradient back-propagation training with direction remembrance and optimal learning rate calculation, is used to realize a robust NN-based identifier.
- Network model predictions are validated against actual data.
- In online mode, the actual process variables, for example those obtained as a result of lab analysis, are used as inputs to the NN for automatic adaptation of its prediction in response to changes in process.
- The NN function block is imbedded in the process control system to simplify implementation and improve reliability.
- An intuitive and user friendly GUI minimizes the engineering and implementation effort while maintaining the underlying NN technology.
- A 2 stage preprocessing algorithm for identification of process input analysis.
- Automatic determination of the number of hidden nodes.

FAULT DETECTION AND ISOLATION STRATEGY

The performance of all input/output devices of a multivariable severe control system is of critic relevance. For that reason redundancy is a common alternative to fault tolerant control systems monitoring. Consequently, proposed strategy concerns to both aspects of redundancy combined between them as required:

- Functional redundancy.
- Hardware redundancy.

Functional redundancy deals with two or more functions describing the same process (Deckert *et al*., 1977), while hardware redundancy is referred to 2 or more hardware devices applied in measuring the same variable.

Supervision task is being carried out in two phases: fault detection and fault isolation. Depending on process characteristics there will be necessary to propose functional and hardware redundancy.

Fault detection is inferred by evaluating functions achieved by functional redundancy with parity relations.

Fault isolation is inferred by logic evaluation of hardware redundancy with parity relations on pairs of devices, which means that fault isolation concerns to discrimination of a faulty sensor by means of a novel method. The main objective in applying functional redundancy is to detect and isolate the group of devices that fails. So that, in order to ensure the dynamic equilibrium, action/reaction forces inherent to dynamic processes are balanced by functional approximation based models according NNBM.

Given a general dynamic process modelled by means of functional approximation procedures under NNBM1, NNBM2 NNBM3 and NNBM4 where Y_A and Z_R are action and reaction functions, Y and Z are NNBM outputs of the action/reaction functions, Y' and Z' are redundant NNBM outputs of Y and Z, it follows that:

$$\begin{aligned} &\text{DMP1}: Y = f(Y_1, Y_2, \ldots Y_N) \\ &\text{DMP2}: Y' = f(Y_1', Y_2', \ldots Y_N') \\ &\text{DMP3}: Z = f(Z_1, Z_2, \ldots Z_N) \\ &\text{DMP4}: Z' = f(Z_1', Z_2', \ldots Z_N') \end{aligned} \tag{9}$$

where,

$Y_1, Y_2,..Y_N$ = are inputs from measuring devices to DMP1. $Y_1', Y_2',..Y_N'$ are inputs from redundant measuring devices to DMP2.

$Z_1, Z_2,..Z_N$ = are inputs from measuring devices to DMP3.

$Z_1', Z_2',..Z_N'$ = are inputs from redundant hardware devices to DMP4.

Given a dynamic process where an input or action force Y_A (manipulated variable) is modelled as $Y = f(Y_1, Y_A,...Y_N)$, the output or reaction force Z_R is modelled as $Z = f(Z_1, Z_A,...Z_N)$, which is a function of process variables, then, the condition for dynamic equilibrium requires the assumption:

$$Y_A = Z_R \tag{10}$$

In order to establish reasoning bases regarding devices performance, following propositions are considered:

The condition for functional redundancy between groups of devices requires the existence of instrumentation groups modelled such that rigorously $Y' = Y$, $Z' = Z$, equations which in practice are relaxed to the approach:

$$Y' \cong Y, \quad Z' \cong Z \tag{11}$$

The condition for the existence of hardware redundancy requires:

$$\begin{aligned} &Y_1 \cong Y_1', Y_2 \cong Y_2', ...Y_N \cong Y_N' \\ &Z_1 \cong Z_1', Z_2 \cong Z_2', ...Z_N \cong Z_N' \end{aligned} \tag{12}$$

A necessary condition but not sufficient to confirm the correct operation of instrumentation is the correctness of the involved NNBMs, which means the absence of modelling errors in the functional approximation devices. Under the necessary condition consisting in the absence of modelling errors and assuming that $Y \cong Z$ and that only irrelevant short periods of time $Y \neq Z$, then it is admitted that both main groups of devices operate correctly with an exception. Furthermore, if $Y' \cong Z'$ and that only irrelevant short periods of time $Y' \neq Z'$, then it is admitted that both groups of redundant devices operate correctly with an exception. Consequently, if $Y \cong Z$, $Y \cong Y'$ and $Z \cong Z'$ then the redundant groups of devices Y' and Z' operate correctly because $Y \cong Z'$ and $Y' \cong Z$. The mentioned exception concerns to the possibility of collapse of all devices in both groups. In such a case, then $Y = Z = 0$, $Y' = Z' = 0$.

