Neural, Parallel & Scientific Computations 15 (2007) 221-238

Adaptive wavelet neural networks for non-linear modelling and control

V.S. Kodogiannis[1], I. Petrounias[2] and J.N. Lygouras[3]

[1] Centre for Systems Analysis, School of Computer Science, Univ. of Westminster, London, HA1 3TP, U.K

[2]School of Informatics, The University of Manchester, Manchester M60 1QD, UK

[3]Dept of Electrical & Computer Engineering, Democritus University of Thrace, Xanthi, GR-67100, Greece

Abstract

Feed-forward and recurrent neural networks have been successfully used for modelling and control of non-linear systems. The main features of these systems such as the ability to learn from examples and to self-adapt are very well suited for the multi-resolution approach intrinsic to wavelets. Wavelets offer an adequate framework for the representation of "natural" signals and images that are described by piece-wise smooth functions, with rather sharp transitions between neighbouring domains. The combination of wavelet theory and neural networks has lead to the development of wavelet networks (WNNs). WNNs are neural networks using wavelets as activation function, where both the position and the dilation of the wavelets are optimised besides the weights. Their strength lies in the capability of catching essential features in "frequency-rich" signals. In this paper an infinite impulse response (IIR) recurrent structure is combined in cascade to a WNN in a proposed controller-scheme. The effectiveness of the proposed controller is illustrated through an application to composition control in a continuously stirred tank reactor (CSTR) system. Simulation results demonstrate the applicability of the proposed design method to non-linear control systems.

Keywords: wavelet theory, neural networks, infinite impulse response, modelling, control

1. INTRODUCTION

The currently existing complex plants cannot be accurately described by traditional rigorous mathematical models, and there are increasing needs for highly accurate control and autonomous behaviour in control, robotics and artificial life communities. The conventional approaches for understanding and predicting the behaviour of such systems based on analytical techniques can prove to be inadequate. These difficulties lead to a number of challenging problems, *i.e.*, embed the human intelligence into a machine, because there is a huge gap between the human intelligence and the machine intelligence. With emerging development in neural networks (NN) and fuzzy logic technologies, non-linear modelling designs can expand to even greater horizons. Generally, in many applications multi/single layer feed-forward neural networks have demonstrated an amazing ability to *learn* the desired map from discrete data. A number of rigorous mathematical proofs have been provided to explain this uncanny ability of feed-forward neural networks to approximate maps [1].

Received December 19, 2005; final form May 24, 2007

In recent years, wavelets have become a very active subject in many scientific and engineering research areas. Especially WNNs, inspired by both the feed-forward neural networks and wavelet decompositions, have received considerable attention [2] and become a popular tool for function approximation. The main characteristic of WNN is that some kinds of wavelet functions are used as the nonlinear transformation function in the hidden layer, instead of the usual sigmoid function. Incorporating the time-frequency localisation properties of wavelets and the learning abilities of NNs, WNNs have shown its advantages over the regular methods such as NNs for complex nonlinear system modelling. Unlike the multilayer perceptron which is a *global network*, WNN is a *local network* in which the output function is well localised in both time and frequency domains. In a local network only a small subset of weights are active at each point in the output space and the training of the network in one part of the input space does not corrupt that which has already been learned in more distant regions. Thus the learning speed of the local network is generally much faster than the global network. The radial basis function (RBF) network is also an example of a *spatially local* network. In addition, *local recurrent* networks or *temporal local* networks such as centre recurrent, linear recurrent, and IIR recurrent structures can be used in cascade with RBF networks to provide a double local network architecture resulting in even quicker learning and faster convergence [3].

At present, there are two different kinds of WNN structure, one with fixed wavelet bases, where the dilation and translation parameters of wavelet basis are fixed, and only the output layer weights are adjustable. For the WNN with fixed wavelets, the main problem is the selection of wavelet bases/frames. The wavelet basis has to be selected appropriately since the choice of the wavelet basis can be critical to approximation performance. Another type is the variable wavelet bases, where the dilation parameters, translation parameters and the output layer weights are adjustable. Stability of the WNN structures as identification/controller schemes (structural WNN's adaptation, parameter adjusting and control of nonlinear systems) has been carried out using the Lyapunov theory [16].

In terms of engineering applications, WNNs have also shown promising results in both signal representation and classification [4-5]. The main objective of this work is to investigate the applicability of an Adaptive Neuro-Wavelet Network (ANW) architecture that incorporates an impulse infinite-response (IIR) filter for modelling and control of non-linear processes. Traditional self-tuning adaptive control approaches are limited in that they cannot deal with complex nonlinear systems. Typically, these techniques assume that the control model is operating in a linear region. The parameters of a linearised plant model are estimated recursively and used to update the controllers. Generally, it is not possible to design a controller based on mathematical analysis for such plants that consist of the nonlinearity and the uncertainty. The problem is exacerbated when the functions describing the plants are unknown and change with time [6]. It is important to develop an effective technique in which the structure of the unknown, linear/nonlinear plant models can be identified as an adaptive process; and controllers have to be designed which act rapidly, accurately and in a stable fashion. Developments in neuro-control design [7] have proved to be useful for a wide class of practical situations showing that they can cope with significant unknown nonlinearity. The idea of neuro-control is to first process an identification model that approximates the unknown dynamics of the plant in which the parameters of the neural network are adjusted off-line. In a recent research study a PI control strategy using WNN schemes combined with IIR filters has been proposed for the identification/control of a wind

turbine used in a wind energy conversion system [17]. The authors utilised adaptive RASP (Rational Functions with Second-Order) wavelet functions. These functions although are characterized from a fast training time when they are incorporated in a WNN structure. However oscillations and instability has been reported when the used number of RASP functions is exceed a specific number.

