Fuzzy rules emulated network and its application on nonlinear control systems

C. Treesatayapun a,b,*

a Robotics and Advanced Manufacturing Division, Research Center for Advanced Studies, CINVESTAV-Saltillo, Mexico
b Department of Electrical Engineering, North-Chiangmai University, Chiangmai, Thailand

Received 20 February 2003; received in revised form 9 February 2007; accepted 27 March 2007
Available online 20 September 2007

Abstract

In this paper, the discrete-time nonlinear systems identification and control based on an adaptive filter are introduced. This adaptive filter is implemented using the adaptive network called Multi Input Fuzzy Rules Emulated Network (MIFREN). Inspired by the neuro-fuzzy network, the structure of MIFREN resembles the human knowledge in the form of fuzzy IF-THEN rules. The initial value of MIFREN’s parameters can be easily selected based on the human knowledge. Then the on-line adaptive process is performed to fine tune these parameters, the convergence of the adaptive process is proven by using Lyapunov-theory-based Adaptive Filtering (LAF). In the control system application, MIFREN is applied to control various selected nonlinear systems together with the proposed control law. Computer simulation results indicate that the proposed controller is able to control the target systems satisfactorily.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Fuzzy control; Neuro-fuzzy control; Neuro-fuzzy learning algorithm; Direct adaptive control

1. Introduction

Recently, the integration between fuzzy logic and neural network known as the neuro-fuzzy or the fuzzy neural network (FNN) has drawn attention from many researchers in the field of control engineering[2,3,11,12,14] and [21]. Combining a learning ability of neural network and human-like reasoning of fuzzy logic makes FNN become a very flexible intelligent control technique for many control applications. However, it is noticed that most proposed FNN control systems have complex network’s structure and require complicate learning algorithms to adjust their parameters in order to achieve satisfied performances. For examples, in [9,6] an FNN controller based on the well-known T–S fuzzy is proposed. All nodes in the membership layer are connected to every node in the rule layer. This means that the number of weight parameters to be adjusted is quickly increased when a new fuzzy rule is added to the control rules. Moreover, their membership function’s parameters are not directly adjusted. In [7] a recurrent method is proposed to improve the performance of FNN, however the network structure becomes more complex. In [8], a technique for pruning of redundant nodes and incompatible rules of the neuro-fuzzy network has been introduced, but it needs more designed parameters and complex calculation.

In this paper, the Multi Input Fuzzy Rules Emulated Network (MIFREN) which is able to handle multiple input signals is proposed. It is applied to identify a discrete-time nonlinear system. The control laws for discrete-time nonlinear plant are devised by using the adaptive filter based on MIFREN. The convergence of both nonlinear function error and the output response error is verified by Lyapunov theory.

Furthermore, the expanded application of MIFREN is proposed. The proposed controller consists of an adaptable network called Multi Input Fuzzy Rules Emulated Network (MIFREN) As will be seen in the following sections, the structure of MIFREN resembles the human knowledge in the form of fuzzy control rules. This structure is simple and allows the initial setting of network parameters to be intuitively selected. After setting its parameters, an on-line adaptation is performed during its operation to fine tune the values. Hence, the controller is able to adapt itself to the change of environment.

This paper is organized as follows. First, the MIFREN is introduced in Section 2.1. Next, the parameter adaptation technique is explained and its convergence is analyzed using Lyapunov theory-based adaptive filter (LAF). In Section 3, the
simulation results of using MiFREN as the system identification are illustrated. In section 4 the structure of control system based on MiFREN is introduced. The control law which is suitable for MiFREN is discussed in Section 4.1. Then some computer simulation results when applying MiFREN to control the various selected nonlinear discrete-time systems are given in Section 4.2. Finally, some conclusions are given in the last section.

2. Multi Input Fuzzy Rules Emulated Network

In this section, the structure of MiFREN is explained in some details. Then the parameter adaptation method is derived and the convergence of the error is discussed.

