

Genetic algorithms for local controller network construction

S.K. Sharma, S.F. McLoone and G.W. Irwin

Abstract: Local Controller Networks (LCNs) provide nonlinear control by interpolating between a set of locally valid, subcontrollers covering the operating range of the plant. Constructing such networks typically requires knowledge of valid local models. This paper describes a new genetic learning approach to the construction of LCNs directly from the dynamic equations of the plant, or from modelling data. The advantage is that *a priori* knowledge about valid local models is not needed. In addition to allowing simultaneous optimisation of both the controller and validation function parameters, the approach aids transparency by ensuring that each local controller acts independently of the rest at its operating point. It thus is valuable for simultaneous design of the LCNs and identification of the operating regimes of an unknown plant. Application results from a highly nonlinear pH neutralisation process and its associated neural network representation are utilised to illustrate these issues.

1 Introduction

Local Controller Networks (LCNs) provide a divide-and-conquer approach to the design of global controllers for complex nonlinear systems. The operating space is decomposed into a number of regimes and the required global controller is then formed by interpolating between simpler sub-controllers that are locally valid [1, 2]. Knowledge of these local operating regimes is therefore a key requirement for building such controllers. It is known that a large class of nonlinear systems can be controlled in this way, including most batch processes and many control system applications [3]. Apart from normal operation, the control system may also have to function correctly during startup and shutdown cycles and to operate during maintenance and fault conditions, all of which constitute different operating regimes. In most applications, the design of a suitable LCN is done on the basis of *a priori* plant knowledge [4–11]. The interpolation is done such that the local controller which is most valid at an operating point will be given the greatest weight, neighbouring ones will be weighted less, and those for distant operating regimes will not contribute anything to the global control at that point. Each local controller is thus associated with a weighting function that provides smooth interpolation and also indicates the relative validity of each local controller at a given operating point.

In [6] and [7], linear local models were first designed at each *a priori* known operating point and interpolated with

normalised Gaussian validity functions to provide a global model of a non-linear plant. A separate linear controller for each local model was then designed and interpolated with the same validity functions to construct the global controller. The validity functions thus only allow a single linear controller to act at an operating point and facilitate smooth transition to a different controller when the operating point changes. This approach works well when the local operating regimes are known but, in practice, their identification is difficult. The complexity of the target mapping, the representational ability of the local models associated with the validity functions and the availability of data all have to be taken into account [7]. Genetic algorithms have been successfully applied [12, 13] to simultaneously identify the optimum number of local operating regimes and to construct the parameters of local ARX models and their validity functions from plant data. It was also shown in [12] that a GA could be used to design a global controller, with the same validity functions as the local models, by constructing all the local controllers together. This contrasted to earlier approaches where each local controller was designed separately. In all the methods mentioned above, knowledge of the local models and their validity functions at each operating point is required in order to design the local controllers.

This paper extends some preliminary work by the authors [14] on LCN construction using GAs to minimise a mean-square-error criterion. However, no consideration was given in this case to dynamic performance and robustness in the construction of a LCN. This paper introduces a new fitness function for a GA to ensure from the outset that the LCN follows the setpoint trajectory with an acceptable rise-time, overshoot and settling time. A method for selecting a setpoint trajectory is also introduced to cover all the relevant process operating points and the robustness of the LCN controller is illustrated. Further, a procedure for adjusting the step duration in the setpoint trajectory is introduced to produce faster training by the GA. To achieve this a genetic algorithm, enhanced by fuzzy coding [15], is used. The fitness function is selected so that the controller validity functions do not overlap too much, hence producing local controllers that are 'almost orthogonal' [1] to one another.

© IEE, 2005

IEE Proceedings online no. 20045110

doi: 10.1049/ip-cta:20045110

Paper first received 7th July 2004 and in revised form 22nd March 2005

S.K. Sharma is with School of Engineering, University of Plymouth, Plymouth PL4 8AA, UK

S.F. McLoone is with Department of Electronic Engineering, National University of Ireland, Maynooth, Ireland

G.W. Irwin is with School of EE Engineering, Queen's University, Belfast BT9 5AH, UK

E-mail: sanjay.sharma@plymouth.ac.uk

Each local controller can then act independently within its own operating regime without disturbing the others. As well as offering transparency, this facilitates the identification of distinct operating regimes, which is important for practical applications [7, 10].

To show the effectiveness of the proposed scheme, a highly nonlinear pH process is employed. The LCN is developed for a pH process represented by dynamic equations [16] and shown to generate stable control actions spread throughout the required operating region. Since, dynamic equations of the plant are often not available in practice, a neural black-box model of the pH process was thus also considered for LCN construction, with similar results.

2 Local controller networks (LCNs)

The general discrete LCN representation is shown in Fig. 1. Here, the same inputs, \mathbf{x} , are fed to all the controllers and the outputs are weighted according to some scheduling variable or variables, ψ . The LCN output \hat{y} is given by the weighted sum:

$$\hat{y} = \sum_{i=1}^N \rho_i(\psi) f_i(\mathbf{x}) \quad (1)$$

where $\rho_i(\psi)$ is the validity or interpolation function associated with the i th local controller, $f_i(\mathbf{x})$ and N is the number of local controllers.

The validity functions $\rho_i(\psi)$ are normalised so that the total contribution from all the local controllers is 100%. The most widely used $\rho_i(\psi)$ in the literature are normalised Gaussian functions represented as:

$$\rho_i(\psi) = \frac{\exp(-\|\psi - s_i\|^2 / 2\sigma_i^2)}{\sum_{j=1}^N \exp(-\|\psi - s_j\|^2 / 2\sigma_j^2)} \quad (2)$$

while the $f_i(\mathbf{x})$ are linear controllers. Second order ARX-type linear controllers were used in [13] for $f_i(\mathbf{x})$. With this formulation both local controller parameters and initial local controller states had to be optimised. This resulted in a large search space and consequently long training times for the GA. To avoid the controller initialization problem and reduce the GA search space, discrete-type PID controllers are considered for LCN construction, in this paper.

A continuous-time PID-control law is described by

$$u(t) = k_p e(t) + k_d \dot{e}(t) + k_I \int e dt \quad (3)$$

where k_p , k_d and k_I are the proportional, differential and the integral gains, u is the control action and e is the error. The equivalent PID controller in discrete form is

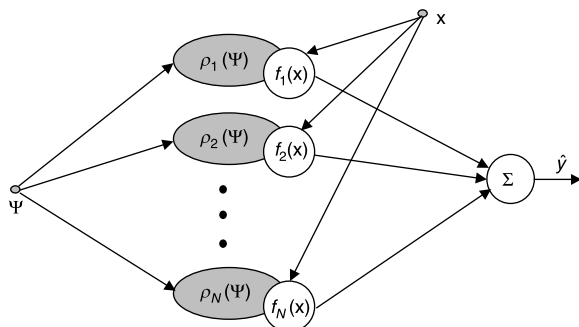


Fig. 1 General architecture of local controller network

$$u(k) = u(k-1) + k_p[e(k) - e(k-1)] + k_d[e(k) - 2e(k-1) + e(k-2)] + T_s k_I e(k) \quad (4)$$

where k is the sample number and T_s is the sampling interval.

For (2) and (4) the unknown parameters in each regime are the validity function centres, s_i , standard deviations σ_i and the PID-control parameters k_p , k_d and k_I . The identification of local operating regimes and simultaneous design of a global LCN is difficult in the absence of *a priori* knowledge about the unknown plant. This paper introduces a new approach for finding such operating regimes and for the construction of a suitable LCN based on genetic algorithms.

3 Genetic construction of LCN

3.1 Fuzzy coding

Fuzzy coding is an indirect method of encoding a real parameter, where a parameter is selected from appropriate fuzzy sets depending upon its contribution within a problem domain. It has been shown that this type of encoding of real parameters can result in increased convergence speed and better accuracy in the final solution [15].

Suppose each parameter has five associated fuzzy sets NM, NS, ZR, PS and PM. The corresponding degrees of membership $NM(x)$, $NS(x)$, $ZR(x)$, $PS(x)$ and $PM(x)$ evaluated at x then represent the degrees of firing (μ). Three types of membership functions, fuzzy sigmoid (FS), fuzzy Gaussian (FG) and fuzzy normal (FN), within the desired range (L, U) of a parameter value are used to represent these five fuzzy sets, as shown in Fig. 2. The FS shape provides global support in the sense that it divides the feature space into two halves, one where the response is approaching 1 for increasing value of parameters and another where it is approaching 0 for decreasing values (Fig. 3). The FG membership function has local support and divides the feature space into two halves in the opposite sense to FS , while FN is of linear shape and is selected to provide a distribution with support lying somewhere between local and global (Fig. 3).