The goal of this research study is to implement a control method that addresses the self-tuning PI control problems for severely nonlinear systems by utilising ANWs to achieve a nonlinear controller design. In contrast to the RASP function, in this study, the usage of Morlet functions is proposed. The Morlet wavelet function is directional (in the sense of being effective in selecting orientations) and capable of fine tuning specific frequencies. These latter capabilities are especially important in filtering out the background noise, and comprise the advantages of the Morlet wavelet with respect to other filters such as RASP and Shannon. In fact, having exponential decay in both the time and frequency domain, the Morlet function has optimal joint time-frequency concentration.

Simulation results to a pH CSTR demonstrate the above concept. In this proposed method, the ANW scheme is needed to learn the characteristics of the plant dynamic systems and make use of it to determine the future inputs that will minimise error performance index so as to compensate the PI controller parameters.

2. CONTINUOUS STIRRED TANK REACTOR (CSTR) FOR pH CONTROL

We investigate a benchmark problem for nonlinear control system design, which is based on a specific continuous stirred tank reactor (CSTR). Let consider a pH CSTR schematically shown in Fig. 1.

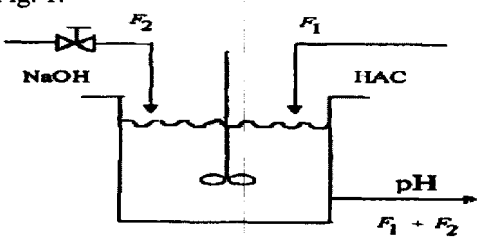

Fig. 1: pH CSTR

The CSTR has two input streams, one containing sodium hydroxide and the other acetic acid. A dynamic model for the pH in the tank can be obtained using the approach presented by Jean Saint-Donat [8]. Fig. 2 shows that the process has highly non-linear steady-state behaviour. It is a weak acid/strong base system which exhibits large gain changes. It can be seen that pH CSTR is highly nonlinear around pH=7. In fact, the steady state pH gain to basic flow changes by a factor of 2.8.

The model is derived from first principles, material balances and chemical equilibria, and has become generally accepted in the literature. The method implements mass balances on components, or combinations of components, called reaction invariants by Gustafsson and Waller [9], of the CSTR solution's ionic species. These reaction invariants are suitable for mass balancing because, unlike the concentration of the hydrogen ion, they do not change as the reaction equilibrium shifts. The equilibrium relations for the weak acid/strong base system are:

$$\frac{[AC^-][H^+]}{HAC} = K_a \quad (1)$$

$$\frac{[Na^+][OH^-]}{[NaOH]} = K_b \quad (2)$$

$$[H^+][OH^-] = K_w \quad (3)$$

where, K_a is the dissociation constant of HAC, K_b is the dissociation constant of $NaOH$ and K_w is the dissociation constant of water. $NaOH$ is a strong base which fully dissociates (i.e. $1/K_b = 0$) and hence Eq. 2 yields $NaOH = 0$.

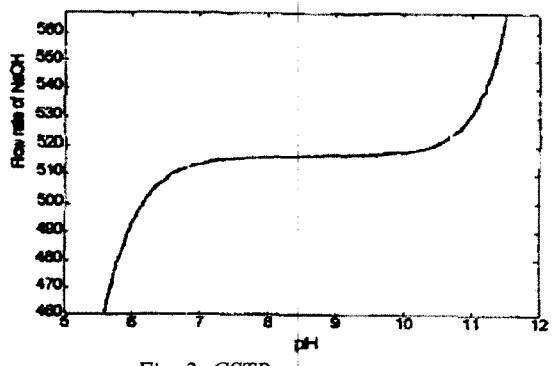

Fig. 2: CSTR process curve

Two reaction invariants for the reactions of the system are the total ionic concentration of the acid and the total ionic concentration of the base:

$$\xi = [HAC] + [AC^-] \quad (4)$$

$$\zeta = [Na]^+ \quad (5)$$

In addition to the chemical equilibria for the acid and base, the solution must remain electrically neutral at all times, giving

$$\zeta + [H^+] = [OH^-] + [AC^-] \quad (6)$$

The reaction invariants, ξ and ζ are found from the following mass balances

Total acetate balance:

$$F_1C_1 - (F_1 + F_2)\xi = V\frac{\partial \xi}{\partial t} \quad (7)$$

Sodium ion balance:

$$F_2C_2 - (F_1 + F_2)\zeta = V\frac{\partial \zeta}{\partial t} \quad (8)$$

where F_1 is the acid flow rate, F_2 is the base flow rate, C_1 is the acid concentration, C_2 is the base concentration and V is the CSTR liquid volume. Finally the pH is calculated using

$$pH = -Log_{10}([H^+]) \quad (9)$$

3. WAVELET NEURAL NETWORKS AS FUNCTION APPROXIMATORS

Pati and Krishnaprasad [10] described a network in which the sigmoidal activation functions of a typical neural network are replaced by particular shifts and dilations of a given mother wavelet. Thus, consider Eq.1 where **T**, a closed proper subset of $\mathbf{R}\times\mathbf{R}$, is the set of all training pairs (x,y):

$$y \approx f(x) = \sum_{m,n} w_{m.n}\psi_{m.n}(x), \quad \forall (x,y) \in \mathbf{T}, w_{m|n}, x, y \in \mathbf{R}, m \in \mathbf{Z}, n \in \mathbf{Z}^{+} \tag{10}$$

where "≈" is defined such that there exists $\varepsilon \in \mathbf{R}^{+}$ so that