2.1. Structure of MiFREN

For a fuzzy inference system with $n$ inputs where each input has $r$ fuzzy state, the total number of fuzzy rules is then $r^n$. Each fuzzy IF-THEN rule can be represented by

**Rule $k$:**

If ($I_1$ is $A_{k_1,1}$) and ($I_2$ is $A_{k_2,2}$) ... and ($I_n$ is $A_{k_n,n}$)

Then $O_k = B_k$

where $I_j$ is the value of the $j$-th crisp input. $A_{i,j}$ denotes the $i$-th fuzzy state for $j$-th input, since there is $r$ possible fuzzy state, then $i \in \{1, 2, \ldots, r\}$. We relate the $k$-th rule to the index $\{k_i\}_{i=1}^n$ via $k = 1 + \sum_{i=1}^n (k_i - 1)r^i$, thus $k \in \{1, 2, \ldots, r^n\}$. And $O_k$ is the fuzzy output of this rule which belongs to the fuzzy set $B_k$. After all rules have been processed, the crisp output $O$ is obtained from $\{O_k\}_{k=1}^{r^n}$ using some defuzzification schemes.

MiFREN is derived based on these fuzzy rules, its structure can be decomposed into 5 layers as shown in Fig. 1. The function of each layer is as follows:

**Layer 1:** Each input $I_j$ ($j = 1, \ldots, n$) in this layer is sent to the corresponding nodes in the next layer directly. Thus there is no computation in this layer.

**Layer 2:** This is called the input membership function (MF) layer. Each node in this layer contains a membership function corresponding to one linguistic level (e.g. negative, nearly zero, etc.). The output at the $i$-th node for the input $I_j$ is denoted by $\mu_{A_{i,j}}$.

**Layer 3:** This layer corresponds to the fuzzy inference. The number of nodes in this layer is $r^n$ nodes. The output signal at each node in the layer is calculated as

$$ f_k = f_{k_1 \ldots k_n} = \prod_{j=1}^{n} \mu_{A_{i,j}} $$

where $k_j \in \{1, 2, \ldots, r\}$ and $k = 1, 2, \ldots, r^n$.

**Layer 4:** This layer may be considered as defuzzification step. It is called the Linear Consequence (LC) layer. There are also $r^n$ nodes in this layer. The output at the $k$-th node in this layer is calculated by

$$ O_k = \beta_k f_k + b_k, $$

where $\beta_k$ and $b_k$ are parameters of $k$-th node. For simplicity, in this work, we define $b_k = 0$ for all $k$. Then Eq. (2) becomes

$$ O_k = \beta_k f_k. $$

**Layer 5:** The structure of this layer is similar to the output layer of an artificial neural network with unity weight. The output of the MiFREN, $O$, is calculated by

$$ O = f_{\text{MF}}(I_1, \ldots, I_n) = \sum_{k=1}^{r^n} O_k = \beta^T F, $$

where $\beta = [\beta_1, \beta_2, \ldots, \beta_{r^n}]^T$ and $F = [f_1, f_2, \ldots, f_{r^n}]^T$.

As will be seen in the computer simulation results, this decomposition into 5 layers enables the user to intuitively set the initial value of MiFREN’s parameters.
2.2. Parameter adaptation algorithm

Since the initial setting of MiFREN parameters are just rough estimation based on an expert’s experience and knowledge. It is necessary to fine tune these values in order to cope with environmental change and also to improve the system performance. Here, an adaptive technique based on adaptive filter[17] is proposed to adjust all LC’s parameters i.e. $\beta$ during the system operation. We define the error function as

$$e(k) = y_p(k) - y(k),$$

(5)

where $y_p(k)$ is the desired response and $y(k)$ is the output of MiFREN at time index $k$, respectively. The system configuration is illustrated in Fig. 2.

