Research Journal of Applied Sciences 5 (4): 274-282, 2010
ISSN: 1815-932X
© Medwell Journals, 2010

Computed Multi-Variable Control Applied on Viscosity Control by Means of In-line Mixing Control Using Neural Networks

Ramon Ferreiro Garcia, Alberto DeMiguel Catoira and Jose Luis Calvo Rolle
Department of Industrial Engineering, University of a Coruna, La Coruva, Spain

Abstract: Model-based functional approximation techniques are being used on in-line mixing processes affected by controlled variables which depend strongly on complementary physical variables. The aim for such type of nonlinear control problems is to compute the proportions of input product flow rates yielding a final product thus satisfying as much physical properties as manipulated input flow rates exists. The core of the contribution is based on functional approximation implemented by means of backpropagation neural networks associated to the computed multivariable control strategy on a viscosity control problem.

Key words: In-line mixing, computed variable control, functional approximation, feedforward neural networks, conjugate gradient algorithm, multivariable control

INTRODUCTION

An important particularly aspect of conventional control was the idea of controlling the variables that are of real interest by computing its values from other measurements (Luyben, 1990). Conventionally, some mixing problems were solved successfully by means of conventional computational methods and means. With the help of modern computational methods, much more complex types of computed variables can now be calculated. Several variables of a process can be measured and all the other variables can be calculated from a rigorous model of the process or from virtual sensors based on soft-computing techniques. For instance, the nearness of flooding in distillation columns can be calculated from heat input, feed flow rate and temperature and pressure data. Another application is the calculation of product purities in a distillation column from measurements of several tray temperatures and flow rates by the use of mass and energy balances, physical property data and vapour-liquid equilibrium information.

The use of available and sophisticated computational methods made these rigorous estimators feasible. It opens up a number of interesting possibilities in the control field without limitations in applying such powerful methods even with the scarcity of engineers who understand both control and chemical engineering processes well enough to apply them effectively.

A typical class of mixing problems involve linear models such the problem of thermal mixing where the problem is to control the temperature of an output flow from a tank by proportioning the input flow into the tank or the problem of concentration mixing where two fluids of different concentrations are mixed to produce a desired concentration by varying the input flow rate. In both cases, the material-balance and energy-balance equations are the basis of process modelling.

Some other mixing problems are not linear such those problems involving temperature, pressure, viscosity, conductivity, pH (Mahuli *et al.*, 1992) composition, etc. For instance, the liquids used in hydraulic systems generally exhibit large changes in viscosity with relativity small changes in temperature. The relative changes in viscosity per degree is called the temperature coefficient of viscosity C_T and is expressed as:

$$C_T = \frac{d\mu}{\mu dT} \tag{1}$$

Where:
T = The actual temperature
μ = The absolute viscosity

Pressurized liquids tend to increase viscosity where this phenomenon is particular evident in oils. At low or moderate pressure this increase is relatively small but at high pressures, the viscosity increases quite rapidly (Stern *et al.*, 1958). The exponential relationship between viscosity and pressure is given by the expression:

$$\mu = \mu_0 e^{\alpha P} \tag{2}$$

Where:
α = The pressure coefficient of viscosity
μ_0 = The viscosity at atmospheric pressure
μ = The viscosity at pressure P

Corresponding Author: Ramon Ferreiro Garcia, Department of Industrial Engineering, University of a Coruna, La Coruva, Spain

Equation 2 is actually only a rough approximation, since the pressure coefficient α is not a constant but is dependent on pressure, temperature and type of liquid (liquid components). Because of this rough approximation, precision control problems require another alternative modelling method.

The continuous mixing control problem: In the implementation of composition control systems in-line mixers are often considered. Properly applied, these devices are effective but careful attention to the following design criteria is required: reagent delivery hysteresis, loop gain and neutralization stage interaction (Hoyle and McMillan, 1995). An in-line mixer can be a dynamic mixer such as a centrifugal pump or a baffled section of pipe called a static mixer as shown in Fig. 1. The static mixer provides radial mixing but little backmixing. It can be considered to be a plug flow process dominated by dead time. Disturbances and noise pass through the mixer unattenuated. With such a mixer, the best controller response to fast disturbance and noise is no response at all because any corrective action will arrive too late and will create yet another disturbance. The advantages of in-line mixers are its small dead time, loop period and recovery time. Conventionally, control structures based in the combination of feedback, feedforwared, cascade and ratio control are used.