Instead of providing a real value of a parameter the GA selects the optimum number of fuzzy sets and the μ associated with the chosen shape of membership function. Each chromosome parameter then takes a value between L and U , depending upon the type of membership function, the fuzzy sets selected and the corresponding μ values.

Figure 4 shows a chromosome representation in fuzzy coding where each parameter is represented by two sections. In the first, the fuzzy sets associated with each parameter are binary encoded, with a '1' representing the corresponding set selected. Here each parameter has five associated fuzzy sets NM, NS, ZR, PS and PM. In the second section, each parameter contains the degrees of membership $NM(x)$, $NZ(x)$, $ZR(x)$, $PS(x)$ and $PM(x)$ evaluated at x for each of these fuzzy sets. These are encoded as real numbers and represent the degrees of firing.

The GA now optimises μ and the number of fuzzy sets, while the actual parameter value of interest is obtained through defuzzification. Thus, every parameter can easily have different ranges by proper choice of the L and U values. The selection of the membership functions to use for a particular parameter depends upon the nature of its contribution in a given problem. A detailed analysis of this issue can be found in our previous paper [15].

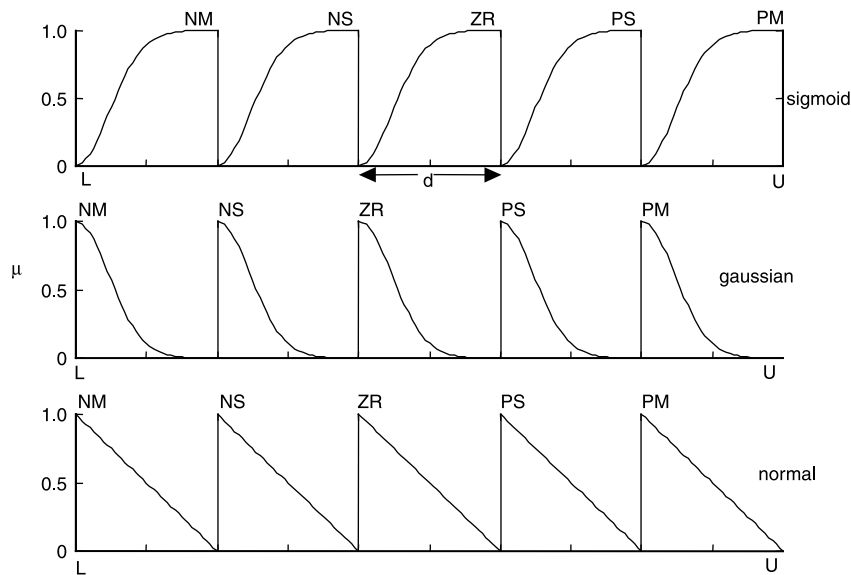


Fig. 2 Membership functions for FS, FG and FN distributions

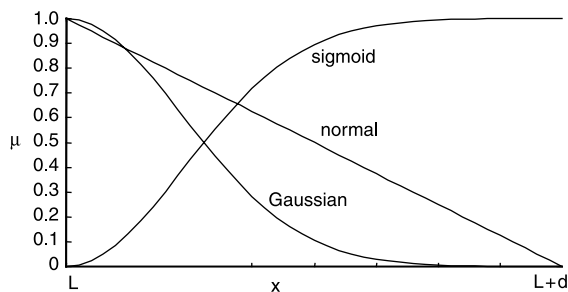


Fig. 3 FS, FN and FG distributions for the NM membership function

The genetic representation of LCN based on fuzzy coding is described next.

3.2 Genetic representation of LCN

Genetic algorithms can simultaneously identify the number of local controllers and the unknown parameters in (2) and (4). The genetic representation of the LCN is shown in Fig. 5 where each individual in the population is composed of two segments of strings. The higher segment is coded in binary and represents the number of local controllers, where a '1' denotes that a particular local controller is selected. The total number of bits in this first segment equals the total number of local controllers considered. In the application investigated here 10 controllers are initially retained. The lower segment is a fuzzy coding to represent the parameters associated with each local controller and its Gaussian validity function.

The GA then simultaneously searches for the optimal number of local controllers (from a given maximum number), the parameters of these local controllers and the parameters of the validity functions. Rather than selecting the parameters randomly, when a particular local controller is reselected, the weight values previously found are again used in order to produce better convergence. The genetic learning scheme applied to the encoded chromosomes is now described.

3.3 Genetic learning scheme for LCN

The encoded chromosomes representing the LCNs were randomly initialised for genetic evaluation. The fitness function in the GA was selected so that preference was

given to the chromosomes that correspond to LCN with fewer valid local controllers and which follows the setpoint trajectory with faster rise-time, lower overshoot and shorter settling time. The validity of local controllers was determined by $\rho_i(\psi)$, with a value of $\rho_i(\psi) = 1$ in a particular operating region indicating a valid local controller.

Each chromosome, corresponding to a LCN, can follow the step trajectory in many ways. Figs. 6a, 6b and 6c shows the three possible closed-loop step responses considered, with a total of N_r reference trajectory samples. Fig. 6a represents an under-damped response trajectory, Fig. 6b shows a less under-damped one while Fig. 6c is the critically or over-damped case. Chromosomes representing other response trajectories will take more time to produce a final solution which follows the setpoint trajectory with acceptable rise-time, overshoot and settling time. Therefore, at each generation those chromosomes which produce one of these three setpoint trajectories were retained in the population. The rest of the chromosomes were randomly reinitialised in the genetic evolution process. This ensures that, right from the beginning, the population contains more chromosomes with the required features which will lead to a faster convergence of the GA.

The errors E corresponding to these three trajectories were as defined in (5), (6) and (7) respectively.

$$E = (e_3 - e_2)/(e_1 - e_2) \quad (5)$$

$$E = -e_2/e_1 \quad (6)$$

$$E = \frac{1}{N_r} \sum_{i=1}^{N_r} \frac{S_i - R_i}{S_i} \quad (7)$$

where e_1 , e_2 and e_3 are the first overshoot, first undershoot and second overshoot respectively. S_i and R_i are the values of setpoint and response trajectory at sample i . A smaller value of E reduces damping and overshoot in the first two trajectories while improving the rise-time in the third. In all cases the E values lay in the range 0 to 1.

The analytical expression for the fitness function of an individual i in the population was of the form

$$J_i = \frac{P_i}{(1 + E)(TC_i)} \quad (8)$$

where TC_i is the total number of local controllers and P_i is a positive weighting. The P_i for a given LCN was calculated

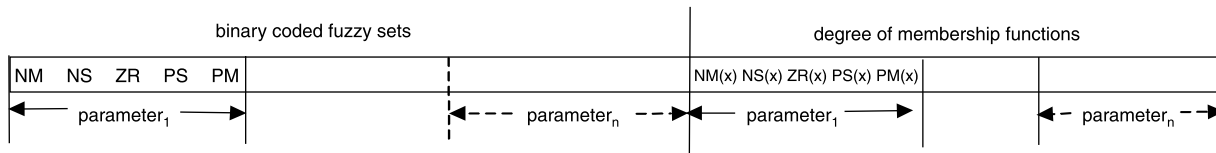


Fig. 4 Fuzzy coding representation of a chromosome

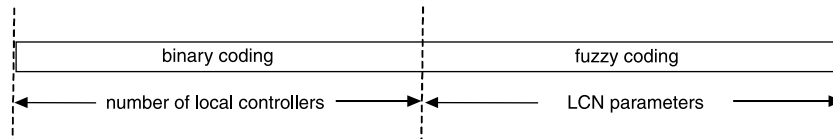


Fig. 5 Genetic representation of the LCN

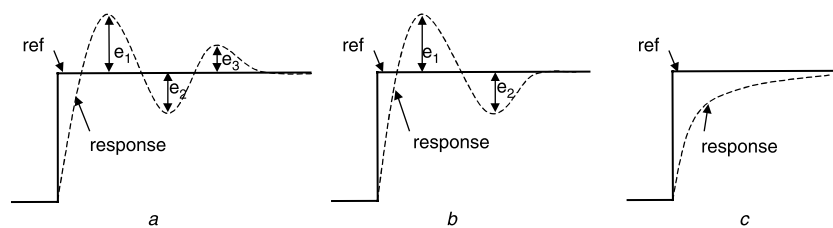


Fig. 6 Trajectory considered in a fitness function of GA

- a Under-damped response
- b Less under-damped response
- c Critically damped or over-damped response

at each sample as the total number of points for which $\rho_i(\psi) = 1$, within the operating region. This helped to promote transparency by encouraging all valid local controllers to be mutually orthogonal, such that each local controller acted independently of the rest at its operating point. The next Section describes the genetic operators applied with this learning scheme.