Proof: given $Y \cong Z$, $Y \cong Y'$ and $Z \cong Z'$ then $Y' \cong Z'$ which is the balance asseveration between Y_A and Z_R.

Theorem 1: Under the assumption of $Y \neq Z$, at least one of both groups of devices of measuring system fails.

Proof: $Y_A = Z_R$, that means dynamic equilibrium must be balanced or dynamic balance cannot be violated. Consequently, if no fault exists, $Y \cong Z$. So hat, if processing system (NNBM) do not fail, then data acquisition system (measuring devices of Y, Z or both) fails. Consequently from theorem 1 follows that If $Y' \neq Z'$ $\Rightarrow$at least one of both groups of redundant measuring system fails. Furthermore, if $Y' \neq Z'$ and $Y \neq Z$ $\Rightarrow$ at least one of the main groups and one of its redundant groups of measuring devices, is faulty.

Individually faulty groups isolation is carried out by functional redundant analysis of residuals applied on all groups of measuring devices. In the task of faulty groups isolation, the following theorem is to be proposed and applied.

Theorem 2: Any residual R_{ij} approaching null value, guarantee the correct operation of both groups of devices involved in such residual.

Proof: $Y \cong Z$ is a guarantee of correctness measuring instrumentation groups Y and Z. So that, if $Y \cong Z$ then $R \cong Y\text{-}Z \cong 0$.

As consequence of theorem 2 it can be stated that when comparing three groups of devices G_1, G_2 and G_3, the group of devices that fails is the one excluded from the two groups that approaches null value. According last proposition it follows that given the groups of devices Y, Y', Z and Z', where Y' and Z' are redundant groups of Y and Z, respectively, yields the faulty group as:

$$\begin{aligned} &G_1 \Leftarrow R_{YY'} \wedge R_{YZ} \wedge \bar{R}_{Y'Z} \\ &G_2 \Leftarrow R_{YY'} \wedge R_{Y'Z} \wedge \bar{R}_{YZ} \\ &G_3 \Leftarrow R_{YZ} \wedge R_{Y'Z} \wedge \bar{R}_{YY'} \end{aligned} \tag{13}$$

where,

R_{IJ} = Are the residuals achieved by parity relations applied by means of functional redundancy and the symbol.

$\wedge$ = Is a logic and operator.

So that, applying logical evaluation of achieved residuals by means of the rule based procedure shown by Eq. (8), faults detection and isolation at group's level is being carried out. The meaning of Eq. (8) is illustrated by means of Fig. 3.

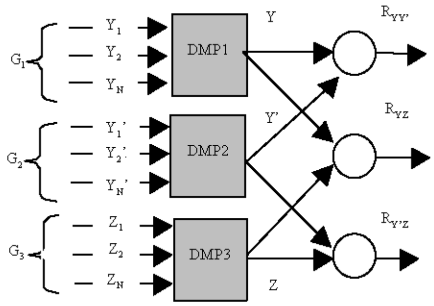

Fig. 3: Fault detection and isolation between redundant groups of devices

Using an alternative redundant group, individual faulty groups isolation is completed under the same reasoning base:

$$\begin{aligned} G_2 &\Leftarrow R_{Y'Z} \wedge R_{Y'Z'} \wedge \bar{R}_{ZZ'} \\ G_3 &\Leftarrow R_{Y'Z} \wedge R_{Z'Z} \wedge \bar{R}_{Y'Z'} \\ G_4 &\Leftarrow R_{ZZ'} \wedge R_{Z'Y'} \wedge \bar{R}_{ZY'} \end{aligned} \tag{14}$$

The meaning of Eq. (14) is illustrated in the Fig. 4.

Using the simplified combination of both main and its redundant groups, yields:

$$\begin{aligned} G_1 &\Leftarrow R_{YY'} \wedge R_{YZ} \wedge \bar{R}_{Y'Z} \\ G_2 &\Leftarrow R_{YY'} \wedge R_{Y'Z} \wedge \bar{R}_{YZ} \\ G_3 &\Leftarrow R_{YZ} \wedge R_{Y'Z} \wedge \bar{R}_{YY'} \\ G_4 &\Leftarrow R_{Y'Z'} \wedge R_{ZZ'} \wedge \bar{R}_{Y'Z} \end{aligned} \tag{15}$$

The meaning of such asseveration concluded by the Eq. (15) is depicted by Fig. 5.