$$\varepsilon > |f(x) - y|^2 \tag{11}$$

and where $\psi_{m,n}$ is a wavelet such that

$$\psi_{mn}(x) = 2^{-m/2}\psi(2^{-m}x - n) \tag{12}$$

Pati's network is similar to the general expression of the discrete wavelet transform. In this case, a network structure is being considered, that is simply a projection onto a basis – an inner product – where the basis is a wavelet basis. The training vectors are thus just been projecting onto the wavelet basis. Since an infinite basis cannot be implemented, a finite subset over the compactly supported interval on which the training data is defined is chosen. Furthermore, the set is also limited to a maximum dilation. Define **I** as the finite set of all shifts and dilations *(m,n)*. Then the training data can now been approximated by the finite set of shifts and dilations $(m,n) \in \mathbf{I}$ and a corresponding set of coefficients (or weights) $\{w_{m,n}\}_{(m,n)\in\mathbf{I}} \subset \mathbf{R}$ [10]. The overall approximation error is determined by

$$E = \sum_{(x,y)\in\mathbf{T}} |f(x) - y|^2 \tag{13}$$

This error functional is nearly identical to that of the Back-propagation (BP) algorithm with only one important difference. It turns out that the error functional described above is convex in terms of the weights $w_{m,n}$. This is quite different from the BP algorithm, which, in general, has a non-linear error surface. Due to the convexity of the error functional, any minimiser is a global minimiser. Furthermore, it is clear that simple iterative schemes such as gradient descent perform adequately since there is no possibility of getting stuck in local minima. The authors have presented a network synthesis algorithm. The algorithm involves determining the set of wavelets for use as activation functions for the hidden layer neurons by considering the time and frequency limits of the training data. Given that the training data is bounded in time and frequency, the exact shifts and dilations of the mother wavelet can be determined which are necessary to adequately cover the time and frequency range of the training data. This number is the upper bound of hidden layer neurons necessary to approximate the functional relationship between x and y to any precision ε. However, this method can be

computationally intractable if the number of required wavelets is very large; *i.e.,* the time and frequency bounds are very large. In an alternative approach, Zhang [11] describes an implementation of a wavelet neural network based on Pati and Krishnaprasad's [10] synthesis algorithm and the orthonormal least squares minimisation method. Zhang proposes to build a candidate set of wavelets based from the initial infinite set of all possible shifts and dilations of the mother wavelet by first truncating it to a finite set based on some *a priori* knowledge about the training data. His method involves using the Gram-Schmidt ortho-normalisation method to determine the N wavelets and their shift and dilation parameters. Finally, the weights are calculated by a simple inversion of an upper triangular matrix. The criteria are given by the time and frequency support of the training data set. The resulting set is a subset of the regular pyramid structure of wavelets usually associated with dyadic multiresolution decomposition. The goal is to select N wavelets from the candidate set, such that these N are optimal with respect to approximation error [11].

4. ADAPTIVE NEURAL WAVELET NETWORK

In contrast to the networks proposed by Pati and Krishnaprasad and Zhang, an alternative approach is proposed that does not fix the shift and dilation parameters [12]. The "Adaptive Neural-Wavelet network" (ANW) concept could be considered as a capable for approximating arbitrary nonlinear functions. Fig. 3 illustrates the schematic of the proposed network. In this proposed architecture, the approximation model consists of an adaptive neural-network topology with the wavelet transformation embedded in the hidden units.

Such architecture approximates any desired signal $z(t)$ by generalising a linear combination if a set of daughter wavelets $h_{a,b}(t)$, where $h_{a,b}(t)$ are generated by dilation a and translation b from a mother wavelet $h(t)$:

$$h_{a,b}(t) = h\left(\frac{t-b}{a}\right) \tag{14}$$

with the dilation factor $a > 0$. Given a list of K candidate wavelets obtained from the discrete wavelet decomposition, the approximated signal of the network $\hat{y}(t)$ can be represented by:

$$\hat{z}(t) = u(t)\sum_{k=1}^{K} w_k h_{a_k,b_k}(t) \tag{15}$$

where w_k are the weight coefficients towards network's output.

The inversion formula of the wavelet transform cannot be expressed directly by finite neural networks, but can be approximated using neural network topology with finite hidden units. This is so because most targets are restricted in both the time and frequency domain. The training

algorithm consists of two processes: the initialisation phase and the parameters' update through the minimisation of error. The network consists of one hidden layer with an appropriate number of nodes, which are defined by the user.

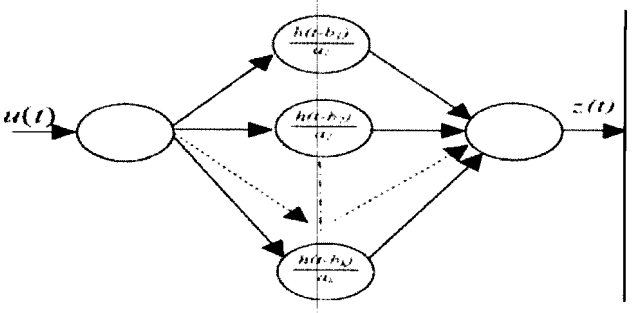

Fig. 3: Schematic of ANW network

It is assumed that the network output function satisfies the admissibility condition and that the network sufficiently distributes K sets of the mother wavelet basis function defined by the user, evenly partitioning the region of interest. The wavelet used in this study is the Morlet function.