The mixer exhibits certain pure time delay due to the inherent transport lag. This time lag D is described as the linear function of fluid flow rate q, the net cross section of the pipe A and the length of pipe L from the control valves to the end of mixer which in time domain yields:

$$D = \frac{L \cdot A}{q} \tag{3}$$

So that the inherent time lag is e^{-DS}

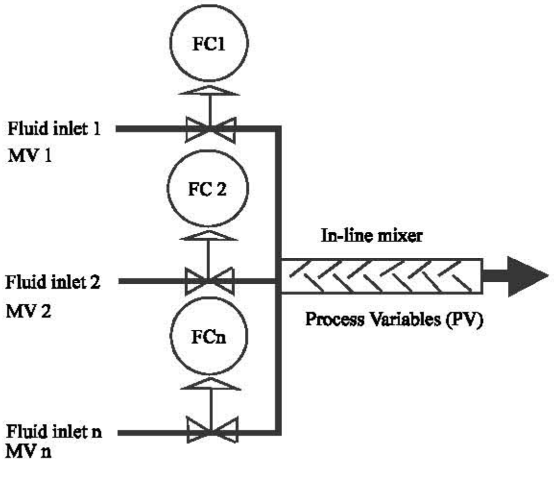

Fig. 1: Continuous mixer structure

The aim of this analysis obeys to two main reasons:

- Avoid the effect of time lags on feedback control which suppose an important disturbance on mixing control loop
- Simplify mentioned conventional control structures. In order to achieve such topics, model based computing variable control methods are proposed

The proposed multivariable control strategies consists of two open loop control schemes:

Computed variable Feedforward-Cascade Contro (FCC) l: Consists in compute the manipulated variables as function of the desired controlled variables and input process variables.

Computed variable Estimated Feedback Contro (EFC) l: Consists in compute the process variables as function of manipulated variables and process characteristics.

MODELLING THE PROPOSED IN-LINE MIXER PROCESSES

When analytical models based in physical laws (first principles) such as energy or material balances does not provide an acceptable or satisfactory solution, model based functional approximation is proposed. In-line mixing processes affected by controlled variables such as conductivity, pH, viscosity which depends strongly on complementary physical variables such as temperature and pressure variations are appropriate candidates to be solved using functional approximation.

For instance, the fact of mixing two products of known characteristics such as flow rate, pressure, temperature and viscosity denoted as ($q_{1,}$ p_1, T_1, μ_1) and ($q_{2,}$ p_2, T_2, μ_2), yields a final product ($q_,$ p, T, μ). By the fact of acting on the input flow rates (two variables) it is not possible to control the output product whose composition depends on four variables. For this type of nonlinear control problem, the aim is to compute the proportions of input product flow rates to achieve a final product satisfying as much physical properties as manipulated input flow rates exists using some innovative methodologies.

Functional approximation implemented on the basis of backpropagation neural networks associated to the described control strategies (FCC and EFC), conforms the core of the contribution.

Using conductivities on concentration mixing problem: The conductivity of a mixed fluid can be a measure of the concentration. In fact, the mixture of two fluids of different conductivities where the individual conductivities are function of its respective temperatures can be expressed as:

$$(q,C,T) = f(q_1,q_2,q_3, T_1, T_2,T_3,C_1,C_2,C_3) \qquad (4)$$

The two modes of computed variable control are:

$$\begin{aligned}&(q_D,C_D,T_D) = f(q_1,q_2,q_3,T_1,T_2,T_3,C_1,C_2,C_3)\\&(q_{1(SP)},q_{2(SP)},q_{3(SP)}) = (q_D,C_D,T,T_1,T_2,T_3,C_1,C_2,C_3)\end{aligned} \qquad (5)$$

For the nonlinear mixing process described by Eq. 4 some functional dependences from physical properties are estimated as:

$$\begin{aligned}&q = f(q_1,q_2)\\&T = f(q_1,q_2,T_1,T_2)\\&C = f(T,C_1,C_2) = f(q_1,q_2,T_1,T_2,C_1,C_2)\end{aligned} \qquad (6)$$

The open loop control problem by computed variable control strategies consists in computing the setpoints of a number of manipulated variables (flow rates) to satisfy the required values of a number less or equal of controlled variables.