Isolation of a faulty device: Nevertheless, fault isolation at device level requires to add a step more which consists in exploit the concept of hardware redundancy, where the faulty device is isolated by the following rule-based inferential procedure:

$$\begin{aligned} Y_1 &\Leftarrow G_1 \wedge R_{Y1} \\ Y_2 &\Leftarrow G_1 \wedge R_{Y2} \\ &\vdots \\ Y_n &\Leftarrow G_1 \wedge R_{Yn} \\ Y_1' &\Leftarrow G_2 \wedge R_{Y1} \\ Y_2' &\Leftarrow G_2 \wedge R_{Y2} \\ &\vdots \\ Y_n' &\Leftarrow G_2 \wedge R_{Yn} \end{aligned} \tag{16}$$

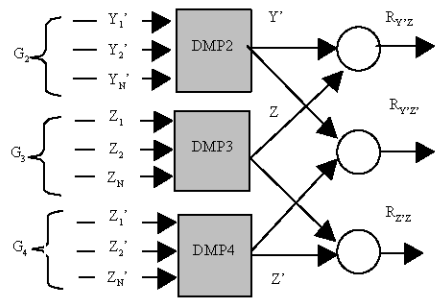

Fig. 4: Fault detection and isolation between alternative redundant groups of devices

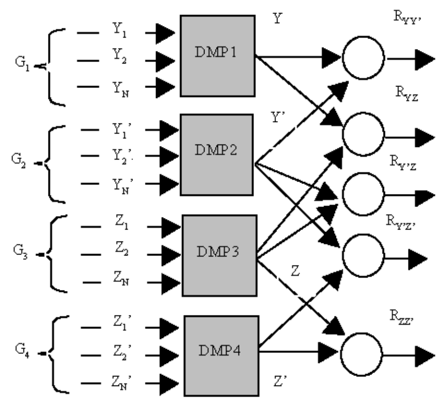

Fig. 5: Fault detection and isolation between both, main and its redundant groups

Where,

$$R_{Y1} = Y_1 - Y_1', R_{Y2} = Y_2 - Y_2', R_{Yn} = Y_n - Y_n'$$

$$\begin{aligned} Z_1 &\Leftarrow G_3 \wedge R_{Z1} \\ Z_2 &\Leftarrow G_3 \wedge R_{Z2} \\ &\vdots \\ Z_n &\Leftarrow G_3 \wedge R_{Zn} \\ Z_1' &\Leftarrow G_4 \wedge R_{Z1} \\ Z_2' &\Leftarrow G_4 \wedge R_{Z2} \\ &\vdots \\ Z_n' &\Leftarrow G_4 \wedge R_{Zn} \end{aligned} \tag{17}$$

Where,

$$R_{Z1} = Z_1 - Z_1', R_{Z2} = Z_2 - Z_2', R_{Zn} = Z_n - Z_n'$$

Decision-making and reconfiguration: The decision under a single faulty device with redundancy consists in enable the redundant stand-by device when a fault appear in a device of a main group of devices as soon as possible in order to avoid additional disturbances due to instrumentation faults, avoiding the imminent shut down of the plant. So that, in the same sample cycle where Eq. (16) or (17) detects a fault, reconfiguration must be carried out.

In this study, it has been shown that combining hardware redundancy with functional redundancy, ambiguity is avoided and the FD and FI problem is deterministically solved under some constraints such as:

- Residuals evaluation must be performed only under steady state dynamics.
- Determinism exists only under a unique fault and not more than one at a time under normal process operation.

IMPLEMENTATION PROCEDURE ON A HEAT EXCHANGER CONTROL SUPERVISION TASK

Given a pilot plant consisting in a heater exchanger process defined by means of functional approximation devices under NNBM1, NNBM2 and both redundant groups of devices under NNBM3 and NNBM4, as shown in Fig. 6, by applying Eq. (9) yields:

$$\begin{aligned} &\mathrm{DMP1}: Y = f(U, \Delta p, \Delta T) \\ &\mathrm{DMP2}: Y' = f(U', \Delta p', \Delta T') \\ &\mathrm{DMP3}: Z = f(qi, Ti, T) \\ &\mathrm{DMP4}: Z' = f(qi', Ti', T') \end{aligned} \tag{18}$$

Where,

U, Δp, ΔT	=	Inputs to DMP1.
U', Δp', ΔT'	=	Redundant inputs to DMP2.
q, Ti, T	=	Inputs to DMP3.
q', Ti',.T,'	=	Redundant inputs to DMP4.
Y, Y', Z and Z'	=	DMP1, DMP2, DMP3 and DMP4 outputs, respectively.

