Controlling the Multi-Scroll Chaotic Attractors Using a Fuzzy Neural Networks Compensator

Da Lin,* Xingyuan Wang,† and Lulu Wang

School of Electronic and Information Engineering,
Dalian University of Technology, Dalian 116024, China
(Received February 25, 2009)

In this paper controlling multi-scroll chaotic attractors from a saturated function series is studied. For many chaotic systems that can be decomposed into a sum of a linear and nonlinear part, under some mild conditions fuzzy neural networks (FNNs) can be used to well approximate the nonlinear part of the system dynamics. The resulting system is then dominated by the linear part, with some small or weak residual nonlinearity due to the FNN approximation errors. Thus, a simple linear state feedback controller can be proposed to drive the multi-scroll chaotic attractors to the desired targets or periodic trajectory. In addition to some theoretical analysis, computer simulations on multi-scroll chaotic attractors are presented to demonstrate the effectiveness of the proposed control scheme.

PACS numbers: 05.45.Gg

I. INTRODUCTION

It is well known that chaos is useful and has great potential in many real-world engineering fields, such as digital data encryption and secure communications, biomedical engineering, flow dynamics and liquid mixing, power-system protection, and so on [1, 2].

Recently, extended from Chua’s circuit, the theoretical design and hardware implementation of different kinds of chaotic oscillators have attracted increasing attention, targeting real-world applications of many chaos-based technologies and information systems. This has stimulated the current research interest in creating various complex multi-scroll chaotic attractors by using simple electronic circuits and devices. Since Suykens and Vandewalle [3–5] first introduced a family of \( n \)-double scroll chaotic attractors from the so-called quasi-linear function approach, many different approaches or techniques have been proposed to generate multi-scroll chaotic attractors, such as the hysteresis series switching method and the thresholding approach among many others [6–8].

Chaos control is an important topic in the nonlinear science. In essence, chaos control is guiding a chaotic system to reach a desired goal dynamics via various controllers. Chaos control approaches can be classified into two categories, namely feedback and non-feedback methods [9]. The feedback methods stabilize one of the unstable periodic orbits embedded in its chaotic attractor by applying a small time-dependent perturbation, which is proportional to the deviation between the desired and actual trajectories calculated at every time instant. Such examples as the OGY method [10], the occasional proportional feedback technique...
[11], the delayed feedback method [12], linear feed-back control [13], active control [14], and adaptive control [15], are all typical feedback methods. For the systems in which the state variables are not accessible on-line or for high-speed systems, it is difficult to apply the feedback scheme, and non-feedback methods are more appropriate [16-29].

At present, most of the previous chaos control studies focused on the general topologically simple chaotic attractors. However, multi-scroll chaotic attractors have many practical applications in, for example, broadband signal generation, cellular neural networks (CNNs), secure and digital communications, and perhaps efficient liquid mixing, to name a few from among others. Suykens and Vandewalle [30] used the 5-scroll chaotic attractor in their KU Leuven time-series prediction competition. To increase the complexity of the chaotic dynamics for higher security [31], multi-scroll chaotic attractors could be used for communication instead of the general topologically simple chaotic attractors. Yalcin et al. [32] experimentally verified a nonlinear $H_\infty$ synchronization scheme with 5-scroll chaotic attractors. Tang et al. [33] proposed a secure digital communication system that can resist intrusion of eavesdroppers during the transmission of signals and data, for which multi-scroll attractors may have better performance. More recently, Yalcin et al. [34-36] also designed a novel “true random bit generator” based on a double-scroll chaotic attractor. Yalcin et al. [37] introduced cellular neural networks, multi-scroll chaos, and synchronization. In general, multi-scroll chaotic attractors are verified by numerical simulations, but lately this has seen improvement in terms of theoretical proofs and analog circuit realizations [38, 39]. Despite the physical difficulties and limitations, today one is able to physically implement up to a maximum of 1-D 14-scroll, 2-D $14 \times 10$-grid scroll, and 3-D $10 \times 10 \times 10$-grid scroll chaotic attractors by electronic circuits [8]. It remains a technical challenge to produce more scrolls via hardware implementation, though numerical simulations can do much better. On the other hand, rigorous theoretical proofs are also quite difficult, due to the complex dynamical behaviors and the lack of suitable mathematical tools, especially for switching systems. Therefore, from a nonlinear dynamical physics point of view, to physically realize and also theoretically prove the existence of chaotic attractors with a multidirectional orientation and a large number of scrolls appears to be an important and stimulating subject for future research. Exploring the promising potential of multi-scroll chaotic attractors for engineering applications calls for more effort and greater endeavors. So, it is necessary and important to study the multi-scroll chaotic attractors’ control.