3.4 GA operators

In this paper a steady-state GA [17], with a mutation probability (p_m) of 0.01 and crossover probability (p_c) of 0.65, was applied to an initial population of 20 chromosomes. Five fuzzy sets were selected to represent a parameter in fuzzy coding for LCN construction and NN modelling. For each successive generation the two best chromosomes in the population were retained, producing offspring to replace the two worst chromosomes. Crossover and mutation, with the following modifications, were applied to the whole population. Uniform crossover [18], where bits were interchanged on the basis of a generated mask, was applied to the binary coded part. In the fuzzy coded part, the GA operators were applied between selected chromosomes, as detailed in reference [15]. In the fuzzy coding, the values for Gaussian centres and covariances were selected within the operating ranges of scheduled variables (ψ) while the ranges for the local PID-controllers parameters lay between ± 2 .

4 The pH neutralisation process

A schematic diagram of the pH neutralization process is shown in Fig. 7. Acid, buffer and base streams are mixed in a tank and the effluent pH is measured. This is a highly nonlinear process and hence offers a suitable application for the demonstration and evaluation of LCN techniques. The non-linearity in the process can readily be seen from the open-loop responses to $\pm 10\%$ step changes in base flow rate (q_3). The process gain varies by more than 250% for these

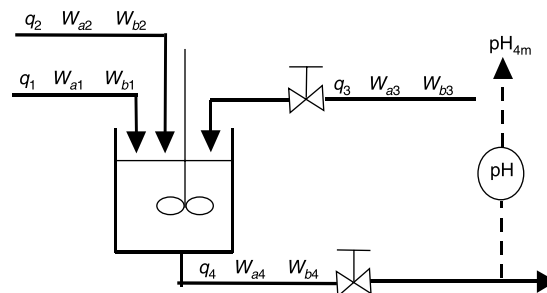


Fig. 7 pH neutralisation system

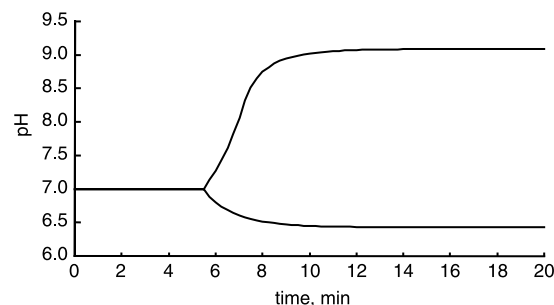


Fig. 8 Open-loop response of the pH neutralisation system for step changes in the base flowrates

small input changes as illustrated in Fig. 8. This chemical process was described by three nonlinear differential equations and a nonlinear output equation for the pH [16]:

$$\dot{h} = \frac{1}{A}(q_1 + q_2 + q_3 - C_v h^{0.5}), \quad (9)$$

Table 1: Nominal pH system operating conditions [16]

A (cross-sectional area) = 207 cm ²	W _{b2} (reaction invariant) = 3 × 10 ⁻² M
C _v (volume constant) = 8.75 ml cm ⁻¹ s ⁻¹	W _{b3} (reaction invariant) = 5 × 10 ⁻⁵ M
pK ₁ (constant) = 6.35	q ₁ (acid flow rate) = 16.6 ml s ⁻¹
pK ₂ (constant) = 10.25	q ₂ (buffer flow rate) = 0.55 ml s ⁻¹
W _{a1} (reaction invariant) = 3 × 10 ⁻³ M	q ₃ (base flow rate) = 15.6 ml s ⁻¹
W _{a2} (reaction invariant) = -3 × 10 ⁻² M	h (liquid level) = 14.0 cm
W _{a3} (reaction invariant) = -3.05 × 10 ⁻³ M	pH ₄ (pH value) = 7.0
W _{b1} (reaction invariant) = 0	θ (output time delay) = 0.5 min

$$\dot{W}_{a4} = \frac{1}{Ah} \left[(W_{a1} - W_{a4})q_1 + (W_{a2} - W_{a4})q_2 + (W_{a3} - W_{a4})q_3 \right] \quad (10)$$

$$\dot{W}_{b4} = \frac{1}{Ah} \left[(W_{b1} - W_{b4})q_1 + (W_{b2} - W_{b4})q_2 + (W_{b3} - W_{b4})q_3 \right] \quad (11)$$

$$W_{a4} + 10^{pH_4-14} + W_{b4} \frac{1 + 2 \times 10^{pH_4-pK_2}}{1 + 10^{pK_1-pH_4} + 10^{pH_4-pK_2}} - 10^{-pH_4} = 0 \quad (12)$$

where h is the liquid level, W_{a4} and W_{b4} are the reaction invariants of the effluent stream, and q_1 , q_2 and q_3 are the acid, buffer and base flowrates respectively. The pH measurement is assumed to be delayed by a transportation lag β so $pH_{4m} = pH_4(t - \beta)$. Acid and buffer flow rates were kept constant at their nominal values, with the base flow rate manipulated to excite the plant dynamics across the operating range. The nominal model parameters and operating conditions used in the simulation studies are shown in Table 1. The output time delay is 0.5 min and the sampling period (T_s) was chosen as 0.25 min.

The construction of a LCN for this pH neutralisation process is described next.

5 Local controller network (LCN) design

Figure 9 shows the design of a LCN involving PID-type local controllers. Here the LCN consists of m PID-type local controllers. The output of the i th PID-type local controller at sample k is $c_i(k)$ and the overall LCN output is defined as $c(k) = \sum_{i=1}^m c_i(k)$. The control action applied to the pH process at sample k is given by $u(k) = c(k) + u(k - 1)$. The same error, $e(k) = r(k) - y(k)$, is applied to all local controllers in the network. The scheduling variable

for the validity function, $\rho_i(\psi)$, were chosen as $\psi = [y(k - 1), u(k - 1)]$, where $y(k)$ is the pH process output and $r(k)$ is the reference setpoint. Genetic algorithms were now used to construct a LCN for this process.

1000 random steps across the operating range of the base flow rate were first applied to the open-loop pH process. The step duration was chosen by trial-and-error and 75 min was found to be appropriate to allow the pH output to reach steady-state. The steady-state values of pH and the corresponding base flow rate were then plotted, as shown in Fig. 10. This is a typical titration curve for a pH neutralisation process and shows clearly the highly non-linear nature of the plant. From this titration curve the setpoint trajectory for training the LCN was determined as follows. Those points on the curve whose successive *slope* values were greater than 0.3 were chosen where *slope* is defined as, $slope = |\Delta pH_{4m}|/|\Delta q_3|$. The rectangles in Fig. 10 mark the selected points on the titration curve. All the important bends in the titration curve were effectively covered in this manner. With more than two variables, k-means clustering [19] can be used to select suitable points. These selected points were now taken at random and their corresponding pH values used to form a setpoint reference trajectory for designing the LCN. At each generation, the best chromosome was evaluated to represent a LCN and its response to the required setpoint was considered acceptable only if it was within 0.2% of the target value for the last 2 minutes of each step.