It means to compute $q_{1(SP)}$, $q_{2(SP)}$ such that by means of an in-line mixing process, it will be possible to achieve a final product thus satisfying desired characteristics such as flow rate q_D and conductivity C_D or temperature and conductivity C_D. According this definition, Eq. 18 for the case of two degrees of freedom, yields:

$$\begin{aligned}&(q_D,C_D) = f(q_1,q_2,T_1,T_2,C_1,C_2)\\&(T_D,C_D) = f(q_1,q_2,T_1,T_2,C_1,C_2)\end{aligned} \qquad (7)$$

The definition and application of the appropriate inverse model when applicable because direct models are also useful provides a (Wachira *et al*., 2005) function suitable to specify the desired final product (q_D, C_D) as:

$$(q_{1(SP)},q_{2(SP)}) = f(q_D,C_D,T_1,T_2,C_1,C_2) \qquad (8)$$

So that the functional approximators given by Eq. 6 and 7 can be implemented on the basis of backpropagation neural networks as shown in Fig. 2. Figure 3 shows the computed variable control strategy for both computed variable by FCC and computed variable by EFC.

Viscosity mixing problem under variable temperatures: The problem of mixing two fluids of different viscosities and temperatures is described by direct model:

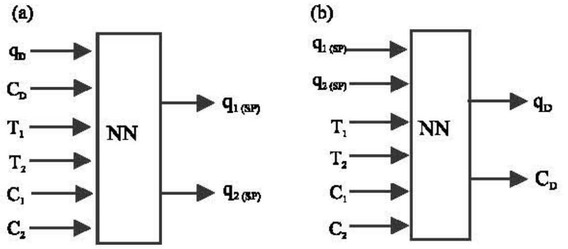

Fig. 2: Computed variable control algorithm based on functional approximation applied to an In-line mixer. (a) Algorithm for FCC (b) Algorithm for EFC

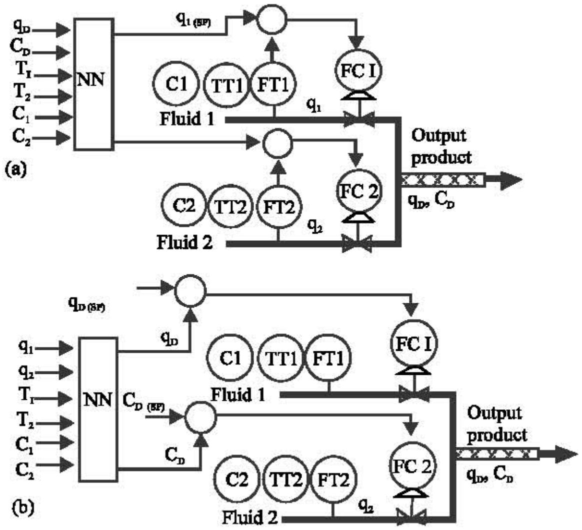

Fig. 3: Computed variable control strategy based on functional approximation applied to an In-line mixer. (a) Computed variable by FCC (b) Computed variable by EFC

$$q,\mu,T = f(q_1,q_2,T_1,T_2,\mu_1,\mu_2) \qquad (9)$$

Proceeding according last mixing problem they are achieved both control algorithms by means of an inverse model: computed variable by FCC and computed variable by EFC as:

$$(q_{1(SP)},q_{2(SP)}) = f(q_D,\mu_D,\mu_1,\mu_2) \qquad (10)$$

$$q_D,\mu_D = f(q_{1(SP)},q_{2(SP)},\mu_1,\mu_2) \qquad (11)$$

Viscosity mixing control problem under variable temperature and pressure: The problem of mixing two fluids of different viscosities, different temperatures and pressure variations is described by a direct model as:

$$q,\mu,T,P = f(q_1,q_2,T_1,T_2,P_1,P_2,\mu_1,\mu_2) \qquad (12)$$

This mixing control problem is solved as in the last case through an inverse model but the function must include the variable pressure yielding:

$$(q_D, \mu_D) = f(q_1, q_2, \mu_1, \mu_2) \tag{13}$$

$$(q_{1(SP)}, q_{2(SP)}) = f(q_D, \mu_D, \mu_1, \mu_2) \tag{14}$$

APPLICATION ON DIESEL ENGINE SPEED CONTROL

This application deals with the task of supplying the necessary Fuel Oil (F.O) to a Diesel engine to get the required power in terms of engine speed (rps). In order to supply a demanded fuel flow rate at a required viscosity without heating the fuel (that is at ambient temperature), two fuel lines at different viscosities (heavy F.O and light F.O) provides the required amount of fuel at the proper viscosity by means of an in-line mixer controlled on the basis of proposed control strategies.

So that Eq. 10 and 11 solves the problem of mixing control which is shown in Fig. 4 and 5, respectively. Figure 4 shows the computed variable control by feedforward-cascade strategy while in Fig. 5 it is shown the computed variable control by estimated feedback strategy.