This paper addresses chaos control in multi-scroll chaotic attractors from saturated function series [40]. In Ref. [41] a fairly general method for the control of chaotic systems using the radial basis function networks (RBFNs) was proposed. On this basis, a similar but different method for the control of multi-scroll chaotic attractors using fuzzy neural networks (FNNs) is proposed. The FNNs are multilayer feedforward-type neural networks. The FNNs can automatically identify the fuzzy rules and tune the membership functions by modifying the connection weights of the networks using the back-propagation (BP) algorithm [42]. The method can identify the fuzzy model of the nonlinear part of the multi-scroll chaotic attractors automatically.

For many chaotic systems that can be decomposed into a sum of a linear and a non-linear part, under some mild conditions the FNNs are used in this paper to well approximate
the nonlinear part of the system dynamics. The resulting system is then dominated by
the linear part, with some small or weak residual nonlinearities resulting from the FNNs
approximation errors. Thus, a simple linear state feedback controller can be proposed, to
drive the multi-scroll chaotic attractors to the desired targets or periodic trajectory. Finally,
simulation results are presented to demonstrate the effectiveness of the proposed control
scheme.

II. DESCRIPTION FOR THE MULTI-SCROLL CHAOTIC ATTRACTORS

The multi-scroll chaotic attractors from saturated function series were designed and
analyzed in Ref. [40]. The saturated function series \( f(x; k, h, p, q) \) is as follows:

\[
\begin{align*}
f(x; k, h, p, q) &= \begin{cases} 
(2q + 1)k, & \text{if } x > qh + 1, \\
k(x - ih) + 2ik, & \text{if } |x - ih| \leq 1, \quad -p \leq i \leq q, \\
(2i + 1)k, & \text{if } ih + 1 < x < (i + 1)h - 1, \quad -p \leq i \leq q - 1, \\
-(2p + 1)k, & \text{if } x < -ph - 1.
\end{cases}
\end{align*}
\]

Consider the following 3-D linear autonomous system:

\[
\begin{cases}
\dot{x} = y, \\
\dot{y} = z, \\
\dot{z} = -ax - by - cz,
\end{cases}
\] (2)

where \( x, y, z \) are state variables, and \( a, b, c \) are positive real constants.

In the following, to create 1-D \( n \)-scroll chaotic attractors \( (n \geq 3) \), a saturated function
series controller is added to system (2), leading to

\[
\begin{cases}
\dot{x} = y, \\
\dot{y} = z, \\
\dot{z} = -ax - by - cz + d_1 f(x; k_1, h_1, p_1, q_1),
\end{cases}
\] (3)

where \( f(x; k_1, h_1, p_1, q_1) \) is defined by (1) and \( a, b, c, d_1 \) are positive constants.

Here, a saturated function series controller is added to system (2) for generating 2-D
\( n \times m \)-grid scroll chaotic attractors. The controlled system is described by

\[
\begin{cases}
\dot{x} = y - \frac{\partial}{\partial x} f(y; k_2, h_2, p_2, q_2), \\
\dot{y} = z, \\
\dot{z} = -ax - by - cz + d_1 f(x; k_1, h_1, p_1, q_1) + d_2 f(y; k_2, h_2, p_2, q_2),
\end{cases}
\] (4)

where \( f(x; k_1, h_1, p_1, q_1) \) and \( f(y; k_2, h_2, p_2, q_2) \) are defined by (1) and \( a, b, c, d_1, d_2 \) are positive constants.