Since equal weight was not given to transient and steady-state performance, each LCN was unable to produce a good control performance for reference trajectory steps of the same duration. An interval of 10 minutes was found to be appropriate to achieve at least a target step and this was thus selected as the initial step duration for use in the setpoint trajectory. Starting with this initial step duration the GA strategy, as previously described, was run for 200 generations and the LCN response was evaluated for the chromosome with the best fitness value. If the response trajectory for a particular step duration failed to meet the

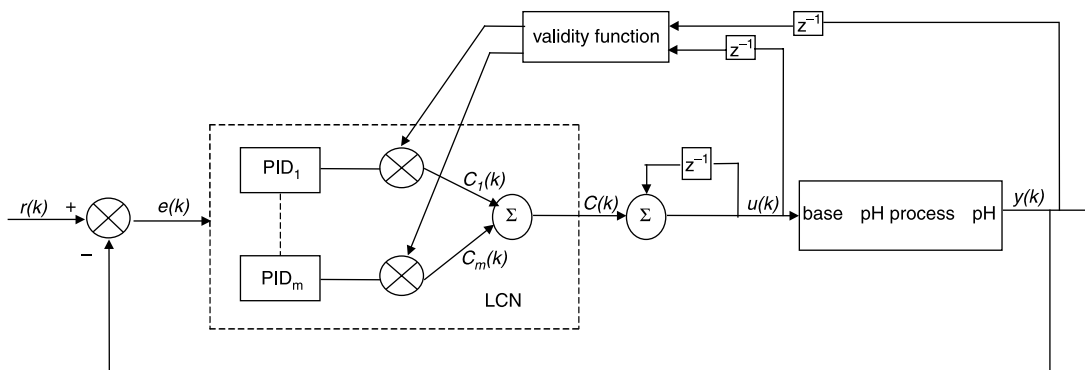


Fig. 9 PID-type local controller network (LCN) acting on a pH process

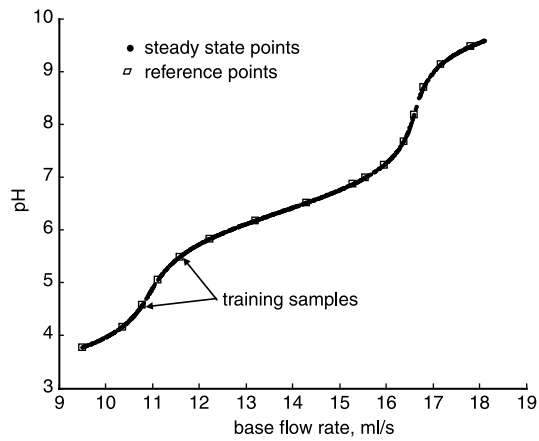


Fig. 10 Titration curve of a pH process for selecting a training data for the LCN design

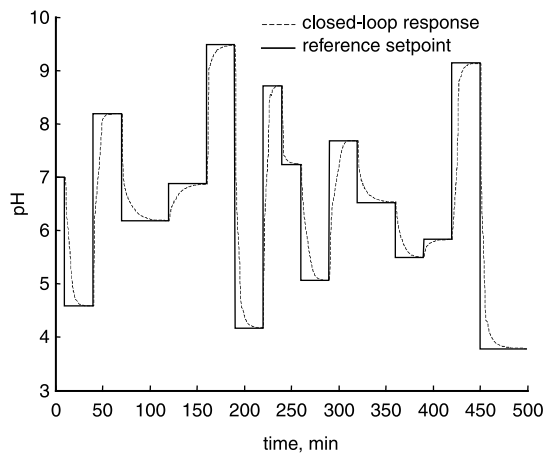


Fig. 11 Setpoint tracking by a trained LCN acting on the pH process

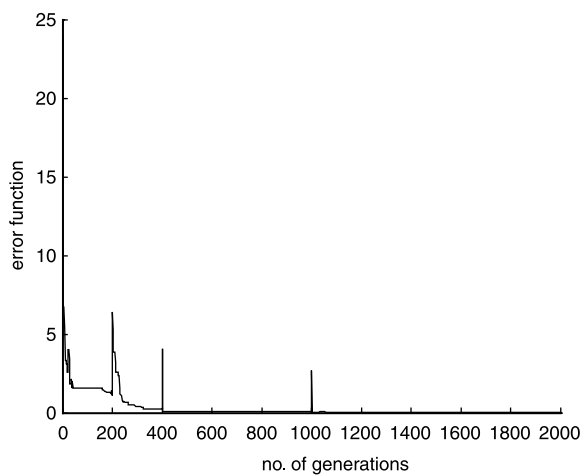


Fig. 12 Variation in error function with number of generations

required setpoint, this was increased by 5 min and the GA was again run for 200 generations with this new setpoint trajectory. This process was repeated until the response trajectory fulfilled the desired control performance criteria for all the steps in the setpoint trajectory.

A total of 20 chromosomes in a population were considered for genetic learning. At each generation, chromosomes were ranked according to fitness function defined in eqn. (8). After running the algorithms initially for 1000 generations, the number of local controllers from the best chromosome were selected as the optimum one and a new group of chromosomes was formed using only

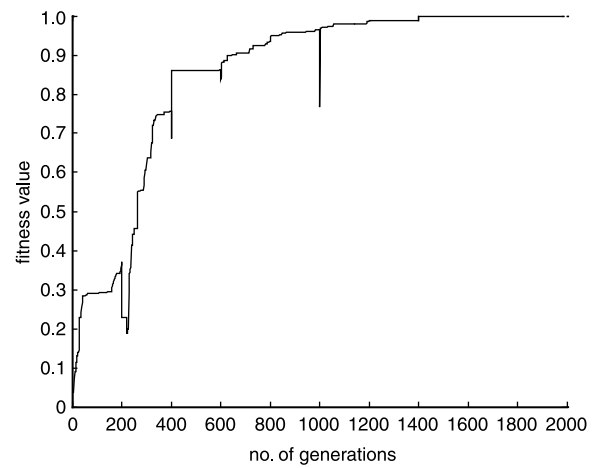


Fig. 13 Variation in fitness value with number of generations

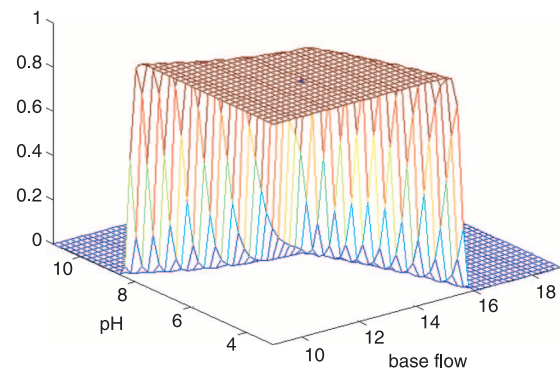


Fig. 14 Validity function ρ_1 for local controller C_1

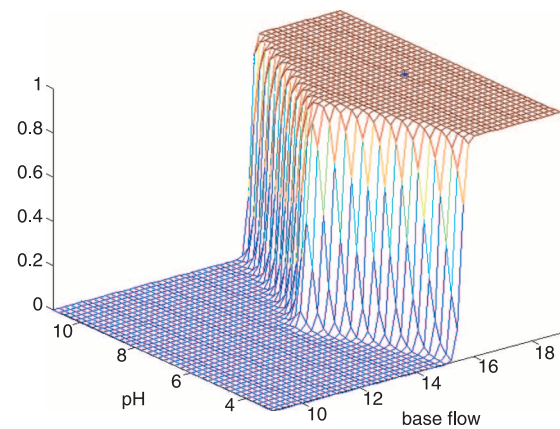


Fig. 15 Validity function ρ_2 for local controller C_2

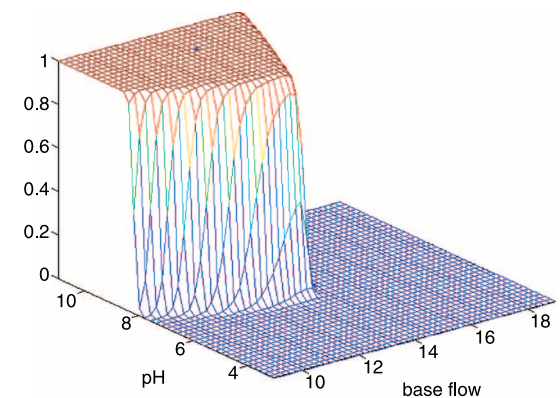


Fig. 16 Validity function ρ_3 for local controller C_3

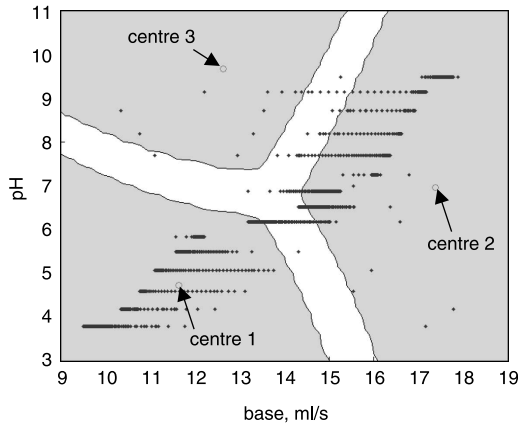


Fig. 17 Plan view of validity regions with training data

the parameter values for these local controllers, along with the centres and covariances of the associated validity function. Fine-tuning of these local controller parameters and their Gaussian interpolation functions was then performed through genetic evaluation by running the algorithm for a further 1000 generations. This two-step approach speeds up the learning process, as it is only necessary to deal with the subset of controllers selected from the initial search.