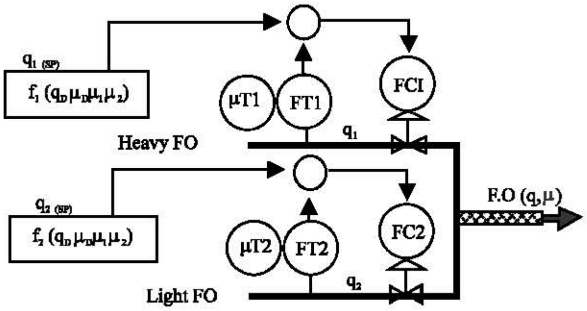

Fig. 4: Computed variable control by feedforward-cascade strategy on an in-line mixer to achieve a required viscosity and flow rate

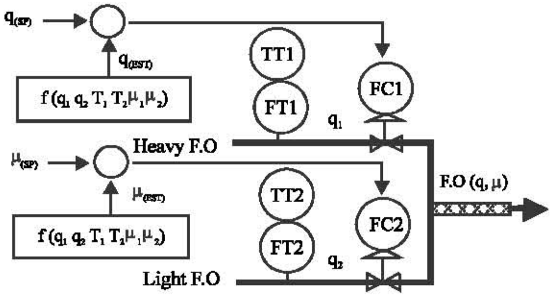

Fig. 5: Computed variable control by estimated feedback strategy on an in-line mixer to achieve a required viscosity and flow rate

Next step of design procedure deals with controller design, validation and implementation. As stated before, controller is implemented on the basis of functional approximation. The function approximators will be implemented by means of feedforward neural networks (Demuth and Beale, 2000; Hagan *et al*., 1996).

Neural network controller implementation: Feedforward networks are the most suitable architectures to be used as general functional approximators.

The basic Backpropagation training algorithm in which the weights are moved in the direction of the negative gradient is used as training algorithm for feedforward networks. The term Backpropagation refers to the manner in which the gradient is computed for nonlinear multilayer networks.

There are a number of variations on the basic algorithm that are based on other standard optimization techniques such as conjugate gradient and Newton methods.

Backpropagation was created by generalizing the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions (Widrow and Hoff, 1960).

Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors or classify input vectors in an appropriate way as defined by you. Properly trained Backpropagation networks tend to give reasonable answers when presented with inputs that they have never seen. Typically, a new input leads to an output similar to the correct output for input vectors used in training that are similar to the new input being presented.

This generalization property makes it possible to train a network on a representative set of input/target pairs and get good results without training the network on all possible input/output pairs. Generally, the four steps in the training process are:

- Assemble the training data
- Create the network object
- Train the network
- Simulate the network response to new inputs

Conjugate gradient algorithms: The basic backpropagation algorithm adjusts the weights in the steepest descent direction (negative of the gradient). In this direction the performance function decreases most rapidly. It must be noted that, although the function decreases most rapidly along the negative of the gradient, this does ensure the fastest convergence. In the conjugate gradient algorithms (Hagan and Menhaj, 1994), (Powell, 1977) a search is performed along conjugate directions which produces generally faster convergence than steepest descent directions. (Beale, 1972), (Charalambous, 1992).

In most of the training algorithms, a learning rate is used to determine the length of the weight update (step size). In most of the conjugate gradient algorithms, the step size is adjusted at each iteration (Moller, 1993). A search is made along the conjugate gradient direction to determine the step size which minimizes the performance function along that line (Andrasik *et al*., 2004).

Fletcher-reeves update (traincgf): All of the conjugate gradient algorithms start out by searching in the steepest descent direction (negative of the gradient) on the first iteration (Fletcher and Reeves, 1964).

$$p_0 = -g_0 \tag{15}$$

A line search is then performed to determine the optimal distance to move along the current search direction:

$$x_{k+1} = x_k + \alpha_k p_k \tag{16}$$

Then the next search direction is determined so that it is conjugate to previous search directions. The general procedure for determining the new search direction is to combine the new steepest descent direction with the previous search direction:

$$p_k = -g_k + \beta_k p_{ki-1} \tag{17}$$

The various versions of conjugate gradient are distinguished by the manner in which the constant is computed. For the Fletcher-Reeves update the procedure is:

$$\beta_k = \frac{g_k{}^T g_k}{g_{k-1}^T g_{k-1}} \tag{18}$$

This is the ratio of the norm squared of the current gradient to the norm squared of the previous gradient (Fletcher and Reeves, 1964; Hagan *et al*., 1996). The conjugate gradient algorithms are usually much faster than variable learning rate Backpropagation, although the results will vary from one problem to another. The conjugate gradient algorithms require only a little more storage than the simpler algorithms, so they are a good choice for networks with a large number of weights.