The aim of this study is to implement the proposed procedure applying detection, isolation, decision making and recovery tasks under described methodology on a severe heater exchanger control system. The heater is being controlled under feedback, feedforward and cascade modes used computed variable in cascade loop or feedback inner loop. For this reason, it is crucial to keep the data acquisition system running properly even under faults in data acquisition system. Fault tolerance requires the capacity to reconfigure the plant when a fault is detected and isolated, avoiding disturbances on the controlled variables. Recovery is a process carried out on the controlled plant after a decision making procedure to keep the plant running under required performance standards.

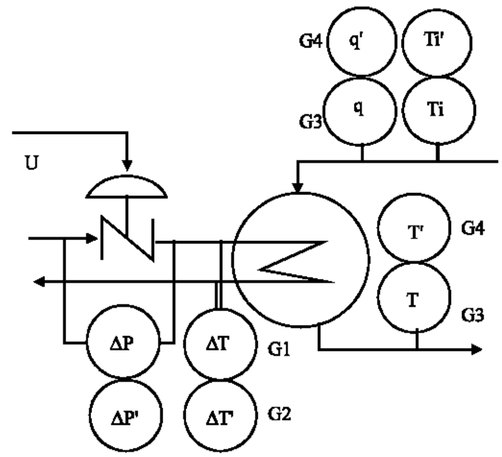

Fig. 6: Main and redundant devices allocated to the pilot plant

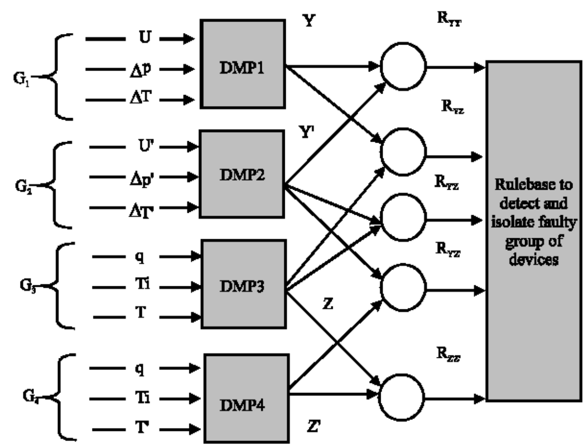

Fig. 7: Achieving residuals Fault detection and isolation between both, main and its redundant groups

Fault detection: Fault detection at groups level requires the application of Eq. (13), (14) and globally (15). Consequently, applying the described procedure follows that, using the simplified combination of both main and its redundant groups shown in Eq. (15), yields:

$$\begin{aligned} &G_1 \Leftarrow R_{YY'} \wedge R_{YZ} \wedge \bar{R}_{Y'Z} \\ &G_2 \Leftarrow R_{YY'} \wedge R_{Y'Z} \wedge \bar{R}_{YZ} \\ &G_3 \Leftarrow R_{YZ} \wedge R_{Y'Z} \wedge \bar{R}_{YY'} \\ &G_4 \Leftarrow R_{Y'Z'} \wedge R_{ZZ'} \wedge \bar{R}_{Y'Z} \end{aligned} \tag{19}$$

The meaning of such asseveration concluded by the Eq. (10) is depicted by Fig. 7.

Fault isolation at device level: Exploiting the concept of hardware redundancy, a faulty device is isolated by the rule-based inferential procedure expressed in (16) and (17):

$$\begin{aligned} U &\Leftarrow G_1 \wedge R_{Y1} \\ \Delta P &\Leftarrow G_1 \wedge R_{Y2} \\ \Delta T &\Leftarrow G_1 \wedge R_{Y3} \\ U' &\Leftarrow G_2 \wedge R_{Y1} \\ \Delta P' &\Leftarrow G_2 \wedge R_{Y2} \\ \Delta T' &\Leftarrow G_2 \wedge R_{Y3} \end{aligned} \tag{20}$$

Where,

$$R_{Y1} = U - U', R_{Y2} = \Delta P - \Delta P', R_{Y3} = \Delta T - \Delta T'$$

$$\begin{aligned} q &\Leftarrow G_3 \wedge R_{Z1} \\ Ti &\Leftarrow G_3 \wedge R_{Z2} \\ T &\Leftarrow G_3 \wedge R_{Z3} \\ q' &\Leftarrow G_4 \wedge R_{Z1} \\ Ti' &\Leftarrow G_4 \wedge R_{Z2} \\ T' &\Leftarrow G_4 \wedge R_{Z3} \end{aligned} \tag{21}$$

Where

$$R_{Z1} = q - q', R_{Z2} = Ti - Ti', R_{Z3} = T - T'$$

Recovery task: Once a main sensor fault is isolated, recovery is the unique active task. It consists in the addressing of redundant multiplexed device interchanging a redundant stand-by sensor by the actual faulty on-line sensor in the minimum required time.