Figure 11 compares a typical setpoint trajectory and actual closed-loop response from a local controller

network trained in this manner and shows that good tracking performance is achieved. This graph also shows how the step duration of the setpoint signal has been adapted to achieve a balance between transient and steady state behaviour. Figures 12 and 13 show the variation of the error function and fitness value with respect to successive generations respectively. Finally, it is worth noting that in this case the GA construction process selected a LCN with three local controllers. This is consistent with previous results for this application [12] where three local models were shown to be adequate for LMN construction and subsequent LCN design. Figures 14, 15 and 16 illustrate the respective interpolation regions for these three local controllers. Further, Fig. 17 contains a plan view with the training data superimposed. It can be seen that the GA has selected the local controllers that are occupying 100% and thus transparent somewhere in the operating regions. The three PID-type local controllers defining the LCN were:

$$C_1(k) = 0.9448[e(k) - e(k-1)] - 0.2433[e(k) - 2e(k-1) + e(k-2)] + 0.0956e(k) \quad (13)$$

$$C_2(k) = 0.6164[e(k) - e(k-1)] + 0.3882[e(k) - 2e(k-1) + e(k-2)] + 0.0759e(k) \quad (14)$$

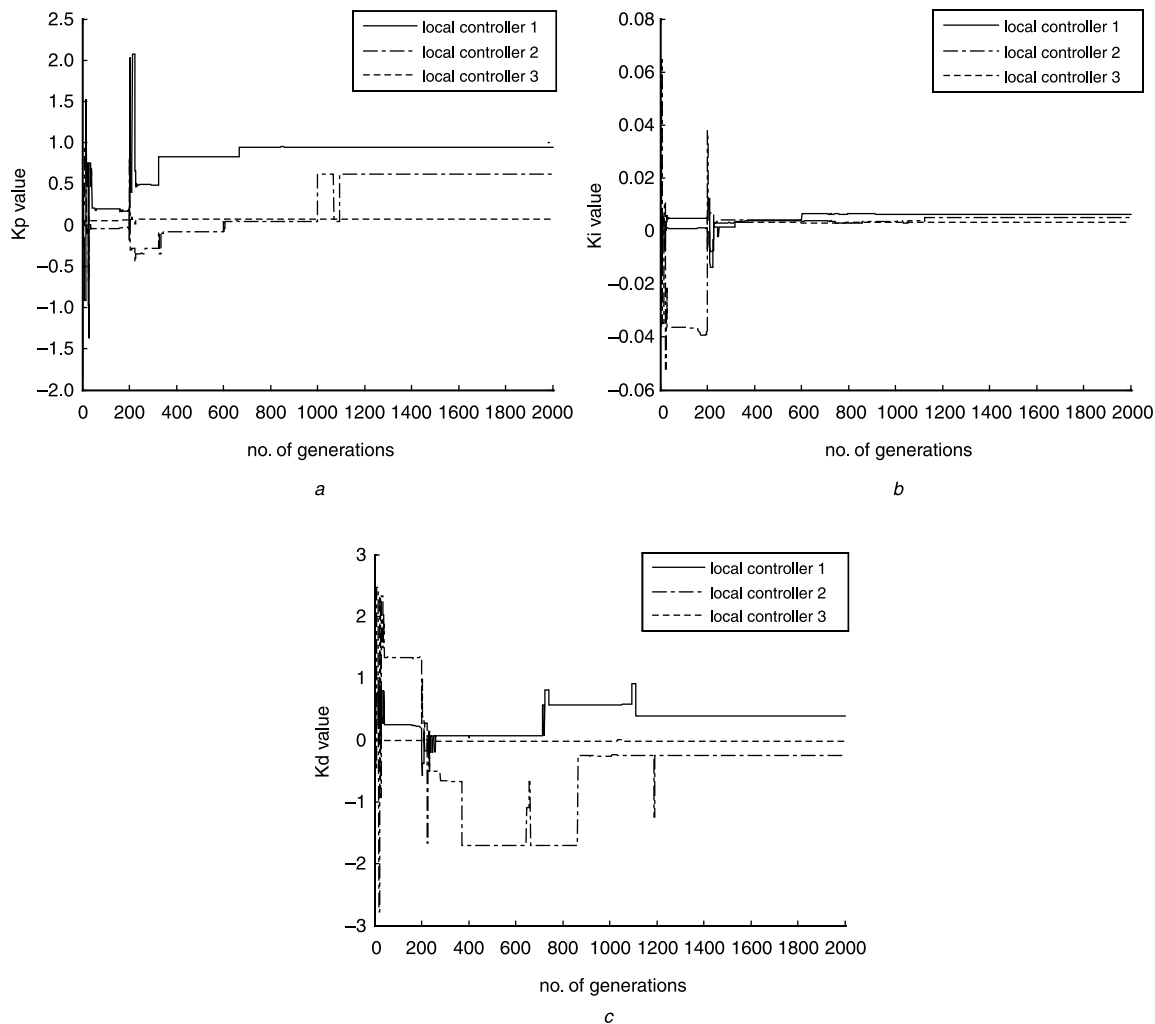


Fig. 18 Variation in PID gains with number of GA generations

- a k_p
- b k_i
- c k_d

$$C_3(k) = 0.0712[e(k) - e(k-1)] - 0.0209[e(k) - 2e(k-1) + e(k-2)] + 0.0499e(k) \quad (15)$$

where $e(k) = r(k) - y(k)$ is again the error between the reference and controller trajectory. The centres and covariances for the Gaussian interpolation functions were ρ_1 : (4.7272, 11.6520) and (0.5319, 0.6327), ρ_2 : (6.9552, 17.3752) and (0.6143, 0.7082) and ρ_3 : (9.6695, 12.6338) and (0.6252, 0.4898).

The convergence trends of PID gains with successive generations are shown in Figs. 18a, 18b and 18c respectively whereas Figs. 19a, 19b, 20a and 20b illustrate the corresponding variation of centres and widths of the Gaussian functions for pH and base flow rate respectively.

The robustness of the controller was next tested on two different, previously unseen, setpoint trajectories which spanned the entire operating region. The first consisted of pH steps of unequal height ranging from 0.5 to 5.0. The duration of each was selected to be 50 min to allow for the longest settling time in the process. Figure 21 confirms the excellent setpoint tracking produced by the LCN. Results in references [6, 7] and [11, 12] show that, while the stability

of a local controller acting on a local plant model is easily proved, the global stability of the overall closed-loop system is more problematic. Another way to test the robustness of the controller was to show that it was stable throughout the operating space of the pH process. To this end, a random sequence of 90 steps, each of 50 min duration, covering the entire pH operating region was generated. Figure 22 shows that the LCN generated a stable closed-loop response and smoothly followed the reference trajectory in this case also. Figure 23 is the plan view of the LCN with the control actions (base flow rate values) generated with the corresponding output pH value. This confirms that the LCN effectively covered the whole operating space and produced the required stable control action.