Here, a saturated function series controller is added to system (4) for creating 3-D
\( n \times m \times l \)-grid scroll chaotic attractors. The controlled system is

\[
\begin{cases}
\dot{x} = y - \frac{\partial}{\partial x} f(y; k_2, h_2, p_2, q_2), \\
\dot{y} = z - \frac{\partial}{\partial y} f(z; k_3, h_3, p_3, q_3), \\
\dot{z} = -ax - by - cz + d_1 f(x; k_1, h_1, p_1, q_1) + d_2 f(y; k_2, h_2, p_2, q_2) + d_3 f(z; k_3, h_3, p_3, q_3),
\end{cases}
\]
with \( f(x; k_1, h_1, p_1, q_1) \), \( f(y; k_2, h_2, p_2, q_2) \), and \( f(z; k_3, h_3, p_3, q_3) \) defined by (1) and \( a, b, c, d_1, d_2, d_3 \) are positive constants.

Fig. 1 displays a 6-scroll chaotic attractor of system (3), where \( a = b = c = d_1 = 0.7, k_1 = 9, h_1 = 18, p_1 = 2, q_1 = 2 \). The Lyapunov exponent spectrum of this 6-scroll chaotic attractor includes \( LE_1 = 0.1486, LE_2 = 0, LE_3 = -0.8457 \). Fig. 2 shows a \( 6 \times 6 \)-grid scroll chaotic attractor of system (4), where \( a = b = c = d_1 = d_2 = 0.7, k_1 = k_2 = 50, h_1 = h_2 = 100, p_1 = q_1 = p_2 = q_2 = 2 \). The Lyapunov exponent spectrum of this \( 6 \times 6 \)-grid scroll attractor includes \( LE_1 = 0.1599, LE_2 = 0, LE_3 = -0.8622 \). Fig. 3 shows a \( 6 \times 6 \times 6 \)-grid scroll chaotic attractors of system (5), where \( a = d_1 = 0.7, b = c = d_2 = d_3 = 0.8, k_1 = 100, h_1 = 200, k_2 = k_3 = 40, h_2 = h_3 = 80, p_1 = p_2 = p_3 = q_1 = q_2 = q_3 = 2 \). The Lyapunov exponent spectrum of this \( 6 \times 6 \times 6 \)-grid scroll attractors includes \( LE_1 = 0.0885, LE_2 = 0, \) and \( LE_3 = -0.7157 \).

Fig. 1: 6-scroll chaotic attractors. (a) \( x - y - z \) space. (b) \( x - y \) plane projection.

III. PROBLEM STATEMENT

Consider the multi-scroll chaotic attractors of the form [41]

\[
\dot{X} = F(X) = f_L(X) + f_N(X) = AX + f_N(X),
\]

where \( X \in \mathbb{R}^3 \) denotes the state vector, \( f_L(X) = AX \) is the linear part and \( f_N(X) \) is the nonlinear part of the system dynamics \( F(X) \). In Eq. (6), \( f_L(X) \) is assumed to be known so that the controller for it can be designed, and \( f_N(X) \) is assumed to be unknown but the inputs of it can be measured.

Let us assume that there exists a nonlinear approximator having a good approximation ability and \( \hat{f}_N(X) \) is the approximation of its nonlinear part \( f_N(X) \). If the nonlinear part \( f_N(X) \) can be approximately “canceled” by \( \hat{f}_N(X) \), then the resulting system
is dominated by the linear part with some small or weak residual nonlinearity due to the approximation error

\[ \tilde{f}_N(X) = f_N(X) - \hat{f}_N(X) \approx 0. \] (7)

In this case, a simple linear state-feedback controller with a nonlinear approximator for nonlinearity compensation can be designed for stabilization and tracking control.

To realize this concept, we need the following mild assumptions, which are quite standard in the nonlinear systems control literature:

(i) The states of the controlled systems (3), (4), and (5) are completely accessible at all times.