The structure of controlled pilot plant is shown in Fig. 8, where feedback, feedforward and cascade measuring devices are installed with redundancy.

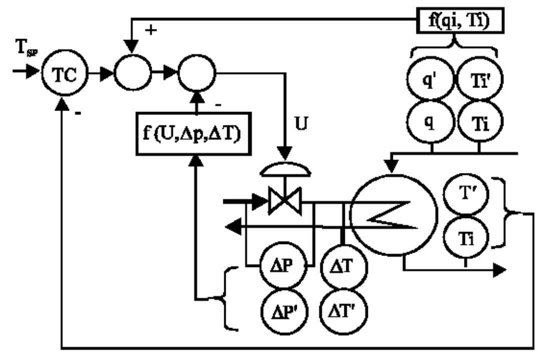

Fig. 8: Block diagram of controlled pilot plant

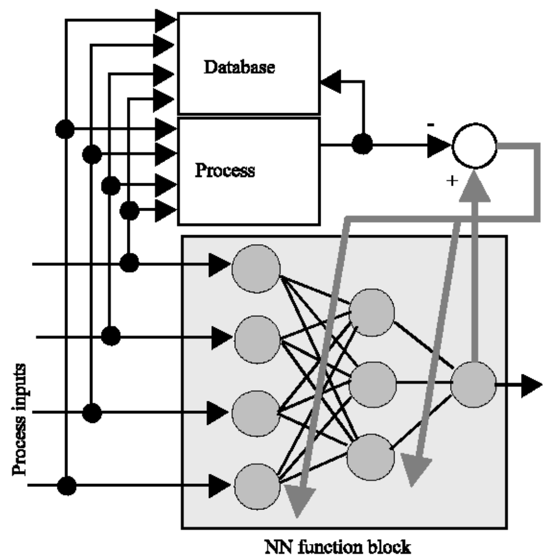

Fig. 9: Structure of a NN function block for on-line training

IMPLEMENTATION TOOLS

Deltav neural tool characteristics: DeltaV Neural provides easy-to-use tools for developing and training the NN model. This tool gives us a practical way to create virtual sensors for measurements previously available only through the use of lab analysis or online analysers. It is easy to understand and use, allowing process engineers to produce extremely accurate results even without prior deep knowledge of NN theory. In Fig. 9 it is shown the structure of a NN Function Block connected to operate in an on-line training phase.

The most relevant characteristics are summarised as:

- Easily creates virtual sensors using NN.
- NN executes right in the DeltaV controller as a function block.
- Automated pre-processing, design, training and verification.
- Expert mode allows interaction in the NN development.

Some other relevant characteristics are:

Variability reduction: Continuous virtual measuring of qualitative and analytical parameters allows for accepted tighter control of many process parameters. This provides automatic compensation for unmeasured disturbances and process changes. DeltaV Neural enables "IF-Then" analysis of a process change based on future prediction of critical parameters.

Process availability: Provides a backup and crosscheck on a measurement provided by a sampled or continuous analyser like mass spectrometers and stack analysers. Provides continuous measurement for all parameters measured by multi-streaming analysers.

Time saving with automatic network training: It supposes an advantage the use of easy-to-understand graphical tools for configuration and training of the network. Drag-and-drop configuration and automatic historical collection make the DeltaV Neural accessible to process engineers in need a real-time qualitative analysis.

Solid and fast execution: Running the NN right in the DeltaV controller means that under redundant controllers, we therefore have redundant NN at no incremental cost. It executes as fast as once every second. In addition to performance benefits, this methodology allows implementation without the requirement for costly host computers interfaced to the DeltaV system in a supervisory fashion.

Advantages in applying this tool with regard to conventional approaches: In practical terms offers an entirely new approach to the implementation of virtual sensors with NNs. Using the DeltaV Neural function block we can identify up to 20 individual process measurements to be correlated with lab entry or continuous analyser data. No step testing or manual disturbance of the process is necessary in order to implement the NN.

DeltaV Neural is implemented as a Function Block that executes in the DeltaV controller. This allows to use the standard tools of DeltaV Control Studio to define the necessary input variables along with manual lab entry data or data from a continuous analyser.

The DeltaV Continuous Historian automatically collects data on the inputs used by the Neural Net Function Block completely eliminating the need to configure a process historian. Alternatively we may import existing historical data into DeltaV Neural using commonly available tools such as Microsoft Excel for data preparation.

DeltaV Neural will automatically perform the training needed to build the network and stop when over training is detected. The historical data used to train the model can be easily viewed and any portions containing abnormal operating conditions may be excluded using easy graphical tools.