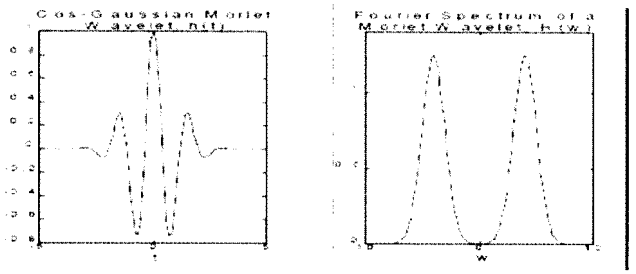

Fig. 4: Cos-Gaussian Morlet Wavelet $h(\tau)$ and its Fourier Transform $H(\omega)$

The Morlet's basic wavelet function is a multiplication of Fourier basis with a Gaussian window [13]:

$$h(\tau) = \exp(j\omega_0\tau)\exp(-0.5\tau^2) \tag{16}$$

Its real part is a Cos-Gaussian and the imaginary part is a Sin-Gaussian function. The Cos-Gaussian wavelet is a real even function. The Fourier transform of the Cos-Gaussian wavelet is the Gaussian functions shifted to ω_0 and $-\omega_0$ respectively:

$$H(\omega) = \sqrt{\frac{\pi}{2}}(\exp[-0.5(\omega-\omega_0)^2] + \exp[-0.5(\omega+\omega_0)^2]) \tag{17}$$

Which is even and real positive valued. Fig. 4 illustrates the plots of the Morlet wavelet and its Fourier spectrum, with $\omega_0 = 4$. The Morlet wavelets do not satisfy the wavelet admissible condition, because

$$H(0) = \sqrt{2\pi}\exp(-0.5\omega_0^2) \neq 0 \tag{18}$$

That leads to $c_h = +\infty$. However, if ω_0 is sufficiently large, say $\omega_o = 4$, then $H(0)$ comes very close to zero and can be practically considered as zero in numerical computations. The neural-network parameters w_k, a_k, b_k an be optimised by minimising the mean square error (E) function over all samples. Thus, $e(t) = z_n(t) - \hat{z}_n(t)$ is a time varying error function at time t, where $y_n(t)$ is the desired (target) response. The (E) function is defined by

$$E = 1/2\sum_{t=1}^{T}(z(t) - \hat{z}(t))^2 \tag{19}$$

We choose to minimise E using the gradient descent minimisation algorithm for the variables a, and b. Therefore we must find the partial derivatives of Eq. 19 with respect to w_k, a_k, b_k.

$$\begin{aligned}
\frac{\partial E}{\partial w_k} &= -\sum_{t=1}^{T}(z(t) - \hat{z}(t))h(\frac{t-b_k}{a_k})u(t) \\
\frac{\partial E}{\partial b_k} &= \sum_{t=1}^{T}(z(t) - \hat{z}(t))u(t)w_k h'(\frac{t-b_k}{a_k})(\frac{1}{a_k}) \\
\frac{\partial E}{\partial a_k} &= \sum_{t=1}^{T}(z(t) - \hat{z}(t))u(t)w_k h'(\frac{t-b_k}{a_k})(\frac{t-b_k}{a_k^2}) = \\
&\sum_{t=1}^{T}(z(t) - \hat{z}(t))u(t)w_k h'(\frac{t-b_k}{a_k})(\frac{t-b_k}{a_k})\left(\frac{1}{a_k}\right) = (\frac{t-b_k}{a_k})\frac{\partial E}{\partial b_k}
\end{aligned} \tag{20}$$

The resulting update for all parameters to be tuned is as follows:

$$\begin{aligned}
a_k^{new} &= a_k^{old} - \eta_a \frac{\partial E}{\partial a_k}, \\
b_k^{new} &= b_k^{old} - \eta_b \frac{\partial E}{\partial b_k}, \\
w_k^{new} &= w_k^{old} - \eta_w \frac{\partial E}{\partial w_k},
\end{aligned} \tag{21}$$

where $\eta_{a,b,w} \in \Re$ is the step-size parameter of the gradient descent update.

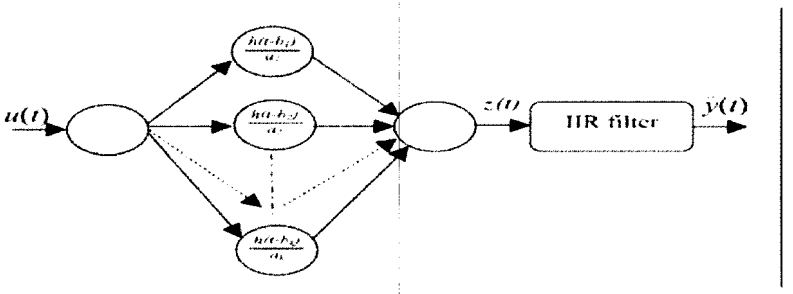

Fig. 5: Schematic of ANW-IIR network

In cascade with this network we have deployed a local infinite impulse response (IIR) block structure. The IIR structure is then used to create a ''double'' local network architecture that provides a computationally efficient method of training the system and results accordingly in quick learning, and fast convergence, subjected to the number of K wavelet functions employed in the ANW scheme [18].

Fig. 5 illustrates the structure which approximating any desired signal $y(t)$ by generalising a linear combination of a set of daughter wavelets $h(t)$ cascaded with the local IIR recurrent networks. The approximated signal of the network $\hat{y}t)$ can be modelled by:

$$\hat{y}(t) = \sum_{i=0}^{M} c_i z(t-i)u(t) + \sum_{j=1}^{N} d_j \hat{y}(t-j)v(t) \tag{22}$$

where M and c_i are the number of feed-forward delays and coefficients of the IIR filter, respectively, N and d_j are the number of feedback delays and recursive filter coefficients, respectively. The IIR structure as shown through Eq. 22 is illustrated in Fig. 6.