The design of a LCN for a NN black-box model of the pH process was now considered to account for the situation where the plant equations are unknown, but input and output-modelling data is available. A process model can first be developed using such data and then be used as the basis for LCN design. Separate training and test data sets [12] were used to create a multilayer-perceptron model of the pH process. A genetic algorithm with fuzzy encoding [15] was run for 5000 generations to obtain the weights and biases of the NN model with six inputs:

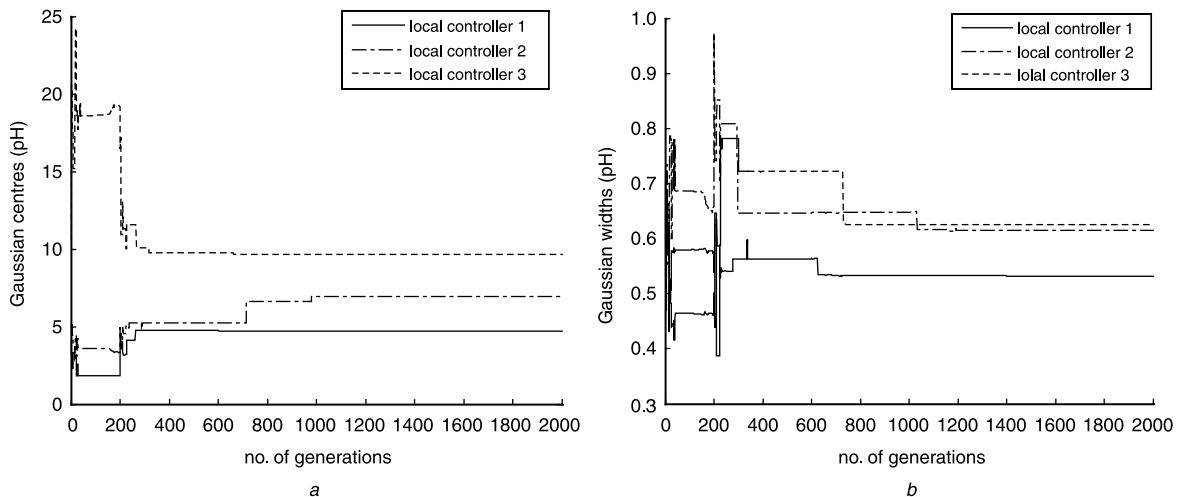


Fig. 19 Variation in Gaussian centres and widths variation with number of GA generations for pH value

a Gaussian centres
b Gaussian widths

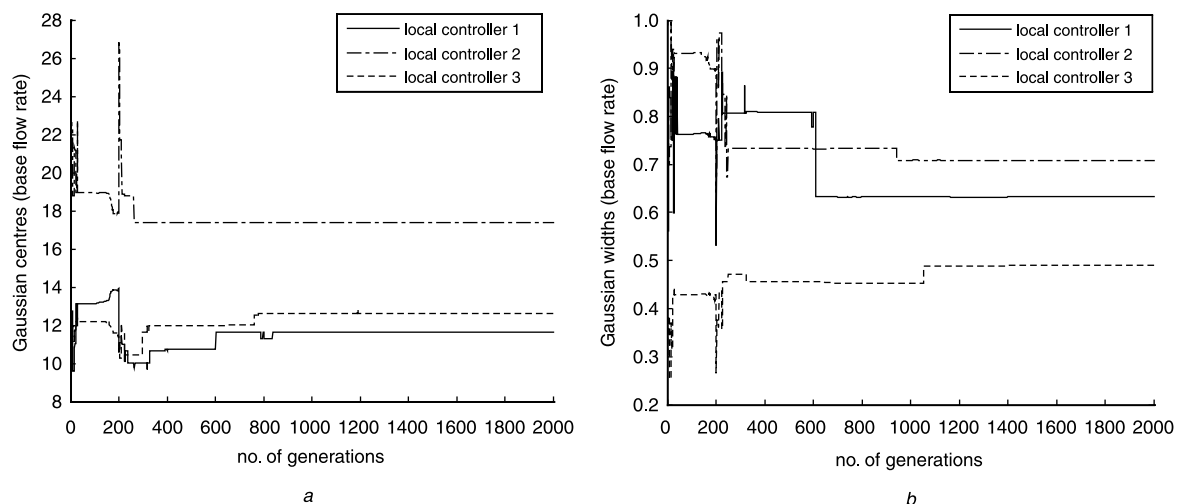


Fig. 20 Variation in Gaussian centres and widths with number of GA generations for base flow rate

a Gaussian centres
b Gaussian widths

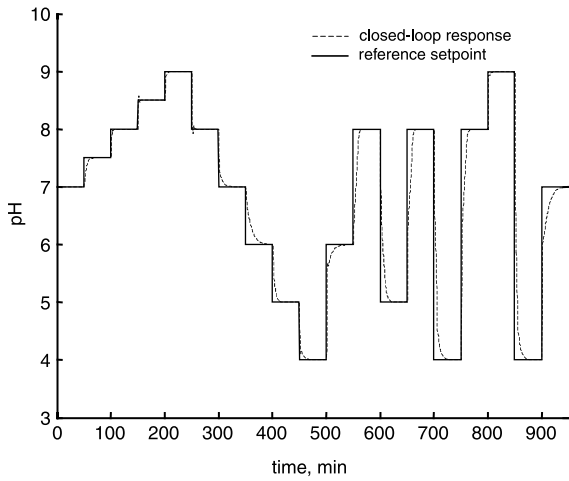


Fig. 21 Simple step trajectory of the LCN in the pH process study case

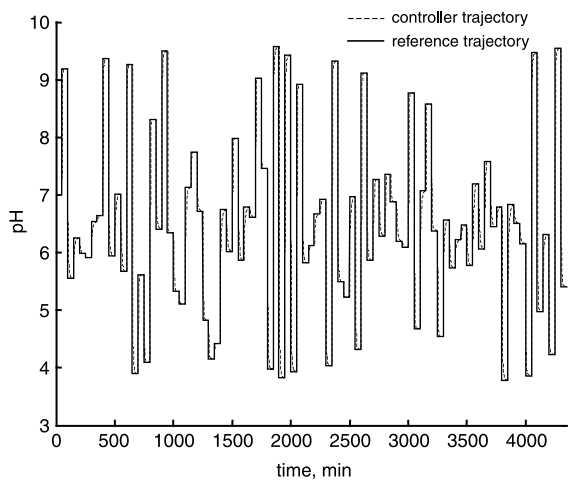


Fig. 22 Random step trajectory followed by the LCN in the pH process study

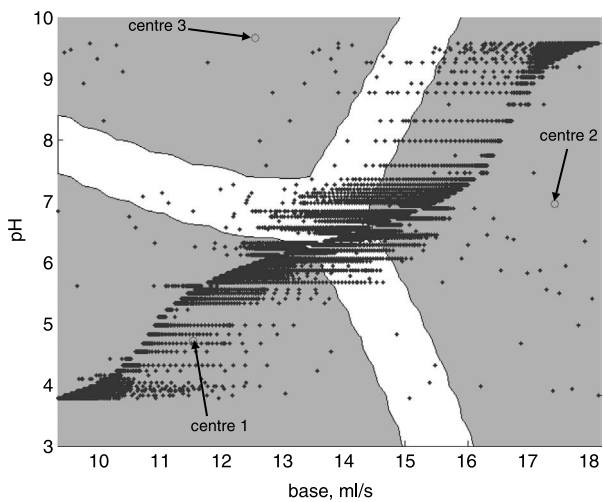
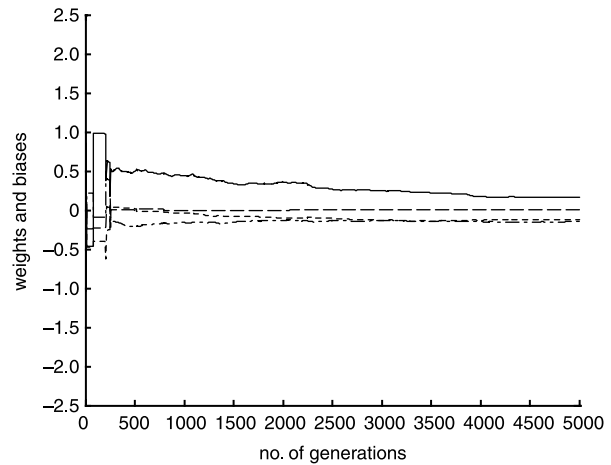