Achieving a representative database: The Number of Data Sets (NDS) has its origin in the general expression $\prod_{i=1}^{i=n} RD_i$ where RD_i is the range division of each variable.

So that database focused on NN training is achieved using 256 datasets covering the ranges shown in Table 1 where NDS = $\prod_{i=1}^{i=n} RD_i$ = 4×4×4×4 = 256 datasets.

These datasets were achieved by measuring the viscosities of both input fluids (light Diesel oil and heavy Diesel oil) at different temperatures and viscosities, yielding the necessary flows (q_1 and q_2) to achieve the desired viscosity μ_D and flow rate q_D under the structure shown in Table 1. The ranges of mixing process input variables are shown in Table 2.

Training with matlab neural toolbox: Several feedforward NN structures has been tested against performance to solve the proposed problem.

Acceptable results were achieved using the neural networks structure shown in Table 3 where neural network toolbox of Matlab has been used. Table 3 shows also the specified training characteristics. Training results are shown in Fig. 6 and 7. Performances of 0.0015 and 0.007 are achieved with 300 epochs. Training characteristics were varied to test different NN architectures such as [8, 1], [10, 1], [6, 6, 1], [8, 8, 1] and finally [10, 10, 1] which is the one used in this research work.

Structure of the engine speed controller: The operating curves for an engine indicate that the output shaft speed ω is a nonlinear function of both the fuel flow rate Q and the load torque T_{orq}.

$$\omega = f(Q, T_{orq})$$

Nevertheless, the fuel flow rate demand Q_D is determined by the required engine speed using a PID control loop where the in-line mixer control module is embedded in series with the PID feedback controller. Thus, the block diagram of the engine speed control loop is shown in Fig. 8. The forward path contains the in-line mixer FCC or EFC controller.

Validation by means of the in-line mixer simulation: In order to show the performance of the proposed feedforward-cascade strategy on the In-line mixer that means given the known variable values Q_D, μ_D, μ_1, μ_1,

Table 1: The partial 64 datasets of the database corresponding to $\mu_D = 4.5$

μ_D	μ_1	μ_2	q_D	q_1	q_2	μ_D	μ_1	μ_2	q_D	q_1	q_2
4.5	4.0	15	1	0.9545	0.045	4.5	4.0	7	1	0.833	0.166
4.5	4.0	15	2	1.9100	0.010	4.5	4.0	7	2	1.660	0.340
4.5	4.0	15	3	2.8640	0.136	4.5	4.0	7	3	2.500	0.500
4.5	4.0	15	4	3.8180	0.182	4.5	4.0	7	4	3.334	0.666
4.5	3.1	15	1	0.8820	0.117	4.5	3.1	7	1	0.640	0.360
4.5	3.1	15	2	1.7650	0.235	4.5	3.1	7	2	1.280	0.720
4.5	3.1	15	3	2.6470	0.353	4.5	3.1	7	3	1.920	1.070
4.5	3.1	15	4	3.5300	0.470	4.5	3.1	7	4	2.560	1.440
4.5	2.5	15	1	0.8400	0.160	4.5	2.5	7	1	0.550	0.450
4.5	2.5	15	2	1.6800	0.320	4.5	2.5	7	2	1.110	0.890
4.5	2.5	15	3	2.5200	0.480	4.5	2.5	7	3	1.667	1.333
4.5	2.5	15	4	3.3600	0.640	4.5	2.5	7	4	2.224	1.776
4.5	1.9	15	1	0.8000	0.200	4.5	1.9	7	1	0.490	0.510
4.5	1.9	15	2	1.6000	0.400	4.5	1.9	7	2	0.980	1.020
4.5	1.9	15	3	2.4000	0.600	4.5	1.9	7	3	1.470	1.530
4.5	1.9	15	4	3.2000	0.800	4.5	1.9	7	4	1.960	2.040
4.5	4.0	10	1	0.9160	0.083	4.5	4.0	5	1	0.500	0.500
4.5	4.0	10	2	1.8330	0.166	4.5	4.0	5	2	1.000	1.000
4.5	4.0	10	3	2.7500	0.250	4.5	4.0	5	3	1.500	1.500
4.5	4.0	10	4	3.6670	0.333	4.5	4.0	5	4	2.000	2.000
4.5	3.1	10	1	0.7970	0.203	4.5	3.1	5	1	0.263	0.736
4.5	3.1	10	2	1.5900	0.410	4.5	3.1	5	2	0.530	1.470
4.5	3.1	10	3	2.3900	0.610	4.5	3.1	5	3	0.790	2.210
4.5	3.1	10	4	3.1900	0.810	4.5	3.1	5	4	1.060	2.940
4.5	2.5	10	1	0.7330	0.266	4.5	2.5	5	1	0.220	0.780
4.5	2.5	10	2	1.4670	0.533	4.5	2.5	5	2	0.400	1.600
4.5	2.5	10	3	2.2000	0.780	4.5	2.5	5	3	0.600	2.400
4.5	2.5	10	4	2.9300	1.060	4.5	2.5	5	4	0.800	3.200
4.5	1.9	10	1	0.6800	0.320	4.5	1.9	5	1	0.160	0.840
4.5	1.9	10	2	1.3600	0.640	4.5	1.9	5	2	0.320	1.680
4.5	1.9	10	3	2.0400	0.960	4.5	1.9	5	3	0.480	2.520
4.5	1.9	10	4	2.7200	1.280	4.5	1.9	5	4	0.650	3.350