(ii) The input and the output values of the systems (3), (4), and (5) nonlinearities can be measured accurately.

Under these conditions, the proposed method is formalized in the next section.
IV. LINEAR CONTROL WITH NONLINEARITY COMPENSATION

We first observe that if the approximator \( \hat{f}_N(X) \) is subtracted from system (6), then it results in

\[
\dot{X} = AX + f_N(X) - \hat{f}_N(X) = AX + \tilde{f}_N(X).
\]  

(8)

For the time being, consider the ideal case where the approximation error is identically zero, i.e., \( \tilde{f}_N(X) \equiv 0 \). The resulting dynamics is purely linear for which a linear controller can be used

\[
\dot{X} = AX + Bu(t),
\]  

(9)

choose \( B \in \mathbb{R}^{3 \times 1} \), \( \{A, B\} \) is assumed to be controllable and \( u(t) \) is a scalar control input of the form

\[
u(t) = -K^T X + v(t),
\]  

(10)

in which \( v(t) \) is an external input (reference signal) and \( K = [k_1, k_2, k_3] \) is a constant feedback gain vector to be designed. \( K \) can be obtained by any pole placement method. With this control law, the closed-loop control system is

\[
\dot{X} = (A - BK)X + Bv(t),
\]  

(11)

where \( A - BK \) is a stable matrix (i.e., all eigenvalues of the matrix \( A - BK \) have strictly negative real parts) obtained by a suitable choice of \( K \).

If the ignored approximation error \( \hat{f}_N(X) \) is not zero, then the closed-loop system dynamics is

\[
\dot{X} = (A - BK)X + Bv(t) + \tilde{f}_N(X).
\]  

(12)

Without loss of generality, assume that the equilibrium point of the controlled system is the origin when \( v(t) = 0 \). Then the following well-known result [43] holds for Eq. (12).

From Theorem 4.3 in Ref. [43], the zero equilibrium point of Eq. (12) is asymptotically stable if

\[
\frac{\|\tilde{f}_N(X)\|}{\|X\|} \to 0, \text{ as } \|X\| \to 0,
\]  

(13)

where \( \|\cdot\| \) denotes the Euclidean norm of a vector.

Here, consider the possibility that the non-zero approximation error, \( \hat{f}_N(X) \neq 0 \), can affect the evolution of the system state so significantly that the system may not be stabilized by the linear controller. This problem, however, can be avoided by an appropriate assignment of the system poles (i.e., eigenvalues of \( A - BK \)), such that they are not too close to the imaginary axis provided that \( \hat{f}_N(X) \) is small.
As long as this approximation error is bounded, the state of the controlled system is guided by the linear control law to zero, so the magnitude of the state vector is reduced with time. And if condition (13) is satisfied by the approximator \( \hat{f}_N(X) \), \( \tilde{f}_N(X) \) is, in turn, made to be even smaller. As a result of the asymptotic evolution process, the state vector of the controlled system approaches zero.

Now, the remaining problem is how to approximate the nonlinear parts of the systems (3), (4), and (5) such that \( \tilde{f}_N(X) \) is bounded and satisfies the condition (13). In this paper, the FNNs are used as such a nonlinear approximator.

The FNNs are multilayer feedforward-type neural networks [42]. In this paper we choose the Gaussian function as a membership function. A schematic diagram of the FNNs structure with 2 input variables, 6 term nodes for each input variable, 1 output node, and 36 rule nodes is shown in Fig. 4. The system consists of four layers. Nodes in layer one are input nodes, which represent input linguistic variables. Nodes in layer two are membership nodes, which act like membership functions. Each membership node is responsible for mapping an input linguistic variable into a possibility distribution for that variable. The rule nodes reside in layer three. Taken together, all the layer three nodes form a fuzzy rule base. Layer four, the last layer, contains the output variable nodes. To adaptively tune the FNNs’ parameters, we adopted the method which was proposed in Ref. [42] (see Appendix for the brief description of the learning algorithms). Based on the universal uniform approximation property of the fuzzy basis functions (FBFs) [44], the approximation error \( \tilde{f}_N(X) \) is small and satisfies the condition (13), provided that the number of rules of the FNNs used is large enough and the learning time is long enough, which is guaranteed by theory.