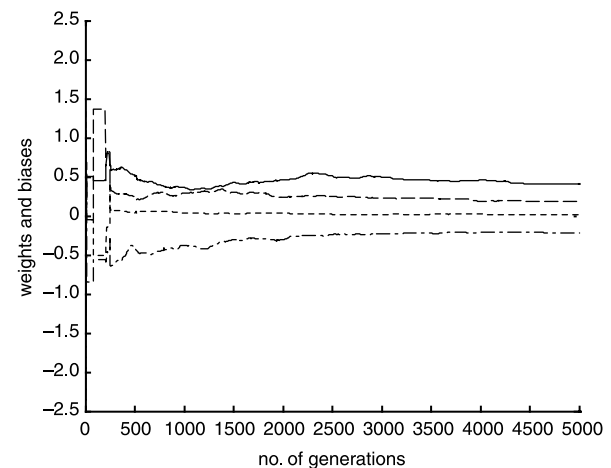
Fig. 23 Plan view of LCN with random steps data for the pH process

$$\begin{aligned} & [q_3(k), q_3(k-1), q_3(k-2), p\hat{H}_{4m}(k-1), \\ & p\hat{H}_{4m}(k-2), p\hat{H}_{4m}(k-3)] \end{aligned} \quad (16)$$

and 20 hidden nodes. The total number of weight and bias parameters in the NN model was 141 (120 between the input and hidden layer and 21 between the hidden and output



a



b

Fig. 24 Convergence of the NN model weight and bias parameters over successive GA generations

a Hidden layer parameters
b Output layer parameters

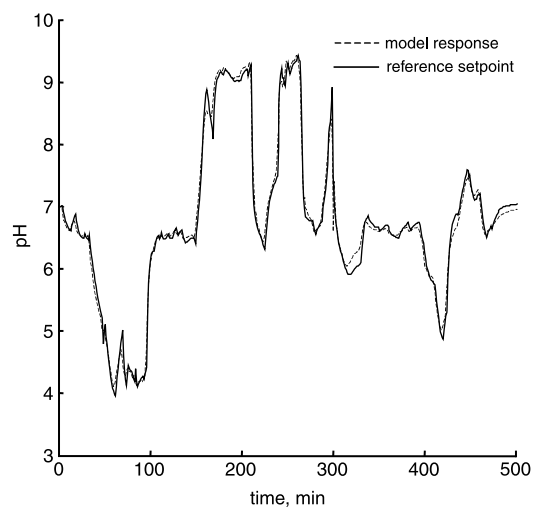


Fig. 25 NN model output on test data

layer). Figure 24a and 24b show typical network weight convergence trends obtained with GA learning for a random selection of these parameters. Figure 25 shows the parallel model performance of the NN model on the test data set which produced a mean-squared pH error of 0.0102. This NN model was now used in place of the actual pH process to train a LCN for the same setpoint trajectory as before.

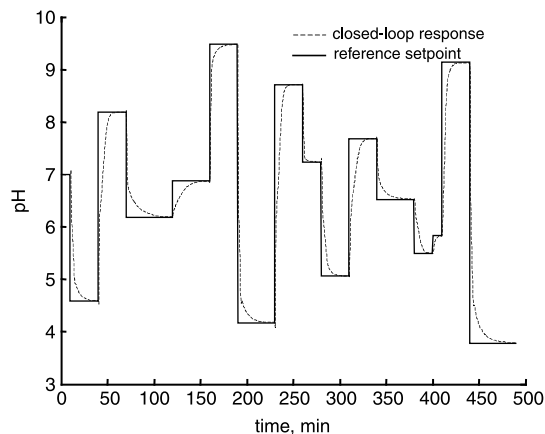


Fig. 26 Setpoint tracking of trained LCN in NN-model case

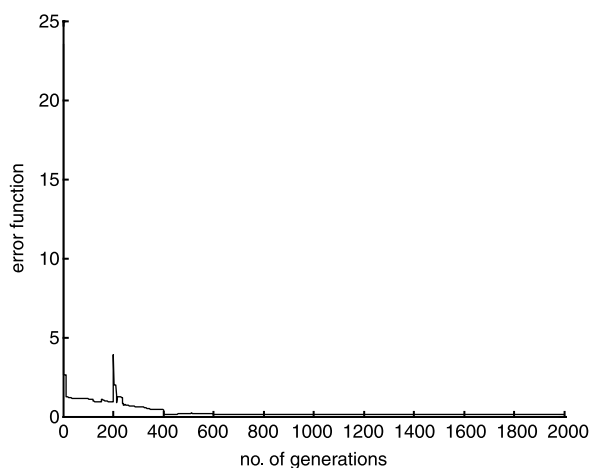


Fig. 27 Variation in the error function with number of generations

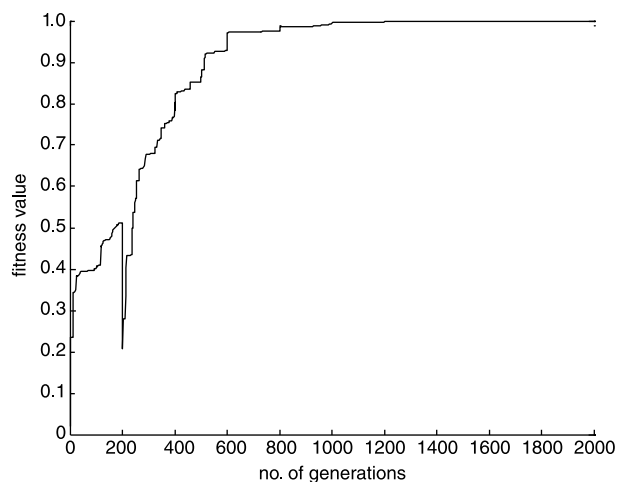


Fig. 28 Variation in fitness value with number of generations

The GA strategy described earlier was again applied to globally optimise the LCN parameters. Figure 26 shows the good setpoint tracking obtained with the resulting LCN. Note that, as before, the step durations have also been adapted during training. Figures 27 and 28 illustrate the variation of error function and fitness value with successive generations respectively in this case. Again the GA selected three local controllers for the pH process. Figure 29 shows the titration curve obtained in this case with the highlighted points indicating the data used for training. This confirms that the training data selected here was also well spread

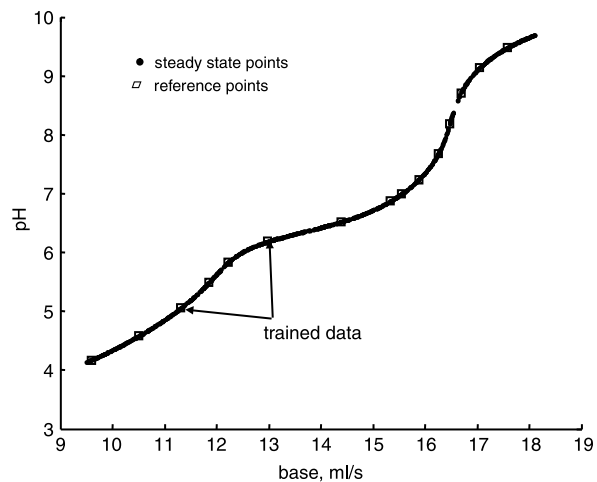


Fig. 29 Titration curve of a NN-model with trained data of the LCN

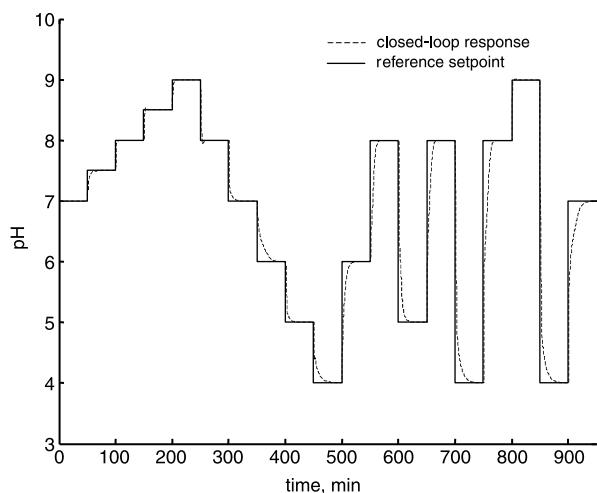


Fig. 30 Simple step trajectory followed by the LCN for the NN-model case

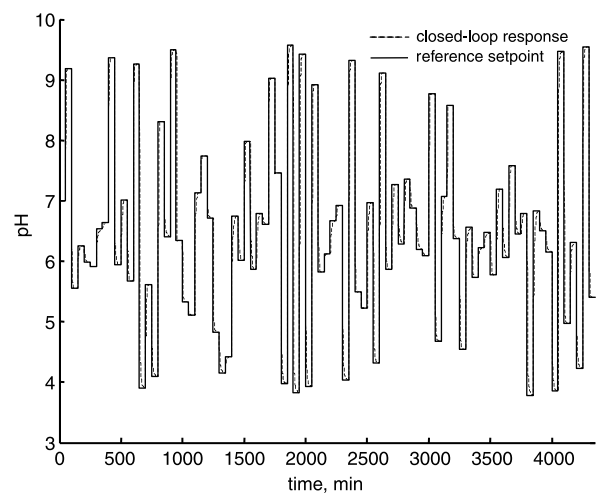


Fig. 31 Random step trajectory followed by the LCN for the NN-model case

across the curve, capturing all the important features of the nonlinearity. Figures 30 and 31 show the good setpoint tracking produced with known and random demands respectively. Figure 32 gives the corresponding plan-view for the random step case, showing the centres of the local controllers. This illustrates that the LCN generated a stable