Upon completion of the automated network training, the sensitivities of each process input may be viewed graphically. Such tool is capable of eliminating any variables shown to have little or no effect on the output.

Additionally, experts have the option to specify such detailed parameters as outlier limits, max/min number of hidden neurons and maximum training epochs.

Verification of actual and predicted values vs. samples gives the user an easily understable picture of how the network behaves. Verification may be done against original data or any other user selectable timeframe. Table 1 shows the NN function block characteristics.

Table 1: Specifications of DeltaV neural tool

Outputs	1
Inputs	20
Controller loading for1-sec. execution rate	1.55% of fully populated NN FB

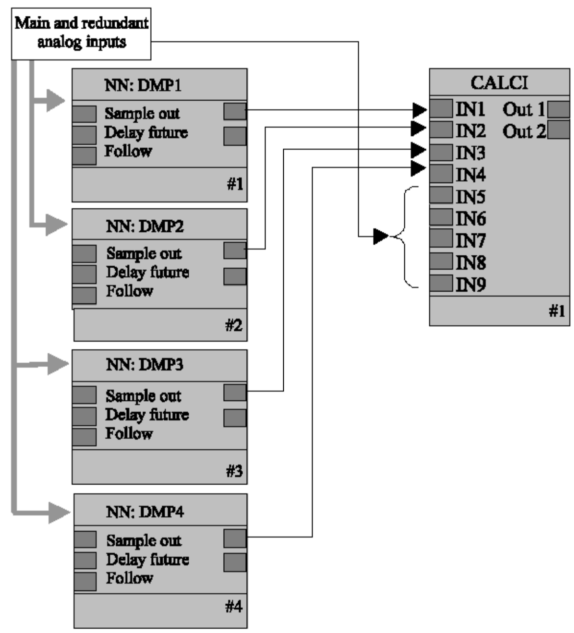

Fig. 10: Implementation layout of supervision task described by figure 8 with DeltaV Neural tool

Implementation: Implementation of proposed methodology is carried out with the facilities provided by a FOUNDATION™ Fieldbus compliant tool Delta (Anonymous, 2001).

As have been said, DeltaV Neural application has its roots in multi-layered feed forward neural network algorithm which is trained using backward propagation under a training conjugate gradient algorithm. Compared to traditional neural network products, such tool permits advanced features, such as automatic network update based on analyser or lab entry of new sample values and estimation of future value of the measurement based on current upstream conditions. The accuracy of the measurement estimate is substantially improved as a result of these enhancements. Inherently, this tool permits the selection of inputs identified as potentially influencing the process dynamics. Consequently, inputs that are most significant, are identified and used in training the neural networks. An advantage in using proposed tool is that understanding the details of the neural network algorithm is not necessary to successfully use the DeltaV Neural tool (2001).

The rule-based decision making procedure is carried out by means of a CALC1 function block as shown in Fig. 10, which permits the edition of rule bases by means of the structured text language of the IEC-1131-3 standard, in which decision-making tasks and devices scheduling are included.

DISCUSSION

The Fig. 11 shows the heat exchanger response and two measuring devices T and T' for the described application. In order to check the performance of supervision task, sensors were manipulated alternatively in sequential order to generate, detect, isolate faults and solve the problem of continuing supplying the proper measured data to the control algorithm. It is shown that after sensor T fails, the heater response deviates from its setpoint. This is due to an adjustable limit value of residual necessary to detect the fault. After fault is detected then, the proper redundant sensor is switched on line and heater exchanger response start a stabilization phase. It is shown that if sensor fault is abrupt (case of power shut down) then residuals are evaluated in the same sample cycle and the change of sensor (system recovery) is also instantaneous.

Besides the solved supervision task carried out on instrument performance status by diagnosing faulty sensors under any fault in a single device under steady state conditions, additional relevant information can be retrieved from closed loop controlled systems, that helps very much in decision making procedure. It has been shown that knowledge about faulty sensors is crucial

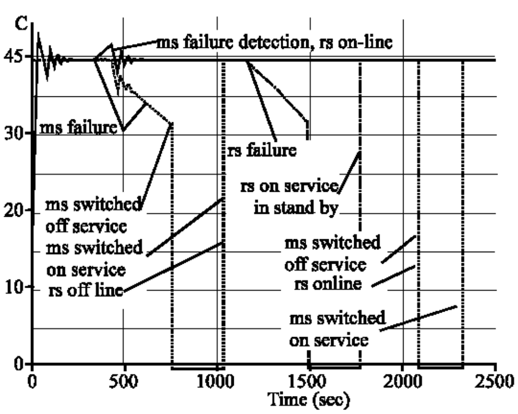

Fig. 11: Layout of a fragment of SCADA for instrumentation supervision of exchanger pilot plant

when such sensors are responsible for influencing control loops as in present case. For this reason, decision-making strategy on recovery task requires strongly the implementation of additional knowledge. Such task is successfully carried out by adding some specific knowledge to the actual rulebase with the addition of only but not limited to an If Then Rule under the achieved status condition. So that, decision making on system recovery task is carried out by means of the following rule under the assumption of correct control algorithm:

IF no sensor fault detected AND steady state error > error_limit, THEN a subtask to analyse the actuator and process performance is to be carried out.