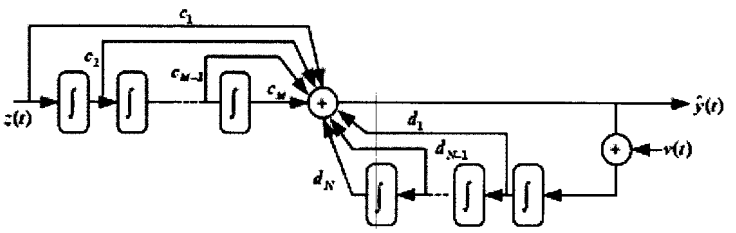

Fig. 6: Schematic of the IIR scheme

The signals $u(t)$ and $v(t)$ are the input and co-input to the system at time t, respectively. Input $v(t)$ is usually kept small for feedback stability purposes. In this extended structure, the neural-network parameters w_k, a_k, b_k, c_i, d_j can be optimised by minimising the mean square error (E) function over all samples using again the gradient descent learning algorithm. The related gradients that are required to update these parameters are given as:

$$\frac{\partial E}{\partial w_k} = -\sum_{t=1}^{T} u(t)e(t) \sum_{i=0}^{M} c_i h\left(\frac{t-b_k}{a_k} - i\right) \tag{23}$$

$$\frac{\partial E}{\partial b_k} = \sum_{t=1}^{T} u(t)e(t) \sum_{i=0}^{M} c_i w_k h'\left(\frac{t-b_k}{a_k} - i\right) \tag{24}$$

$$\frac{\partial E}{\partial a_k} = \sum_{t=1}^{T} u(t)e(t) \sum_{i=0}^{M} c_i \left(\frac{t-b_k}{a_k}\right) w_k h'\left(\frac{t-b_k}{a_k} - i\right) = \left(\frac{t-b_k}{a_k}\right) \frac{\partial E}{\partial b_k} \tag{25}$$

$$\frac{\partial E}{\partial c_i} = -\sum_{t=1}^{T} u(t)e(t)z(t-i) \tag{26}$$

$$\frac{\partial E}{\partial d_j} = -\sum_{t=1}^{T} v(t)e(t)\hat{y}(t-j) \tag{27}$$

The resulting update for the remaining parameters to be tuned is as follows:

$$c_i^{new} = c_i^{old} - \eta_c \frac{\partial E}{\partial c_i}, \tag{28}$$

$$d_j^{new} = d_j^{old} - \eta_d \frac{\partial E}{\partial d_j}, \tag{29}$$

with $\eta_{c,d} \in \Re$ is the step-size parameter of the gradient descent update.

ANW scheme is similar to the structure of a RBF network, in the sense that wavelets are local basis functions that provide less interfering than global ones, leading to a noncomplex dependency in the neural network parameters. However, during the initialisation phase, care should be taken for determining the initial parameters values. All initial weights w_k are initialised to small values between ±0.2 (similarly to the initialisation of MLP networks) while the dilation parameters a_k are initialised using the heuristic rule "*global first nearest-neighbour*" [19]. It uses the uniform average width for all units using the Euclidean distance in the input space between each unit *m* and its nearest neighbour *n*. Note that if the dilation parameters are set too wide, they can cause several overlapping partitions and thus cannot be realised. Setting a_k too narrow may result in longer convergence. Initial translation parameters b_k are spaced equally apart throughout the training data to provide non-overlapping partitions throughout the neighbouring intervals. Finally, the initial IIR coefficients *c* and *d* should be set so that the system has poles inside the unit circle, thus both are set to 0.1.

A common problem of IIR adaptive networks is the problem of guaranteeing stability and convergence. In particular, IIR adaptive networks are prone to instability as a consequence of unbounded growth of the adaptation coefficients. Furthermore, recursively adapting coefficients creates movement in the location of poles from the origin, causing the network to become unstable, even if the adaptation is stable. Finally, the convergence of the steepest descent gradient algorithm that is applied to minimise the error sometimes becomes stuck in a local minimum. The problem of the potential adaptive instability can be solved by successively reducing the learning rate factor. In addition, as it is shown in Fig.6, a "gamma operator", which imposes both trivial stability conditions and is computationally effective has been utilised [20]. The "gamma operator" has the following form:

$$v(t) = \frac{\gamma}{t-(1-\gamma)} \tag{30}$$

where γ is a real parameter that controls the memory depth of this operator and usually is in the range $0 < \gamma < 2$.

5. HYBRID-PI BASED CONTROLLER

Traditional self-tuning adaptive control approaches are limited in that they cannot deal with complex nonlinear systems. Typically, these techniques assume that the control model is operating in a linear region. The parameters of a *linearised* plant model are estimated recursively and used to update the controllers. Generally, it is not possible to design a controller based on mathematical analysis for such plants that consist of the nonlinearity and the uncertainty. The problem is exacerbated when the functions describing the plants are unknown and change with time. Such nonlinear time-varying adaptive control problems are arising with increasing frequency in today's technology. It is important to develop an effective technique in which the structure of the unknown, linear/nonlinear plant models can be identified as an adaptive process; and controllers have to be designed which act rapidly, accurately and in a stable fashion.

The proposed scheme in this research study is the usage of adaptive self-tuning PI controllers using the ANW network. In this methodology, the ANW scheme is needed to learn the characteristics of the plant dynamic systems and make use of it to determine the future inputs that will minimise error performance index so as to compensate the PI controller parameters.

The PI controller is one of the simplest of the traditional feedback controller schemes. Nevertheless, the linear PI algorithm might be difficult to deal with processes with complex dynamics such as those with large dead time, inverse response and highly nonlinear characteristics. To improve the control performance, an adaptive PI algorithm is proposed by utilising the simple PI controller structure based on self-tuning schemes of the ANW parameters. The basic idea of PI control is that the control action $u(k)$ should be proportional to the error and the integral of the error over time. However, limited performance can be of disadvantages to the linear PI controller i.e., the PI mode is used to eliminate the steady-state offset, which sometimes can cause excessive overshoot due to direct implementation of the integral action, *etc.* The proposed adaptive variable PI controller can help to improve the limited performance of the static PI controller dealing with conflict in nature between static accuracy (steady-state error) and dynamic responsiveness (speed of response). Several tuning components determine the contribution of the weights of the error that suits a cost function $E = \frac{1}{2}\sum_{k=1}^{T}(r(k) - \hat{y}(k))^2$, where $r(k)$ is the desired set-point and $\hat{y}(k)$ is the ANW output. Before beginning tracking operation using the ANW based PI controller, the unknown nonlinear CSTR must be identified according to a certain model. In this particular identification process, the model consists of a "*neural*" network topology with the wavelet transform embedded in the hidden units. In cascaded with the network is a local infinite impulse response (IIR) block structure as shown in Fig. 5.