It is desired to adjust $\beta$ in Eq. (4) such that $e(k)$ becomes zero. The value of parameter $\beta$ is updated at each time $k$ by

$$\beta(k) = \beta(k - 1) + \alpha(k)g(k),$$

(6)

where

$$g(k) = \frac{F(k)}{\|F(k)\|^2} \left(1 - \frac{|e(k - 1)|}{|\alpha(k)| + e}\right),$$

(7)

and

$$\alpha(k) = y_p(k) - \beta(k - 1)^T F(k),$$

(8)

where $\eta$ and $e > 0$ are predefined constants.

**Theorem 1.** If the error $e(k)$ is finite for all $k$ and all MiFREN parameters are adjusted according to Eqs. (6)–(8) with $|\eta| < 1$, then $\lim_{k \to \infty} |e(k)| = 0$.

**Proof.** The output of MiFREN, $y(k)$, is computed as,

$$y(k) = \beta(k)^T F(k).$$

(9)

Thus the error signal becomes

$$e(k) = y_p(k) - y(k),
= y_p(k) - \beta(k)^T F(k),
= y_p(k) - \frac{\beta(k - 1)^T + \alpha(k)g(k)}{\|F(k)\|^2} F(k),
= y_p(k) - \frac{\beta(k - 1)^T + \alpha(k)g(k)}{\|F(k)\|^2} F(k) - \alpha(k)g(k)^T F(k),
= \frac{\alpha(k)}{\alpha(k)} F(k) F(k) \left(1 - \frac{|e(k - 1)|}{|\alpha(k)| + e}\right),
= \eta \frac{|e(k - 1)|}{|\alpha(k)| + e} \alpha(k),$$

or

$$|e(k)| = |\eta| \frac{|\alpha(k)|}{|\alpha(k)| + e} |e(k - 1)|.$$  (10)

Since $|\eta| < 1$, $|\alpha(k)| < |\alpha(k)| + e$ and $e(k)$ is finite, we obtain $\lim_{k \to \infty} |e(k)| = 0$.  \[\square\]

![Fig. 2. Configuration of the MiFREN based adaptive filter model.](image)

Define the Lyapunov function $V(k) = e^2(k)$ then

$$\Delta V(k) = V(k) - V(k - 1),
= e^2(k) - e^2(k - 1),
= \frac{\eta^2 \alpha^2(k)}{(|\alpha(k)| + e)^2} e^2(k - 1) - e^2(k - 1),
= \left[\frac{\eta^2 \alpha^2(k)}{(|\alpha(k)| + e)^2} - 1\right] e^2(k - 1).$$

For $\Delta V(k) < 0$ we must have

$$\frac{\eta^2 \alpha^2(k)}{(|\alpha(k)| + e)^2} < 1 \Rightarrow |\eta| < \frac{|\alpha(k)| + e}{|\alpha(k)|},$$

which is true if $|\eta| < 1$. The convergence is verified by computer simulations as shown in the Section 3.

3. MiFREN system identification

In this section the application of MiFREN as the nonliner system identification is introduced. Based on the selected nonlinear system and the proposed hybrid learning algorithm, the simulation results can be obtained as the following:

**3.1. Test system configuration**

To verify the proposed adaptation technique, the example of using MiFREN to identify the selected discrete-time nonlinear system has been performed via computer simulations. The test system configuration is illustrated in Fig. 3.

![Fig. 3. Configuration of the identification test system.](image)
3.2. Example

The system equation of the plant [13] to be identified is described as
\[ y_p(k + 1) = \frac{y_p(k)}{1 + y_p^2(k)} + u^3(k), \]  
(11)
where
\[ u(k) = \frac{2\pi k}{25} + \sin \frac{2\pi k}{10}. \]  
(12)
In this example, at time index \( k \), the plant output \( y_p(k) \) and the control effort \( u(k) \) are fed as the input signals of MiFREN. The estimated plant output signal \( y(k) \) is obtained as
\[ y(k + 1) = f_{MF}(y_p(k), u(k)), \]  
(13)
where \( f_{MF}() \) represents the function of MiFREN. The If-Then rules are defined as the following:

**Rule 1** If \( y_p(k) \) is \( N \) and \( u(k) \) is \( N \) then \( y_1(k + 1) = \beta_1 f_1(k) \),

**Rule 2** If \( y_p(k) \) is \( N \) and \( u(k) \) is \( Z \) then \( y_2(k + 1) = \beta_2 f_2(k) \),

**Rule 3** If \( y_p(k) \) is \( N \) and \( u(k) \) is \( P \) then \( y_3(k + 1) = \beta_3 f_3(k) \),

**Rule 4** If \( y_p(k) \) is \( Z \) and \( u(k) \) is \( N \) then \( y_4(k + 1) = \beta_4 f_4(k) \),

**Rule 5** If \( y_p(k) \) is \( Z \) and \( u(k) \) is \( Z \) then \( y_5(k + 1) = \beta_5 f_5(k) \),

**Rule 6** If \( y_p(k) \) is \( Z \) and \( u(k) \) is \( P \) then \( y_6(k + 1) = \beta_6 f_6(k) \),

**Rule 7** If \( y_p(k) \) is \( P \) and \( u(k) \) is \( N \) then \( y_7(k + 1) = \beta_7 f_7(k) \),

**Rule 8** If \( y_p(k) \) is \( P \) and \( u(k) \) is \( Z \) then \( y_8(k + 1) = \beta_8 f_8(k) \),

**Rule 9** If \( y_p(k) \) is \( P \) and \( u(k) \) is \( P \) then \( y_9(k + 1) = \beta_9 f_9(k) \),

Here, \( N, Z \) and \( P \) denote negative, zero, and positive linguistic level, respectively. The initial setting of \( \beta \) and \( F(k) \) are given as
\[ \beta_1 = -5 \]
\[ \beta_2 = -2 \]
\[ \beta_3 = -1 \]
\[ \beta_4 = 0 \]
\[ \beta_5 = 0 \]
\[ \beta_6 = 1 \]
\[ \beta_7 = 2 \]
\[ \beta_8 = 3 \]
\[ \beta_9 = 5 \]
where \( \mu_\square \) and \( \mu_\square \) are the corresponding membership functions of \( y_p(k) \) and \( u(k) \), respectively. These membership functions are illustrated in Fig. 4(a and b). In this simulation, the designed parameters \( \eta \) and \( \epsilon \) are set to 0.9 and 0.2, respectively. The time variation of adjustable parameters \( \beta(k) \) during the system operation are shown in Fig. 5. The actual plant output signal \( y_p(k) \) and the output of the identification system \( y(k) \) of this example are illustrated in Fig. 6. As also shown in Fig. 7, \( |e(k)| \) becomes zero eventually.

4. MiFREN adaptive controller

In this section, first, the structure of MiFREN and its usage as a control system are explained in some details. The proposed control system configuration can be shown in Fig. 8. Here, MiFREN stands as the nonlinear function approximation based on the parameters tuning in Section 2.2. The plant control signal \( u(k) \) and the control algorithm will be defined in Section 4.1.

4.1. Control Law

In general cases, the \( n \)-th order nonlinear discrete-time plant can be described by
\[
\begin{bmatrix}
  x_1(k + 1) \\
  x_2(k + 1) \\
  \vdots \\
  x_{n-1}(k + 1) \\
  x_n(k + 1)
\end{bmatrix} =
\begin{bmatrix}
  0 & 1 & 0 & \ldots & 0 \\
  0 & 0 & 1 & \ldots & 0 \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \ldots & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x_1(k) \\
  x_2(k) \\
  \vdots \\
  x_{n-1}(k) \\
  x_n(k)
\end{bmatrix}
+ \begin{bmatrix}
  0 \\
  0 \\
  \vdots \\
  0
\end{bmatrix},
\]  
(14)
and
\[ x_n(k + 1) = f(x(k)) + u(k) + d(k), \]  
(15)
where \( x(k) = [x_1(k) \ x_2(k) \ \ldots \ x_n(k)]^T \), \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is the unknown nonlinear function, \( d(k) \) is the unknown but bounded