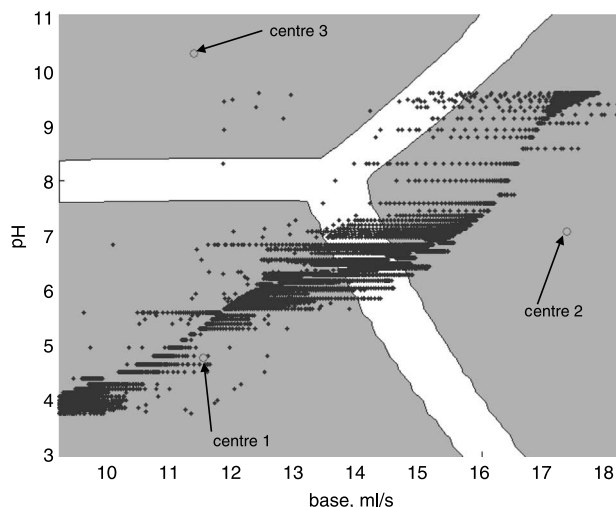


Fig. 32 Plan view of LCN with random step data in the NN-model case

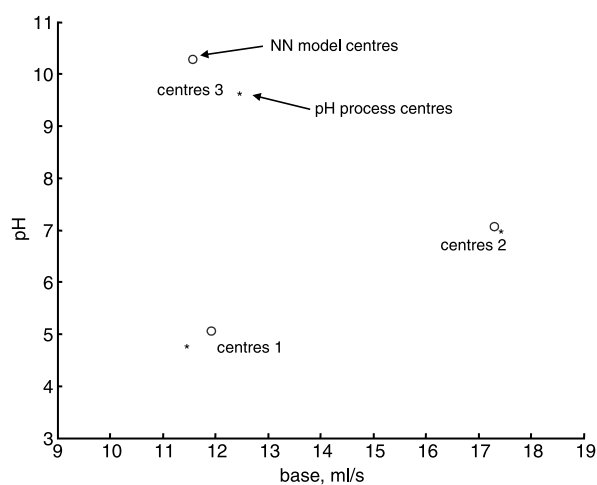


Fig. 33 Corresponding centres of the LCN for the pH process and NN-model

control action which spanned the entire operating space. Figure 33 compares the centres of the local controllers obtained using the pH process equations and the identified neural model. This confirms that the GA correctly found matching centres in both cases which is reassuring.

These results show the capacity of this new approach to designing a LCN for an unknown plant. A NN model of the unknown plant can be used directly to design a LCN in the absence of much *a priori* knowledge.

6 Conclusion

A genetic learning approach to the construction of LCNs has been proposed. The approach optimises the overall LCN structure and the local PID controller parameters and easily facilitates the inclusion of transparency and generalisation constraints. A fuzzy coding is used to provide accuracy and faster GA convergence in real parameter selection. Results for a pH neutralisation process illustrates the potential offered by GAs for constructing LCNs for nonlinear processes. This approach simplifies the earlier approaches where transparency and generalisation in the design of a

local linear controller was only possible with *a priori* knowledge of plant characteristics. The new approach is useful in simultaneously providing information on plant operating regions as well as designing robust local linear controllers for each region. Further, instead of requiring *a priori* knowledge of the plant or its local model representation, a much easier modelling approach like neural networks or fuzzy modelling can be utilised to construct a LCN for the nonlinear plant.

Future work will include the application of this approach to the design of a LCN for turbogenerator control [20] where both the gain and dynamics vary significantly across the operating space. A stability analysis of the global LCN on the basis of [21] will also be considered.

7 References

- Johansen, T.A., and Foss, B.A.: 'Constructing NARMAX models using ARMAX models', *Int. J. Control*, 1993, **58**, pp. 1125–1153
- Johansen, T.A., and Foss, B.A.: 'Semi-empirical modeling of nonlinear dynamic systems through identification of operating regimes and local models', *Neural Network Engineering in Dynamic Systems*, 1995, pp. 105–126
- Rippin, D.W.T.: 'Control of batch processes'. Preprints of the 3rd IFAC DYCORS + 'Symposium', Maastricht, Netherlands, 1989, pp. 115–125
- Townsend, S., and Irwin, G.W.: 'Nonlinear model based predictive control using multiple local models (2001) IEE, Control Engineering Book Series 61, 'Nonlinear predictive control; theory and practice', in Kouvaritakis, B., and Cannon, M. (Eds.), Chapter 11, pp. 223–243 2001, ISBN 0 8596 984 8
- McLoone, S.C., and Irwin, G.W.: 'On velocity-based local model methods for nonlinear identification', *Asian J Control*, 2003, **5**, (2), pp. 309–315
- Townsend, S., Lightbody, G., Brown, M.D., and Irwin, G.W.: 'Nonlinear dynamic matrix control using local model networks', *Trans. Inst. MC*, 1998, **20**, (1), pp. 47–56
- Brown, M.D., Lightbody, G., and Irwin, G.W.: 'Non-linear internal model control using local model networks', *IEE Proc. Control Theory Applic.*, 1997, **144**, (6), pp. 505–514
- Johansen, T.A., Foss, B.A., and Sorensen, A.V.: 'Nonlinear predictive control using local models – applied to a batch fermentation process', *Control Eng. Pract.*, 1995, **3**, (3), pp. 389–396
- Leith, D.J., and Leithead, W.E.: 'Analytical framework for blended multiple model systems using linear local models', *Int. J. Control*, 1999, **72**, pp. 605–619
- Murray-Smith, R., and Johansen, T.A.: 'Multiple model approaches to modeling and control' (Taylor and Francis, 1997)
- Brown, M.D., Lightbody, G., and Irwin, G.W.: 'Non-linear internal model control using local model networks', *IEE Proc. Control Theory Applic.*, 1997, **144**, (6), pp. 505–514
- Sharma, S.K., McLoone, S., and Irwin, G.W.: 'Genetic algorithms for local model and local controller network design'. Proc. American Control Conf. Anchorage, AK, 8–10 May 2002, pp. 1693–1698
- Sharma, S.K., Irwin, G.W., and McLoone, S.F.: 'Optimal construction of local model networks using genetic algorithms'. Proc. IFAC Conf. Emerging Computing Technologies, Hong Kong, Nov. 2001, pp. 590–595
- Sharma, S.K., McLoone, S.F., and Irwin, G.W.: 'Genetic algorithms for simultaneous identification of local operating regimes and local controller network design'. UKACC Control 2004 Conf., University of Bath, U.K., 6–9 September 2004, Paper no. 75, CDROM Proceeding
- Sharma, S.K., and Irwin, G.W.: 'Fuzzy coding of genetic algorithms', *IEEE Trans. Evolut. Comput.*, 2003, **7**, (4), pp. 344–355
- Hall, R.C., and Seborg, D.E.: 'Modelling and self-tuning control of a multivariable pH neutralisation process. Part I: Modelling and multi-loop control'. Proc. American Control Conf., Pittsburgh, 1989, pp. 1822–1827
- Michalewicz, Z.: 'Genetic algorithms + data structures = evolution programs' (Springer-Verlag, New York, 1994)
- Wasserman, P.D.: 'Advanced methods in neural computing' (Van Nostrand Reinhold, New York, 1993), pp. 93–94
- Fasulo, D.: 'An analysis of recent work on clustering algorithms', Technical Report 01-03-02, Department of Computer Science and Engineering, University of Washington, Seattle, WA 98195, April 1999
- Irwin, G.W., Brown, M., and Lightbody, G.: 'Nonlinear identification of turbogenerator by local model networks'. Proc IEE International Control Conf., Publication No. 455, 1998, pp. 1462–1467
- Bubnicki, Z.: 'General approach to stability and stabilization for a class of uncertain discrete non-linear systems', *Int. J. Control*, 2000, **73**, (14), pp. 1298–1306