Let us consider a general SISO dynamical system represented in discrete domain by the state equations

$$\begin{aligned} x(k+1) &= f\big(x(k),u(k),k\big) \\ y(k) &= g\big(x(k),k\big) \end{aligned} \tag{31}$$

where $x(k) \in \Re^n$ and $u(k), y(k) \in \Re$. The only accessible data are the input u and output y.

It has been shown [14] that if the linearised system around the equilibrium state is observable, an input-output representation exists which has the form

$$y(k+1) = \varphi\big(y(k), y(k-1),..., y(k-n+1), u(k), u(k-1),..., u(k-n+1)\big) \tag{32}$$

i.e. a function $\varphi(.)$ exists that maps $y(k)$ and $u(k)$,and their $n-1$ past values into $y(k+1)$. In view of this, a learning-based model $\hat{\varphi}$ can be trained to approximate φ over the domain interest. The considerations are based on the neural network controller design of the control system. The following alternative model of an unknown plant that can simplify the computation of the control input is described by the equation

$$y(k+1) = \varphi\big(y(k)\big) + \Gamma\big(y(k)\big)u(k) \tag{33}$$

for a discrete-time process of dimension 1, where $y(k)$ and $u(k)$ denote the input and the output at the k^{th} instant of time.

If the nonlinearity terms $\varphi(.)$ and $\Gamma(.)$ are known exactly, the required control $u(k)$ for tracking a desired output $r(k+1)$ can be computed at every time instant using the formula

$$u(k) = \frac{r(k+1) - \varphi(y(k))}{\Gamma(y(k))} \tag{34}$$

However, if $\varphi(.)$ and $\Gamma(.)$ are unknown, the idea is to use the proposed ANW scheme to approximate the system dynamics i.e.,

$$\hat{y}(k+1) = \hat{\varphi}\big(y(k), \Theta_\varphi\big) + \hat{\Gamma}\big(y(k), \Theta_\Gamma\big)u(k) \tag{35}$$

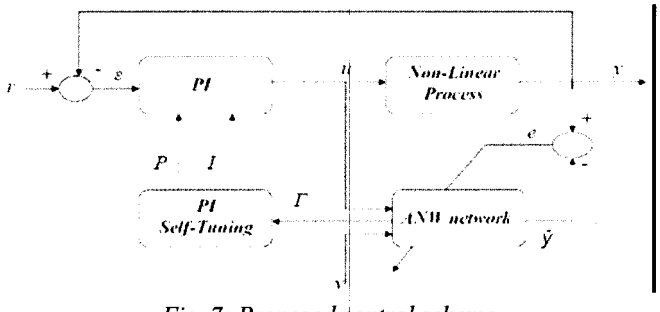

Fig. 7: Proposed control scheme

Comparing the model of Eq. (35) with the one of Eq. (22) we can conclude that

$$\hat{\varphi}\big(y(k), \Theta_\varphi\big) = \sum_{j=1}^{N} d_j \hat{y}(k-j) v(k) \tag{36}$$

$$\hat{\Gamma}\left(y(k),\Theta_{\Gamma}\right)=\sum_{i=0}^{M}c_i z(k-i) \tag{37}$$

After the nonlinearities $\varphi(.)$ and $\Gamma(.)$ are approximated by the two distinct network functions $\hat{\varphi}(.)$ and $\hat{\Gamma}(.)$ with adjustable parameters, represented by Θ_{φ} and Θ_{Γ} respectively, the PI control $u(k)$ for tracking a desired output $r(k+1)$ can be obtained from

$$u(k)=u(k-1)+P[\varepsilon(k)-\varepsilon(k-1)]+I\varepsilon(k) \tag{38}$$

where P and I are proportional and integral gains, $u(k)$ is a plant input at kT, where T is a sampling interval, and

$$\varepsilon(k)=r(k)-y(k) \tag{39}$$

P and I parameters are considered as part of the function of E and can be optimised and updated according to the cost function E,

$$P(k)=P(k-1)+\mu_P e(k)\Gamma(k)\left(\varepsilon(k)-\varepsilon(k-1)\right) \tag{40}$$

$$I(k)=I(k-1)+\mu_I e(k)\Gamma(k)\varepsilon(k) \tag{41}$$

where $\hat{\Gamma}$ comes from Eq. (37), and μ is the fixed learning rate of each adaptive PI parameter. Fig. 7 illustrates the diagram of the resulting network topology based on the PI controller for self-tuning control of CSTR. Stability of the closed loop, where the parameters of a linear controller are been tuned via a learning model of the nonlinear plant/process, has been already been addressed in [21], where the proof of tracking error asymptotic stability is provided.

6. SIMULATION RESULTS

The first step in the ANW model design procedure is the generation of training data set and this requires the design of the process input signal. The design of the input signal for nonlinear system identification is more complicated than it is for linear system identification. The input signal should excite all the frequencies of interest and should also excite the process over the whole of the required operating region [14].

A Random Amplitude Signal (RAS) is commonly used as the process excitation signal to generate open loop data for neural network training. This signal consists of a uniformly distributed random variable applied to the process input at each clock period and is more likely to exercise the process over the desired operating range than a binary signal [15], such as a pseudo-random binary sequence (PRBS) which is widely employed for linear system identification. A Random Amplitude Signal is specified by its clock period, which should be a multiple of the sample time so that the process input is constant between consecutive samples, and by its amplitude range, which may be expressed as a percentage maximum deviation from a steady-state value.