The final controlled system, equipped with FNNs as the nonlinear approximator, is illustrated in Fig. 5, where the block of adaptive algorithm provides the adjusted parameters at every iteration.

V. COMPUTER SIMULATIONS

In this section, some simulation results of the proposed control method for the multi-scroll chaotic attractors are presented. The structure of the FNNs is 2-12-36-1, with on-line adaptive tuning of the FNNs’ parameters. For the systems (3), (4), and (5) the matrix \( A \) is given by

\[
A = \begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
-0.7 & -0.7 & -0.7
\end{bmatrix}.
\]

We choose \( B = [1, 1, 1]^T \), \( K = [1.3387, 1.8226, 0.3387] \) to satisfy \( A - BK \) is a stable matrix. The initial state of the systems (3), (4), and (5) is \( x = 6 \), \( y = 8 \), \( z = -7 \). The tracking errors are defined as \( e_1(t) = x(t) - x_d \), \( e_2(t) = y(t) - y_d \), \( e_3(t) = z(t) - z_d \).

In system (3), let \( x \) tracking be \( x_d \), \( y \) tracking be \( y_d \) and \( z \) tracking be \( z_d \), where \( x_d \)}
y_d, and z_d are given by
\[
\begin{align*}
    x_d &= 2(\sin(\pi t) + \sin(\pi t/2)), \\
    y_d &= 6, \\
    z_d &= 10.
\end{align*}
\] (14)

The controlled system (3) is given by
\[
\begin{align*}
    \dot{x} &= y + 2\pi \cos(\pi t) + \pi \cos(\pi t/2) - 6 + u_1, \\
    \dot{y} &= z - 6 + u_2, \\
    \dot{z} &= -ax - by - cz + d_1 f(x; k_1, h_1, p_1, q_1) - d_1 f_N X(x - x_d, f_N X) + d_1 x_d + 11.2 + u_3,
\end{align*}
\] (15)

where \( u_1 = u_2 = u_3 = -1.3387(x - x_d) - 1.8226(y - y_d) - 0.3387(z - z_d) \), the other parameters are given in Section I. The FNNs’ initial centers and weights are randomly selected, i.e., let \( m(0) \in [5.5, 6.5] \), \( w(0) \in [-1, 1] \), and \( \sigma(0) = [1, 1, 1, 1, 1, 1] \), where \( m \) denotes centers, \( w \) denotes weights, and \( \sigma \) denotes widths. The tracking result is shown in Figs. 6–7.

In system (4), let \( x \) tracking be \( x_d \), \( y \) tracking be \( y_d \), and \( z \) tracking be \( z_d \), where \( x_d \), \( y_d \), and \( z_d \) are given by
\[
\begin{align*}
    x_d &= \sin(\pi t), \\
    y_d &= 4(\sin(\pi t/2) + \sin(\pi t/4)), \\
    z_d &= -6.
\end{align*}
\] (16)
The controlled system (4) is given by

\[
\begin{align*}
\dot{x} &= y - f(y; k_2, h_2, p_2, q_2) + \hat{f}_{NY}(y - y_d, f_{NY}) - y_d + \pi \cos(\pi t) + u_1, \\
\dot{y} &= z + 6 + 2\pi \cos(\pi t/2) + \pi \cos(\pi t/4) + u_2, \\
\dot{z} &= -a(x - x_d) - b(y - y_d) - c(z - z_d) + d_1(f(x; k_1, h_1, p_1, q_1) + f(y; k_2, h_2, p_2, q_2)) \\
&\quad - d_1(\hat{f}_{NX}(x - x_d, f_{NX}) + \hat{f}_{NY}(y - y_d, f_{NY})) + u_3,
\end{align*}
\]

where \( u_1 = u_2 = u_3 = -1.3387(x-x_d) - 1.8226(y-y_d) - 0.3387(z-z_d) \), the other parameters
FIG. 7: The error response of system (15).

are given in Section 1. The FNNs’ initial centers and weights are randomly selected, i.e., let
$m(0) \in [7.5, 8.5], w(0) \in [-1, 1], \text{ and } \sigma(0) = [1, 1, 1, 1, 1, 1].$ The tracking result is shown
in Figs. 8–9.