The ranges of mixing process input variables are shown in Table 2. $q_1 = f_1$ (μ_D, μ_1, μ_2, q_D); $q_2 = f_2$ (μ_D, μ_1, μ_2, q_D)

Table 2: Ranges of variables for construct the training database

μ_1	μ_2	μ_D	q_D
4	15	5.0	1
1	5	3.5	4

Table 3: Training characteristics

NN based function approximator 1	NN based function approximator 2
p = Four column vector [p1, p2, p3, p4]	p = Four column vector [p1, p2, p3, p4];
T1 = A column vector [t1]	T2 = A column vector[t2]
Net = Newff(minmax (p), [10,10,1] {tansig, tansig, purelin}, traincgf)	net = Newff(minmax(p),[10,10,1] {tansig, tansig, purelin},traincgf)
Net.trainParam.show = 50	net.trainParam.show = 50
Net.trainParam.epochs = 200	net.trainParam.epochs = 200
Net.trainParam.goal = 1e-5	net.trainParam.goal = 1e-5
[net,tr] = train(net,p,T1)	[net,tr] = train(net, p, T2)
Gensim (net, -1)	gensim(net,-1)

achieve q_1 and q_2 by applying functional approximation, a simulation block diagram has been constructed with the neural networks achieved by means of training algorithm (traincgf or the conjugate gradient Fletcher-Reeves) of Matlab.

The Block diagram of simulation task is shown in Fig. 8. Signals Q_D, μ_D, μ_1, μ_1 enter both neural networks providing the fuel flow rate setpoints $q_{1(SP)}$ $q_{1(SP)}$ under the structure of Fig. 9. The inputs to the simulation task are variable to investigate the effect of disturbances due to the inputs. The viscosity setpoint remains in the nominal value which is 4.5 cts.

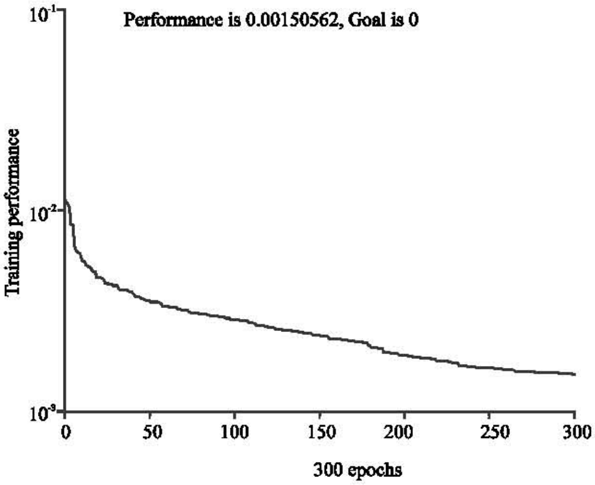

Fig. 6: NN1 Training evolution for function 1

The blocks simulating the neural networks NN1_10-10-1 = f_1 (q_D μ_D μ_1 μ_2) and NN2_10-10-1= f_2(q_D μ_D μ_1 μ_2)