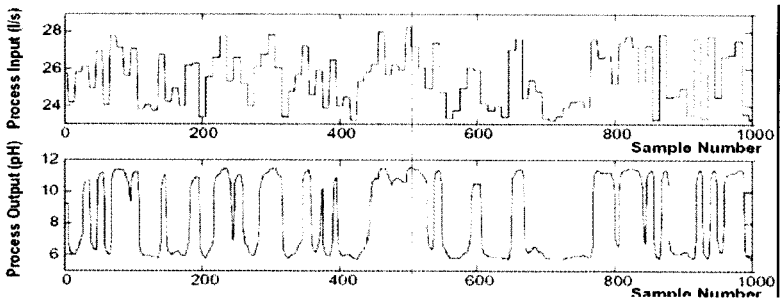

Fig. 8: Process output using RAS

The aim of CSTR process excitation is to generate I/O process data which contains sufficient information for a neural network to identify the non-linear process dynamics over the entire operating range, and a RAS is commonly employed to achieve this. However, there is no reason why this signal should achieve adequate excitation of the non-linear dynamics of all processes. This is particularly so for this CSTR pH process where the strong non-linearity is characterised by its steady-state titration curve (Fig. 2). One of the obstacles to accurately modelling such a process is to obtain output data in the high gain region for network training. When the pH process is excited by a standard RAS, little output data is generated in the area where the process gain is a maximum. The data distribution of Fig. 8 illustrates the lack of output data between pH 7 and 10, and any learning-based network model which is trained with such data could have large prediction errors in this region.

Such training data base was developed by forcing the stream of sodium hydroxide F_2 with a Random Amplitude Signal superimposed on its steady-state value. The parameters for the CSTR considered can be seen in [8]. It is seen in Fig. 8 the changes in consecutive process inputs generated by the RAS are often small and this can result in the process output remaining in a low gain region for several consecutive RAS clock pulses. This is disadvantageous because it promotes an uneven output data distribution. One practical way of improving the uneven distribution of the training data is to force the signal through the region of maximum process gain on each clock pulse.

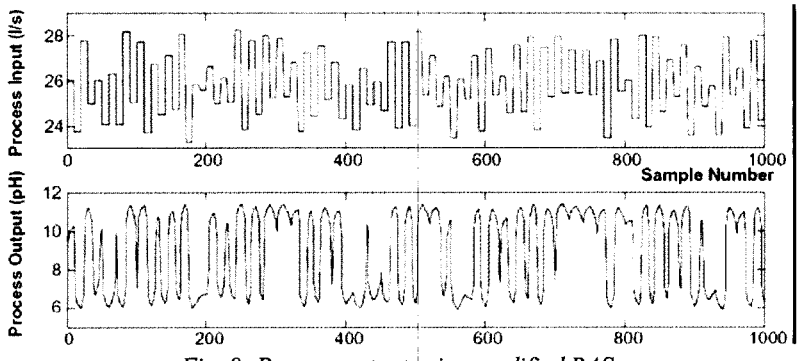

Fig. 9: Process output using modified RAS

The resulting "modified" RAS, illustrated in Fig. 9, has a uniformly distributed input in the two intervals above and below a threshold level which was chosen as the process input at maximum steady-state gain. While the data distribution still appears to be very uneven, there is a threefold increase in the data density between pH 7 and 10 when compared to the output data density generated by a standard RAS.

Using the data extracted from a RAS, the ANW scheme with Morlet mother wavelets is employed to approximate the pH data. IIR block structure with feed forward coefficients M=3 and feedback coefficients N=3 is also implemented. Note that if the dilation parameters are set too wide, they can cause several overlapping partitions and thus cannot be rallied. Fig. 10 illustrates the approximation of the CSTR performance using RAS input signals. The ANW approximates satisfactory the process curve.

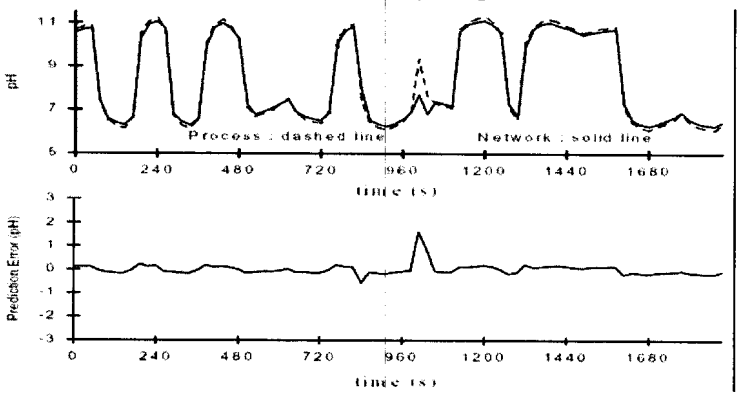

Fig. 10: Performance of modelling the pH Process

In this training process, 36 Morlet wavelets have been employed in the ANW scheme and the error goal of 0.032 was achieved at 250 iterations/epochs. In general, more iterations/epochs as well as larger training dataset are needed for a more accurate performance. The learning rate parameters for weights, dilations, translations, IIR feedforward coefficients, and feedback coefficients were fixed at 0.01, 0.05, 0.05, 0.02, and 0.02, respectively.