FIG. 8: The state response of system (17).

In system (5), let $x$ tracking be $x_d$, $y$ tracking be $y_d$ and $z$ tracking be $z_d$, where $x_d$,
$y_d$, and $z_d$ are given by

$$
\begin{align*}
    x_d &= \sin(\pi t), \\
y_d &= 4(\sin(\pi t/2) + \sin(\pi t/4)), \\
z_d &= 3(\sin(\pi t/3) + \sin(\pi t/6)).
\end{align*}
$$

(18)
The controlled system (5) is given by

\[
\begin{align*}
\dot{x} &= y - f(y; k_2, h_2, p_2, q_2) + \hat{f}_{NY}(y - y_d, f_{NY}) - y_d + \pi \cos(\pi t) + u_1, \\
\dot{y} &= z - f(z; k_3, h_3, p_3, q_3) + \hat{f}_{NZ}(z - z_d, f_{NZ}) - z_d + 2\pi \cos(\pi t/2) + \pi \cos(\pi t/4) + u_2, \\
\dot{z} &= -a(x - x_d) - b(y - y_d) - c(z - z_d) + d_1 f(x; k_1, h_1, p_1, q_1) + d_2 f(y; k_2, h_2, p_2, q_2) \\
&\quad + d_3 f(z; k_3, h_3, p_3, q_3) - d_1 \hat{f}_{NX}(x - x_d, f_{NX}) - d_2 \hat{f}_{NY}(y - y_d, f_{NY}) - d_3 \hat{f}_{NZ}(z - z_d, f_{NZ}) \\
&\quad + \pi \cos(\pi t/3) + \frac{\pi}{2} \cos(\pi t/6) + u_3, \\
\end{align*}
\]

where, \(u_1 = u_2 = u_3 = -1.3387(x - x_d) - 1.8226(y - y_d) - 0.3387(z - z_d)\), the other parameters are given in Section I. The FNNs’ initial centers and weights are randomly selected, i.e., let \(m(0) \in [-6.5, -5.5]\), \(w(0) \in [-1, 1]\), and \(\sigma(0) = [1, 1, 1, 1, 1, 1]\). The tracking result is shown in Figs. 10–11.
VI. CONCLUSIONS

In this paper, a general method for the control of multi-scroll chaotic attractors using a FNNs approximator is proposed. The method utilizes the excellent uniform functional approximation ability of the FNNs and the classical linear state-feedback control design technique. The FNNs incorporated the advantages of fuzzy inference and neuron-learning. The FNN possesses the merits of low-level learning and the computational power of the neural networks, and the high-level human knowledge representation and the thinking of fuzzy theory. In essence, the proposed method belongs to intelligent control. To the best of our knowledge, there are few theoretical results by using conventional methods to control multi-scroll chaotic attractors of diverse structures chaotic dynamical systems [8]. So, the FNNs-based method is more appropriate to control multi-scroll chaotic attractors than those of conventional methods, due to the multi-scroll chaotic attractors having more complex dynamical behavior than the general topologically simple chaotic attractors. In addition, from the control point of view, the analysis has shown that this control method is easy to implement. Simulation results for the multi-scroll chaotic attractors from a saturated function have shown that the proposed control scheme is very effective in stabilizing chaotic systems and controlling the chaotic states to periodic target orbits.