![Fig. 4. Membership functions of (a) \( y_p(k) \) and (b) \( u(k) \).](image-url)
disturbance when \( \|d(k)\| \leq d_m \) and \( u(k) \) is the control effort. In this work, the unknown nonlinear function and disturbance are combined together as \( f_d(k) \) where

\[
f_d(k) = f(x(k)) + d(k).
\] (16)

To restore this unknown function \( f_d(k) \), our self-adjustable network called MIFREN. Let us define the sliding surface \( s(k) \) as

\[
s(k) = c[x(k) - x_d(k)] = ce(k),
\] (17)

where \( x_d(k) \) be the desired value of \( x \) at time \( k \) and \( e = [c_1 \cdots c_n] \) is a constant matrix. Note that \( e \) must not orthogonal to \( e \) and all roots must keep in the unit disk. Then \( s(k+1) \) is

\[
s(k+1) = c_1 e_1(k+1) + c_2 e_2(k+1) + \cdots + c_{n-1} e_{n-1}(k+1) + c_n e_n(k+1).
\] (18)

Fig. 5. Time variation of each \( \beta_i \).

Fig. 6. Time variation of the actual signal \( y_p(k) (\bullet) \) and the modeled signal, \( y(k) (\circ) \).

Fig. 7. Time variation of \( |e(k)| \).
Substitute (14), (15) and (16) into (18), we obtain

\[ s(k + 1) = c_1 e_2(k) + \cdots + c_{n-1} e_n(k) + c_n [f_d(k) + u(k) - x_{nd}(k + 1)] . \] (19)

Thus, let us define the reaching condition as

\[ s(k + 1) = -\nu s(k) , \] (20)

where the definition of the designed parameter \( \nu \) is \( 0 < \nu < 1 \).

Consider the defined condition in (20) and (19) again. We find that

\[-\nu s(k) = c_1 e_2(k) + \cdots + c_{n-1} e_n(k) + c_n [f_d(k) + u(k) - x_{nd}(k + 1)] , \] (21)

or

\[ u(k) = x_{nd}(k + 1) - f_d(k) \]

\[ -\frac{1}{c_n} [c_1 e_2(k) + \cdots + c_{n-1} e_n(k) + \nu s(k)] . \] (22)

The nonlinear term \( f_d(k) \) in (22) is assumed to be unknown. Moreover, we can use MiFREN in Section 2 to approximate \( f_d(k) \) as \( \hat{f}_d(k) \). The control effort \( u(k) \) in (22) can be rewritten as

\[ u(k) = x_{nd}(k + 1) - \hat{f}_d(k) \]

\[ -\frac{1}{c_n} [c_1 e_2(k) + \cdots + c_{n-1} e_n(k) + \nu s(k)] . \] (23)

Thus, the performance of the control law in (23) together with the function approximation of MiFREN can be introduced in the next subsection.

### 4.2. Computer simulation examples

In this section, the performance of MiFREN and the control law given in Section 4.1 can be demonstrated to control two types of nonlinear systems. Computer simulations have been performed to investigate the performance of MiFREN and the proposed controller.

#### 4.2.1. The nonlinear discrete-time system 1

In this work, the selected nonlinear discrete-time system [15] can be described as

\[ x_1(k + 1) = x_2(k) \quad x_2(k + 1) = f(x(k)) + u(k) + d(k) , \] (24)

where

\[ f(x(k)) = -\frac{5}{8} \left[ \frac{x_1(k)}{1 + x_2^2(k)} \right] + 0.3 x_2(k) , \] (25)

![Fig. 8. Control system configuration based on MiFREN.](image1)

![Fig. 9. Membership functions of states \( x_1 \) and \( x_2 \): Example 4.2.1.](image2)
and the unknown disturbance \( d(k) \) is defined as

\[
d(k) = \begin{cases} 
0 & \text{if } 0 \leq k < 2000, \\
1.5 & \text{if } 2000 < k \leq 6000.
\end{cases}
\]  

(26)

The controller has been designed to force the state \( x_2(k) \) to track the reference signal \( x_{2d}(k) \) given as

\[
x_{2d}(k) = \sin (\omega k T_s + \xi),
\]  

(27)

where \( \omega = 0.1, \xi = \frac{\pi}{3} \) and the sampling interval \( T_s = 50 \) ms.