If process has changed then modelling error exist and NNBMs must be updated accordingly. Once updated, the actuator will be the unique device susceptible of fault where recovery is only possible if the plant is equipped with a redundant actuator.

CONCLUSION

A systematic methodology to implement the supervision task of process instrumentation applied on industrial processes has been developed and presented. The approach combines functional approximation implemented on the basis of massive back-propagation NN, with rule based strategies, both implemented with the facilities of an object oriented programming tool: the DeltaV Neural. System recovery by means of a failure analysis strategy has been carried out to make a critic process control safely under measuring instrumentation faults. The application of proposed strategy requires to satisfy the following constraints:

- Any fault belongs to a single device for every sample time.
- Process operation is correct.
- System operates closely to steady state.

The availability of used advanced FOUNDATION™ Fieldbus based tools brings the gap between the proposed methodology and its implementation procedure.

In order to extensively validate such methodology, besides the results presented in this work, some tests were carried out on sub-modules of pilot plant under different conditions and faults, where results are acceptable.

REFERENCES

Anonymous, 2001. DeltaVTM V.5.3. Books on line. Copyright © 1994-2001, Fisher-Rosemount Systems, Inc., Emerson Process Management. USA.

Bakiotis, C. *et al*., 1979. Parameter and Discriminant Analysis for jet engine mechanical state diagnosis. In: Proc. of the IEEE Conf. On Decision and Control, Piscataway, New jersey. IEEE Press, pp: 11-1.

Bakshi, B.R. and G. Stephanopoulos, 1994. Representation of process trends, IV Induction of real time patterns from operating data for diagnosis and supervisory control. Comp. Chem. Eng., 18: 303-332.

Bawazeer, K.H. and A. Zilouchian, 1997. Prediction of Crude Oil Production Quality Parameters Using Neural Networks In: Proceeding IEEE International Conference on Neural Networks, New Orleans.

Bawazeer, K.H., 1996. Prediction of Crude Oil Product Quality Parameters Using Neural Networks, MS Thesis, Florida Atlantic University, Boca Raton, FL, August.

Clark, R.N. *et al*., 1975. Detecting instrument malfunctions in control systems. IEEE Trans. Aerospace and Electrooninc Syst., 11: 465-473.

David, J.G., T. Degroff and J.S. Gero, 2003. Improving neural network models of physical systems through dimensional analysis. Applied Soft Computing. Elsevier, 2: 283-296.

Deckert, J.C. *et al*., 1977. F-8 DFBW sensor failure identification using analytic redundancy. IEEE Trans. on Automatic Control, 22: 795-803.

Demuth, H. And M. Beale, 1998. Neural Network Toolbox for Use with MATLAB, the Math Works Inc., Natick, MA, USA.

Ding, X. and L. Guo, 1996. Observer based fault detection optimised in the frequency domain. In: Proc. 13th IFAC World Congress, Piscataway, New Jersey. IEEE Press, pp: 157-162.

Doyle, R.J. *et al*., 1993. Causal modelling and data-driven simulation for monitoring of continuous systems. Computers in Aerospace, 9: 395-405.

Duda, R.O. and P.E. Hart, 1973. Pattern Classification and Scene Analysis. John Wiley and Sons, New York.

Frank, P. M., 1993. Robust model based fault detection in dynamic systems. On line fault detection and supervision in the chemical process industries. Pergamon Press. Oxford. IFAC Symposium series Number 1, pp: 1-13.

Frank, P.M., 1990. Fault diagnosis in dynamic systems using analytical and knowledge based redundancy: A survey and some new results. Automatica, 26: 459-474.

Frosini, L. and G. Petrecca., 2001Neural networks for load torque monitoring of an induction motor. Applied Soft Computing. Elsevier, 1: 215-223.

Ganesamoorthi, S., J. Colclazier and V. Tzovla, 2000. Automatic Creation of Neural Nets for Use in Process Control Applications. In: Proc ISA. Expo/2000, August 2000, New Orleans LA., USA

Gao, X.Z. and S.J. Ovask, 2001. Soft computing methods in motor fault diagnosis. Applied Soft Computing Elsevier, 1: 73-81.