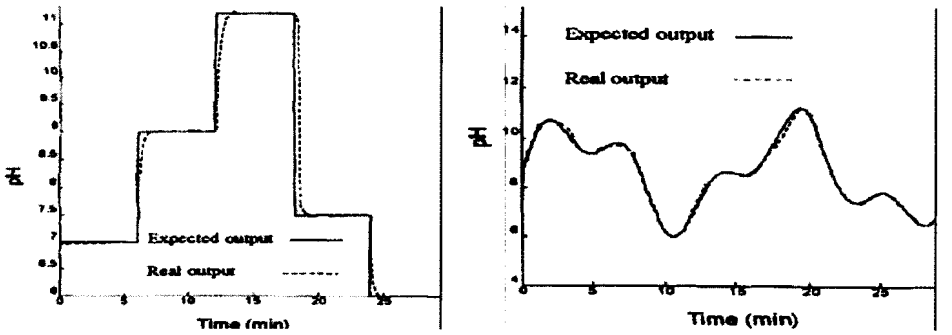

Fig. 11: Performance of proposed control strategy

All initial weights w_k were initialised to ±0.1 and the dilations a_k were set initially to 7.5. Just for comparison, an ANW scheme without the feed-forward and feedback coefficients managed to achieve the same error goal however with an increased number of epochs. Future work will be focused to address the self-structure of the ANW scheme during the training phase, i.e. the number of wavelet functions to be determined during the learning process.

After the identification model is completed, the tracking operation takes command of the neuro-process control to track the desired set-point. On-line control results are shown in Fig. 11. Fig. 11a shows the control result when the system traces step responses, while Fig. 11b shows that when the system traces a smooth curve. The proposed self-tuning neuro-wavelet controller utilised the ANW-IIR scheme. Controller parameters P and I were initially set at 0.2 and 0.1 respectively which then later vary with local control network conditions. The emphasis is on the plant output responses to the reference set point.

7. CONCLUSIONS

This paper discussed the application of neuro-wavelet networks in the implementation of adaptive controllers for the identification and control of nonlinear processes. The approach used, based on a single layer feed forward neural networks with hidden nodes of adaptive Morlet wavelet functions, PI controller and an infinite impulse response (IIR) recurrent structure, allowed fast convergence to a simple nonlinear dynamic behaviour. In this research study, controller ability for set point tracking was demonstrated on a pH CSTR process.

REFERENCES

1. Haykin S. (1999). Neural networks: A comprehensive foundation, Prentice Hall.
2. Zhang J., Gilbert, G. (1995). Wavelet Neural Networks for Function Learning. IEEE Trans. Signal Processing, Vol. 2, No. 6, pp. 1485-1496.
3. Ye X., Loh N.K. (1993). Dynamic System Identification Using Recurrent Radial Basis Function Network. Proceedings of American Control Conference, Vol. 3, pp. 2912-2916.
4. Oussar, Y., Rivals, I., Personnaz, L., Dreyfus, G. (1998). Training wavelet networks for nonlinear dynamic input-output modelling. Neurocomputing, Vol. 20, pp. 173-188.
5. Wai R-J, Chang J-M (2002). Intelligent control of induction servo motor drive via wavelet neural network. Electric Power Systems Research, Vol. 61, pp. 67–76.
6. Fossen T.I., Sagatun S.I., (1991). Adaptive control of nonlinear systems: A case study of underwater robotic systems. Journal of Robotic Systems, Vol. 8, No. 3, pp. 392-412.
7. Hrycej T., (1997). Neurocontrol: towards an industrial control methodology, John Wiley.

8. Saint-Donat J., Bhat N., McAvoy T.J. (1997). Neural net based model predictive control. Int. J. Control, Vol. 6 pp. 1453-1468.
9. Gustaffson T.K., Waller K.V. (1983). Dynamic modelling and reaction invariant control of pH. Chem. Eng. Sci., Vol. 38, pp.389-398.
10. Pati Y.C., Krishnaprasad P.S. (1993). Analysis and synthesis of feed-forward neural networks using discrete affine wavelet transformations. IEEE Transactions on Neural Networks, Vol. 4, pp. 73–85.
11. Zhang Q. (1993). Regressor selection and wavelet network construction. Proceedings of the 32nd Conference on Decision and Control, pp. 3688–3693.
12. Kodogiannis V., Wadge E., Boulougoura M., Christou K. (2004). Feature extraction techniques using adaptive wavelets for signal classification. WSEAS Trans. on Circuits and Systems, Vol. 3, No. 1, pp. 149-154.
13. Daubechies I., (1992). Ten Lectures on Wavelets. Philadelphia: Society for Industrial and Applied Mathematics.
14. Nelles O. (2000). Nonlinear system identification. Springer – Verlag.
15. Prasad G., Swidenbank E., Hogg B.W. (1999). Neural network model-based multivariable predictive control algorithms with application in thermal power plant control. Control and Intelligent Systems, Vol. 27, No. 3, pp. 108-131.
16. C. de Sousa, Jr., E. Moreira Hemerly, R. Kawakami, H. Galvão, (2002). Adaptive Control for Mobile Robot Using Wavelet Networks. IEEE trans. on systems, man, and cybernetics - part B, vol. 32, no. 4, pp. 493-504.
17. Sedighizadeh, M., Kalantar, M., Esfandeh, S., Arzaghi-Harris, D. (2005). Nonlinear model identification and control of wind turbine using wavenets. Proc. of IEEE conf. on Control Applications (CCA 2005), pp. 1057-1062.
18. Kasabov N. (1994). Foundations of neural network fuzzy systems and knowledge engineering, MIT Press.
19. Moody, J., Darken, C.J., (1989). Fast learning in networks of locally tuned processing units. Neural Computation, Vol. 1, pp. 281-294.
20. Principe J., De Vries B., De Oliveira p. G. (1993). The gamma filter: a new class of adaptive IIR filters with restricted feedback. IEEE Trans. on signal processing, Vol. 41, No. 2 pp. 649-656.
21. Sanchez, E.N., Bernal, M.A, (2000). Adaptive recurrent neural control for nonlinear system tracking, IEEE Trans on SMC, part B, Vol. 30, No. 6, pp. 886-889.