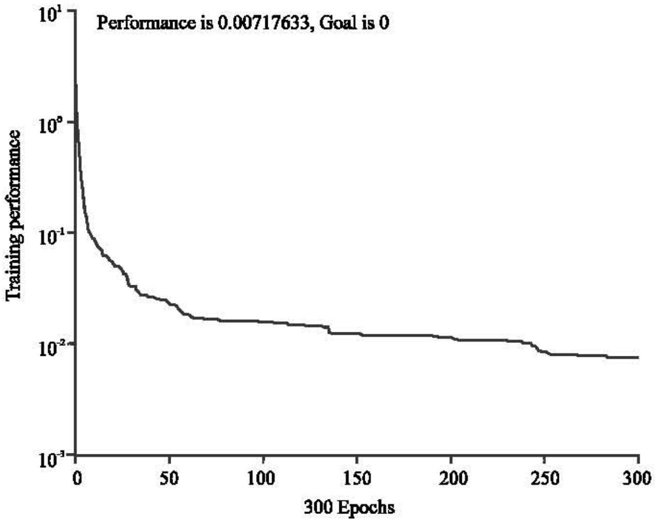

Fig. 7: NN2 training evolution for function 2

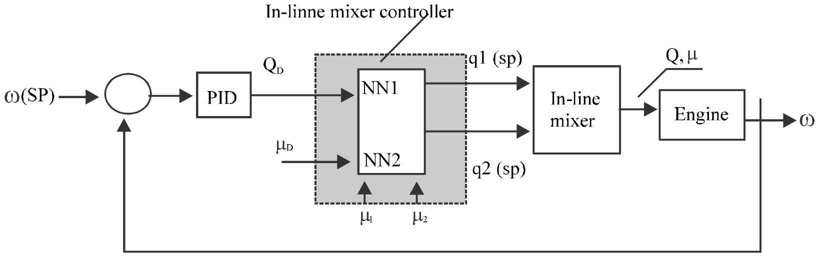

Fig. 8: Engine speed feedback control associated to a nonlinear multivariable feedforward-cascade controller: the in-line mixer controller

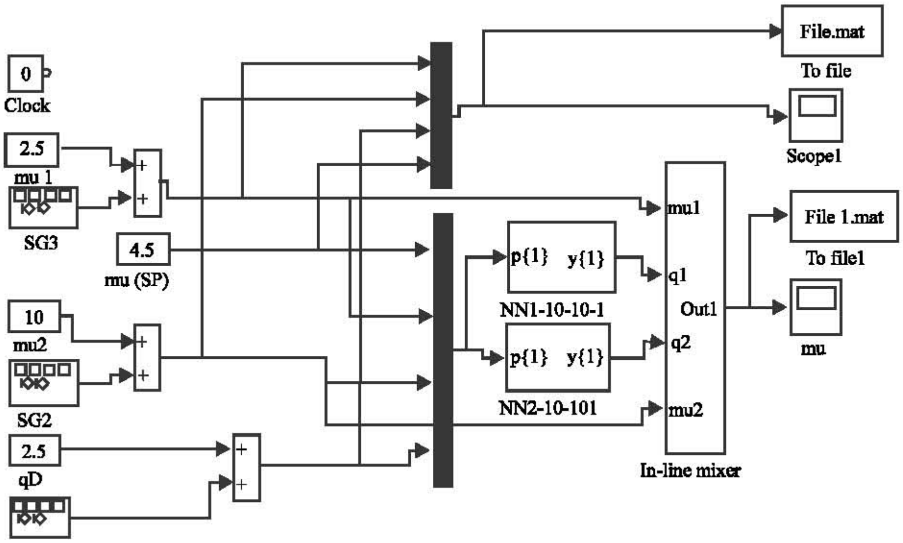

Fig. 9: Simulink block diagram for In-line mixer controller simulation

provides the setpoint flow rates $q_{1(SP)}$ and $q_{2(SP)}$. Necessary to achieve the desired flow rate and viscosity at the In-line mixer output. The results are shown in Fig. 10 where the actual viscosity μ is illustrated in centistokes (cts). The value is into the range of 4.35-4.65 under changes in of input variables or disturbances along its full ranges.