Gertler, J.J., 1998. Fault detection and diagnosis in engineering systems. Marcel Decker, Inc. New York.

Hodouin, D. and S. Makni, 1996. Real time reconciliation of mineral processing plant data using bilinear material balance equations coupled to empirical dynamic models. Int. J. Miner. Proc., 48: 245-264.

Isserman, R. 1993. Fault diagnosis of machines via parameter estimation and knowledge processing: Tutorial paper. Automatica, 29: 815-835.

Isserman, R. 1998. Process Fault Detection based on modelling and estimation methods: A survey. Automatica, 20: 387-404.

Jackson, J.E., 1956. Quality control methods for 2 related variables. Industrial Quality Control., 7: 2-6.

Jackson, J.E., 1959. Quality control methods for several related variables. Technometrics, 1: 359-377.

Kramer, M.A. and F.E. Finch, 1988. Development and classification of expert systems for chemical process fault diagnosis. Robotics and Computer Integrated Manufacturing, 4: 437-446.

Kramer, M.A. and J.B.L. Palowitch, 1987. A rule-based approach to fault diagnosis using the signed direct graph. AIChE. J., 37: 1067-1078.

Lee, G. *et al*., 1999. Multiple-fault diagnosis under uncertain conditions by the quantification of qualitative relations. Ind. Eng. Che. Res., 38: 988-998.

Li, T., 1989. Expert Systems for Engineering Diagnosis: Styles, requirements for tools and adaptability. Ed. Tzafestas, S.G. Knowledge-based Systems Diagnosis, Supervision and Control, Plenum Press, New York, pp: 27-37.

MacGregor, J.F., 1994. Statistical Process Control of multivariate processes. In: Proc. of the IFAC Int. Symp. On Advanced Control of Chemical Processes, pp: 427-435, New York. Pergamon Press.

Mehra, R.K. and J. Peschon, 1971. An Innovation approach to fault detection and diagnosis in dynamic systems. Automatica, 7: 637-640.

Mehta, A., S. Ganesamoorthi and W. Wojsznis, 2001. Feedforward Neural Networks for Process Identification and Prediction. Proc of ISA. Expo/2001, Sept., 2001 Houston. USA.

Mironovski, L.A., 1979. Functional diagnosis of linear dynamic systems. Automation and Remote Control, 40: 1198-1205.

Mironovski, L.A., 1980. Functional diagnosis of linear dynamic systems-A survey. Automation and Remote Control, 41: 1122-1142.

Mo, K.J. *et al*., 1997. Development of operation-aided system for chemical processes. Expert Syst. with Appl., 12: 455-464.

Mo, K.J. *et al*., 1998. Robust fault diagnosis based on clustered symptom trees. Control Eng. Practice, 5: 199-208.

Nekovie, R. and Y. Sun, 1995. Back propagation Network and its Configuration for Blood Vessel Detection in Angiograms. IEEE Trans. Neural Networks, 6: 1.

Piovoso, M.J. and K.A. Kosanovich 1992. Monitoring process performance in real time. In: Proc. of the American Control Conference, Piscataway, New Jersey. IEEE Press, pp: 2359-2363.

Piovoso, M.J. and K.A. Kosanovich, 1994. Applications of multivariate statistical methods to process monitoring and controller design. Int. J. Cont., 59: 743-765.

Roelof, K.B. and W. Pedrycz, 2003. Training a feed-forward network with incomplete data due to missing input variables. Applied Soft Computing Elsevier, 3: 23-36.

Stephen, O.T.O. and R. Singh, 2003. Advanced engine diagnostics using artificial neural networks. Applied Soft Computing Elsevier, 3: 259-271.

Terrence, L.B., G.K. McMillan, W.K. Wojsznis and M.W. Brown, 2003. Advanced Control Unleashed-Plant Performance Management for Optimum Benefit. ISA. The Instrumentation, Systems and Automation Society. USA.

Tzovla, V. and A. Mehta, 2001. Automated Approach to Development and Online Operation of Intelligent Sensors. Proc of ISA. Expo/2001, Houston. USA.

Wise, B.M. and N.B. Gallagher, 1996. The process chemometrics approache to process monitoring and fault detection. J. Proc. Cont., 6: 329-348.

Xu, K., M. Xie, L.C. Tang, S.L. Ho, 2003. Application of neural networks in forecasting engine systems reliability. Applied Soft Comput. Elsevier, 2: 255-268.

Zilouchian, A. and K. Bawazeer, 2001. Application of neural networks in oil refineries. Intelligent Control Systems Using Soft Computing Methodologies, In: Zilouchian, A. and M. Jamshidi (Eds.). CRC Press, USA, pp: 139-158.