Appendix A

The learning algorithms used in this paper for each of the three adjustable parameters of the FNNs are briefly described [42]. The mathematical expressions are all based on the structure of Fig. 4.
A.1 Reasoning method

In this paper, we consider the case where the fuzzy rule base consists of \( M \) rules in the following form [44]:

\[
R_j: \text{If } x_1 \text{ is } A_{j1} \text{ and } x_2 \text{ is } A_{j2} \text{ and } \ldots \text{ and } x_n \text{ is } A_{jn}, \text{ THEN } z \text{ is } B^j,
\]  
(A.1)

where \( j = 1, 2, \ldots, M \), \( x_i (i = 1, 2, \ldots, n) \) are the input variables to the fuzzy neural networks, \( z \) is the output variable of the fuzzy neural networks, and \( A^j \) and \( B^j \) are linguistic terms characterized by fuzzy membership functions \( \mu_{A^j}(x) \) and \( \mu_{B^j}(z) \), respectively.

The Gaussian membership function is defined by

\[
\mu_{A^j}(x) = \exp\left[ -\frac{(x - m_{ij})^2}{\sigma_{ij}^2} \right],
\]  
(A.2)

where \( m_{ij} \) and \( \sigma_{ij} \) are, respectively, the center and the width of the Gaussian function in the \( j \)th term of the \( i \)th input linguistic variable.

Definition 1: Define fuzzy basis functions (FBFs) as [44]

\[
p_j(x) = \prod_{i=1}^{n} \frac{\mu_{A^j_i}(x_i)}{\sum_{j=1}^{M} \prod_{i=1}^{n} \mu_{A^j_i}(x_i)} (j = 1, 2, \ldots, M),
\]  
(A.3)

where \( \mu_{A^j_i}(x_i) \) are the Gaussian membership functions (A.2). Then, by the singleton fuzzifier, product inference, centroid defuzzifier, and Gaussian membership function, the output of the fuzzy neural networks is defined as

\[
f(x) = \sum_{j=1}^{M} p_j(x)\theta_j,
\]  
(A.4)

where \( \theta_j \in \mathbb{R} \) is an adjustable parameter.

A.2 Basic nodes operation

Next, we shall indicate the signal propagation and the basic function of every node in each layer. Consider that the structure of the FNNs is 2-12-36-1.

Layer 1: input layer
For the \( j \)th node of layer 1, the net input and the net output are represented as:

\[
net^1_j = x^1_i, \quad y^1_j = net^1_j,
\]  
(A.5)

where \( x^1_i \) represents the \( i \)th (\( i = 1, 2 \)) input of layer 1, \( j = 1, 2 \).

Layer 2: membership layer
In this layer, each node performs a membership function. The Gaussian function, a particular example of radial basis functions, is adopted here as a membership function. Then,

\[
net^2_j = \mu_{A^j}(m_{ij}, \sigma_{ij}) = \frac{(x^2_i - m_{ij})^2}{(\sigma_{ij})^2}, \quad y^2_j = f^2_{ij}(net^2_j) = \exp(net^2_j),
\]  
(A.6)

where \( m_{ij} \) and \( \sigma_{ij} \) are, respectively, the center and the width of the Gaussian function in the \( j \)th term of the \( i \)th input linguistic variable \( x^2_i \).

Layer 3: rule layer
The links in this layer are used to implement the antecedent matching. The matching operation or the fuzzy AND aggregation operation is chosen as the simple PRODUCT operation instead of the MIN operation. Then, for the \( j \)th rule node

\[
net^3_j = x^1_1 \cdot x^1_2, \quad y^3_j = net^3_j,
\]  
(A.7)

Layer 4: output layer
Since the overall net output is a linear combination of the consequences of all rules, the net input and output of the jth node in this layer are simply defined by

\[ net^4_j = \sum_{i=1}^{M} \theta_{ij} x^4_i, \quad y^4_j = net^4_j, \]  

(A.8)

where \( y^4_j \) denotes the output of the FNNs, and \( \theta_{ij} \) is the link weight between layer 3 and layer 4.

Note that \( x^k_i \) denotes the input of ith node, \( net^k_j \), \( y^k_j \) denote the net output of jth node and the output of jth node, respectively, and the superscript denotes the layer number. The learning process to train the fuzzy neural networks will be discussed in the following section.