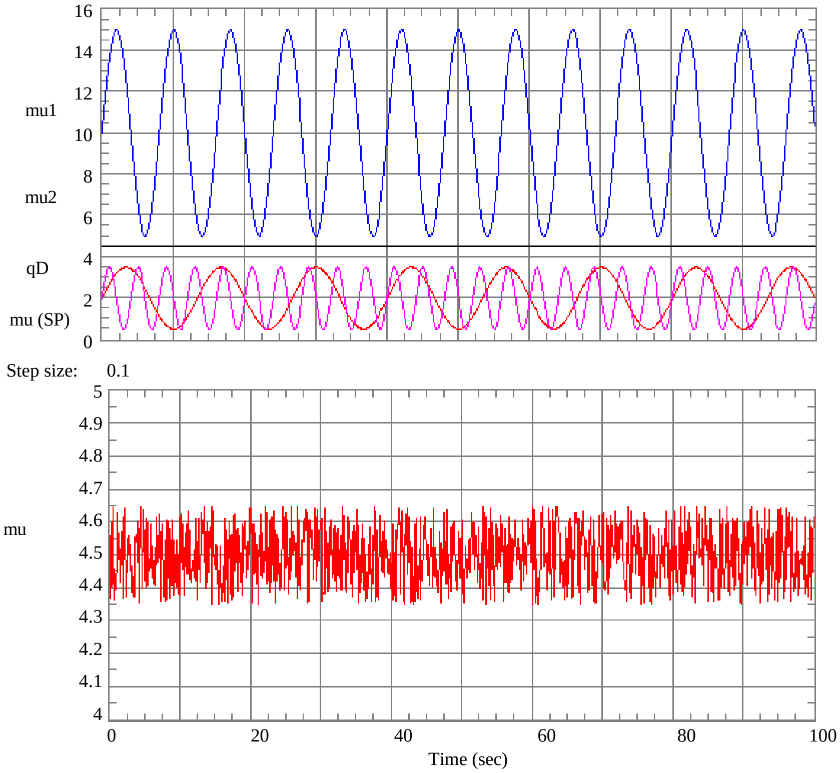

Fig. 10: Input-output signals of In-line mixer controller simulation

CONCLUSION

Multi-variable control strategies to solve efficiently linear and nonlinear in-line mixing problems have been presented.

These contributions are based on two computed multi-variable control techniques: Computed multi-variable control using the developed FCC strategy Computed multi-variable control using the developed EFC strategy Model based functional approximation was implemented on the basis of feedforward neural networks trained by means of a Backpropagation training algorithm: the conjugate gradient of Fletcher-Reeves.

With regard to results, the value of viscosity achieved by the In-line mixer shown in Fig. 9 is into the range of acceptable and expected results. Extensive simulation tests on EFC strategy yields the same results as FCC which is shown.

With regard to performance, rapid response, precision and robustness under changes (disturbances) of input variables are the main topics. A relevant advantage that can be pointed out is the sake of the lack of a conventional mixing tank and the lack of feedback measuring devices due to proposed strategies, minimum time delay on the mixing process control is achieved.

REFERENCES

Andrasik, A., A. Meszaros and S.F. de Azevedo, 2004. On-line tuning of a neural PID controller based on plant hybrid modelling. Comput. Chem. Eng., 28: 1499-1509.

Beale, E.M.L., 1972. A Derivation of Conjugate Gradients. In: Numerical Methods for Non Linear Optimisation, Lootsma, F.A. (Ed.). Academic Press, London, pp: 39-43.

Charalambous, C., 1992. Conjugate gradient algorithm for efficient training of artificial neural networks. IEEE. Proc., 139: 301-310.

Demuth, H. and M. Beale, 2000. Neural Network Toolbox for Use with MATLAB, User Guide. MathWorks Inc., United Kingdom.

Fletcher, R. and C.M. Reeves, 1964. Function minimization by conjugate gradients. Comput. J., 7: 149-154.

Hagan, M.T. and M. Menhaj, 1994. Training feed forward networks with the Marquardt algorithm. IEEE Trans. Neural Networks, 5: 989-993.

Hagan, M.T., H.B. Demuth and M.H. Beale, 1996. Neural Network Design. 1st Edn., PWS Publishing Co., Boston, MA, USA., ISBN: 0-53494332-2.

Hoyle, D.L. and G.K. McMillan, 1995. pH Control in Instrument Engineers Handbook. In: Process Control, Chief, B.L. (Ed.). Bitterworth-Heinemann, UK., pp: 1328-1361.

Luyben, W.L., 1990. Process Modelling, Simulation and Control for Chemical Engineers. McGraw-Hill Inc., USA., pp: 257-259.

Mahuli, S.K., R.R. Rhinehart and J.B. Riggs, 1992. Nonlinear model based control of pH. AIChE Annual Meeting, Paper 149g.

Moller, F.M., 1993. A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks, 6: 525-533.

Powell, M.J.D., 1977. Restart procedures for the conjugate gradient method. Math. Program., 12: 241-254.

Stern, H., E.E. Lewis and J.R. Granan, 1958. Control Engineers Handbook. In: Hydraulic Controls, Truxal Jhon G. (Ed.). McGraw-Hill Inc., USA., pp: 15.1-15.10.

Wachira, D., P. Thitiyasook, A. Arpornwichanop, P. Kittisupakorn and M.A. Hussain, 2005. Neural network inverse model-based controller for the control of a steel pickling process. Comp. Chem. Eng., 29: 2110-2119.

Widrow, B. and M.E. Hoff, 1960. Adaptive Switching Circuits. 1960 IRE WESCON Convention Record, New York, pp: 96-104.