The control parameters are given as \( c_1 = 0.1, \ c_2 = 0.5 \) and \( v = 0.2 \). The initial membership functions of states \( x_1 \) and \( x_2 \) are defined in Fig. 9.

The tracking performance and the disturbance rejection can be illustrated in Fig. 10. The result is close to the designed target. Fig. 11 represents the performance of nonlinear function approximation where \( e_f(k) \) are defined as

\[
e_f(k) = f_d(k) - \tilde{f}_d(k).
\]  

(28)

Fig. 11. Tracking performance of nonlinear function approximation: Example 4.2.1.

The time variation of sliding function is shown in Fig. 12. Finally, Fig. 13 illustrates the relation between \( e_1(k) \) and \( e_2(k) \) together with the sliding condition at \( s(k) = 0 \).

4.2.2. The nonlinear discrete-time system 2

The characteristic of the nonlinear discrete-time robotic plant [16] is given as

\[
\begin{bmatrix}
 x_1(k+1) \\
 x_2(k+1)
\end{bmatrix} =
\begin{bmatrix}
 0 & 1 \\
 0 & 0
\end{bmatrix}
\begin{bmatrix}
 x_1(k) \\
 x_2(k)
\end{bmatrix} +
\begin{bmatrix}
 0 \\
 f(x(k))
\end{bmatrix} +
\begin{bmatrix}
 0 \\
 1
\end{bmatrix} u(k),
\]  

(29)

where \( u(k) \) is the control signal, and

\[
f(x(k)) = (2T_s - 1)x_1(k) + 2(1 - T_s)x_2(k) \\
+ 10T_s^2 \sin (x_1(k)).
\]  

Fig. 12. Sliding function: \( s(k) \): Example 4.2.1.

Fig. 13. \( e_1(k) \leftrightarrow e_2(k) \): Example 4.2.1.
where $T_s$ is the sampling interval and equals to 0.01 s. The reference signal $x_d$ is set as

$$x_d(k+1) = \frac{x_{d1}(k+1)}{x_{d2}(k+1)} = \frac{\sin\left(\frac{\pi}{5}kT\right)}{\sin\left(\frac{\pi}{5}(k+1)T\right)}.$$ (30)

The control parameters are also given as the same in the previous example or $c_1 = 0.1, c_2 = 0.5$ and $\nu = 0.2$. The initial membership functions of states $x_1$ and $x_2$ are defined in Fig. 14.

The tracking performance can be illustrated in Fig. 15. The performance of nonlinear function approximation of MiFREN is represented in Fig. 16 where $e_f(k)$ are defined as

$$e_f(k) = f(k) - \hat{f}(k).$$ (31)

The time variation of sliding function is shown in Fig. 17. The relation between $e_1(k)$ and $e_2(k)$ together with the sliding condition at $s(k) = 0$ is drawn in Fig. 18. The final figure, Fig. 19, introduces the phase plane of state variables $x_1(k)$ and $x_2(k)$. 
5. Conclusions

The structure of an adaptive filter based on the self-adjustable network called MiFREN has been presented in this article. All linear parameters in MiFREN have been automatically tuned by Lyapunov-theory-based Adaptive Filtering (LAF). Under the designed parameters selecting $(\eta$ and $e$), the convergence of the proposed algorithm can be insisted by the simulation results to identify the selected nonlinear system.

For widely useful algorithm, the control law based on MiFREN identification of the system nonlinear function has been proposed. To introduce the accomplishment of this proposed algorithm, the applications of MiFREN to control various kinds of nonlinear systems have been performed.

In the simulation results, the proposed controller can force the plant's output to follow the reference signal and setting point satisfactorily. We are currently investigating other applications of MiFREN controller.

References