### A.3 Supervised gradient descent learning

#### Layer 4: If the cost function to be minimized is defined as

\[ E = \frac{1}{2} (d - y^4)^2 = \frac{1}{2} e^2, \]  

(A.9)

where \( d \) is the desired output and \( y^4 \) is the current output, the error term to be propagated is given by

\[ \delta^4_1 = -\frac{\partial E}{\partial net^4_1} = d - y^4 = e, \]  

(A.10)

then, the weight \( \theta(l = 1, 2, \ldots, 36) \) is updated by the amount

\[ \Delta \theta_i = -\frac{\partial E}{\partial \theta_i} = -\frac{\partial E}{\partial net^4_1} \frac{\partial net^4_1}{\partial \theta_i} = \delta^4_1 \cdot \theta^4_j \]  

(A.11)

Layer 3: Only the error term needs to be calculated and propagated.

\[ \delta^3_j = \frac{\partial E}{\partial net^3_j} = e \]  

(A.12)

Layer 2: The multiplication operation is done in this layer. The adaptive rule for \( m_{ij} \) and \( \sigma_{ij} \) are as follows. First, the error term is computed,

\[ \delta^2_j = \frac{\partial E}{\partial net^2_j} = \frac{\partial E}{\partial y^2_j} \frac{\partial y^2_j}{\partial net^2_j} - \left( \sum_{k} \frac{\partial E}{\partial net^3_k} \frac{\partial net^3_k}{\partial y^2_j} \right) \cdot \left( \sum_{k} \frac{\partial y^2_k}{\partial net^3_k} \right) = (\sum_{k} \delta^3_k \cdot y^2_k) \cdot y^2_j (j = 1, \ldots, 36), \]  

(A.13)

where the subscript \( k \) denotes the rule node in connection with the jth node in Layer 2, and \( i \) denotes the other node in Layer 2 with connection with the \( k \)th node in Layer 3 \( (i \neq j) \).

Then, the adaptive rule of \( m_{ij} \) is

\[ \Delta m_{ij} = -\frac{\partial E}{\partial m_{ij}} = -\frac{\partial E}{\partial net^2_j} \frac{\partial net^2_j}{\partial m_{ij}} = \delta^2_j \cdot \frac{2(y^2_i - m_{ij})}{\sigma^2_{ij}}, \]  

(A.14)

and the adaptive rule of \( \sigma_{ij} \) is

\[ \Delta \sigma_{ij} = -\frac{\partial E}{\partial \sigma_{ij}} = -\frac{\partial E}{\partial net^2_j} \frac{\partial net^2_j}{\partial \sigma_{ij}} = \delta^2_j \cdot \frac{2(y^2_i - m_{ij})^2}{\sigma^4_{ij}}, \]  

(A.15)

where \( i = 1, 2; j = 1, 2, \ldots, 6 \).

The parameters of FNNs can be adjusted using the gradient method at the \( n \)th time step:

\[ \theta_l(n + 1) = \theta_l(n) + \eta_1 \Delta \theta_l(n), \]  

(A.16)

\[ m_{ij}(n + 1) = m_{ij}(n) + \eta_2 \Delta m_{ij}(n), \]  

(A.17)

\[ \sigma_{ij}(n + 1) = \sigma_{ij}(n) + \eta_3 \Delta \sigma_{ij}(n), \]  

(A.18)

where \( \eta_1, \eta_2, \) and \( \eta_3 \) are learning rates. This completes the derivation of the supervised gradient descent learning algorithm.
Acknowledgements

The author would like to thank the reviewers for their valuable and detailed comments that helped improve the contents and the presentation of this paper. This research is supported by the National Natural Science Foundation of China (No: 60573172), the Superior University doctor subject special scientific research foundation of China (No: 20070141014) and the National Natural Science Foundation of Liaoning province (No: 20082165).

References

* Electronic address: linda_740609@yahoo.cn
† Electronic address: wangxy@dlut.edu.cn