CHAPTER 3

Fuzzy Rules Emulated Network

In this chapter, the simple structure and computation network called Fuzzy Rules Emulated Network (FREN), which can emulate a set of fuzzy IF-THEN rules, is proposed. The network structure of FREN is explained firstly. Then the relationship with the Mamdani fuzzy inference system and the radial basis function network are given. Finally, the parameter adaptation method for both linear and nonlinear parameters is derived and the learning rate selection criteria is discussed.

3.1 Structure of Fuzzy Rules Emulated Network

For a single input single output system, A general fuzzy inference system can be represented by the IF-THEN as,

$$\text{RULE } i\text{: IF } I \text{ IS } A_i \text{ THEN } B_i = f_i(\mu_{A_i})$$

where I denotes the input variable of this fuzzy system. This rule indicates that if I is the crisp value which belongs to the fuzzy set A_i with the membership value of μ_{A_i} then the fuzzy value of the output of this rule, denoted by B_i, is equal to $f_i(\mu_{A_i})$. After all rules have been processed, the crisp output O is calculated using some defuzzification schemes.

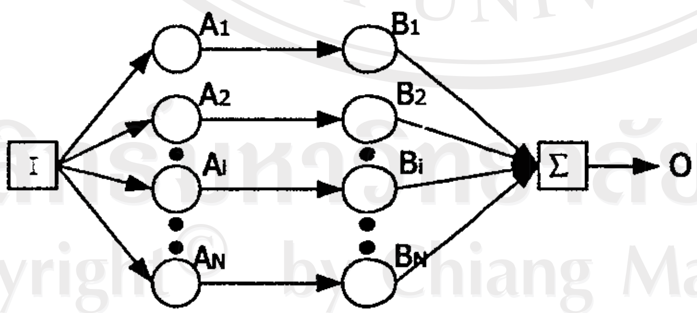

Figure 3.1: Structure of FREN

FREN is derived based on these fuzzy rules, its structure can be decomposed into 4 layers as shown in Fig.3.1. The function of each layer is as follows:

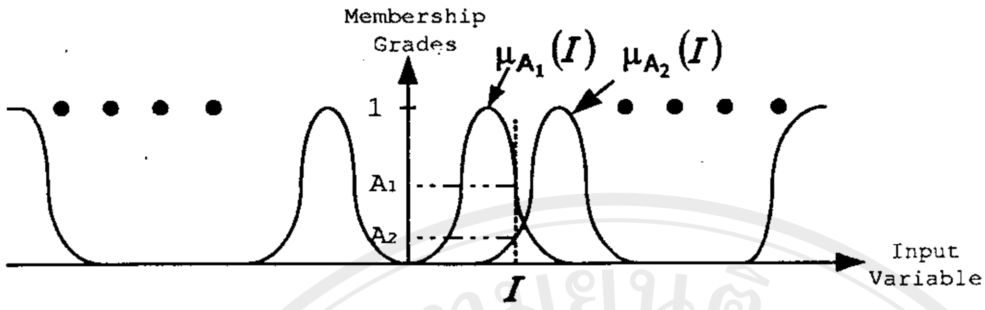

Figure 3.2: Example of membership function (MF)

Layer 1: The input I of this layer is sent to each node 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 is calculated by

$$A_i = \mu_{A_i}(I), \tag{3.1}$$

where $\mu_{A_i}(\cdot)$ denotes a membership function of a fuzzy set A at i-th node ($i = 1, 2, \ldots, N$). Examples of membership functions are given in Fig.3.2.

Layer 3: This layer may be considered as defuzzification step. It is called the Linear Consequence (LC) layer. There are also N nodes in this layer. The output at the i-th node in this layer can be calculated by

$$B_i = (h_i - k_i)A_i + k_i, \tag{3.2}$$

where h_i and k_i are parameters of i-th node. Examples of linear consequence are shown in Fig. 3.3.

Layer 4: The structure of this layer is similar to the output layer of an artificial neural network. The output of the FREN, O, is calculated in this layer as

$$O = \sum_{i=1}^{N} B_i. \tag{3.3}$$

From Eqs. (3.1) and (3.2), Eq. (3.3) can be rewritten as

$$O = \sum_{i=1}^{N} (h_i - k_i)\mu_{A_i}(I) + k_i. \tag{3.4}$$

If there is no bias then $k_i = 0$ and Eq.(3.4) becomes

$$O = \sum_{i=1}^{N} h_i \mu_{A_i}(I). \tag{3.5}$$

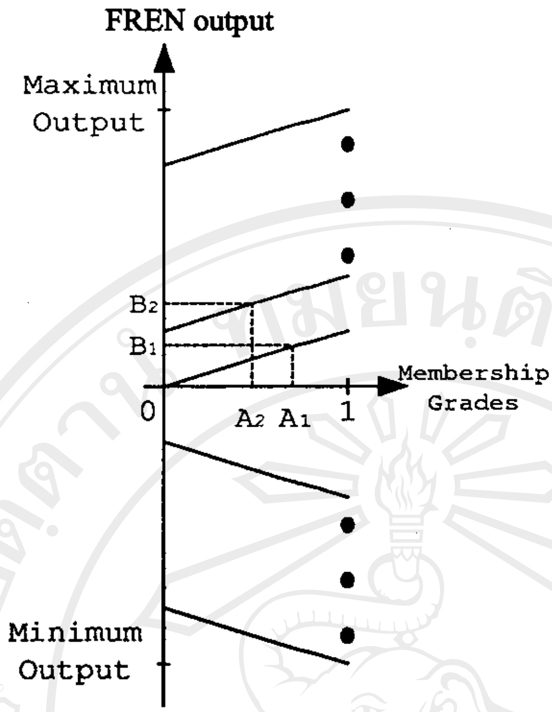

Figure 3.3: Example of linear consequence (LC)

This decomposition into 4 layers enables the designer to intutively set the initial value of FREN's parameters. As an example, consider the following 4 fuzzy rules of the input and the output relations,

RULE 1 IF I IS PL THEN O IS PL
RULE 2 IF I IS PM THEN O IS PM
RULE 3 IF I IS NM THEN O IS NM
RULE 4 IF I IS NL THEN O IS NL,

here PL, PM, NM and NL denote positive large, positive medium, negative medium and negative large linguistic level, respectively.

Assume that the input signal $I \in [-1, 1]$ and the output signal $O \in [-10, 10]$. The value of the output signal, O, is controlled by LC parameters (e.g. h_i and k_i for $i = 1, 2, 3, 4$.) In this example, h_1 is set to the maximum value ($h_1 = 10$) and h_4 is set to the minimum value ($h_4 = -10$). Other parameters are $h_2 = h_1/2 = 5$, $h_3 = h_4/2 = -5$, and $k_i = 0$ for $i = 1, 2, 3, 4$. Then, MF parameters are selected to cover the input range. The initial setting of all parameters can

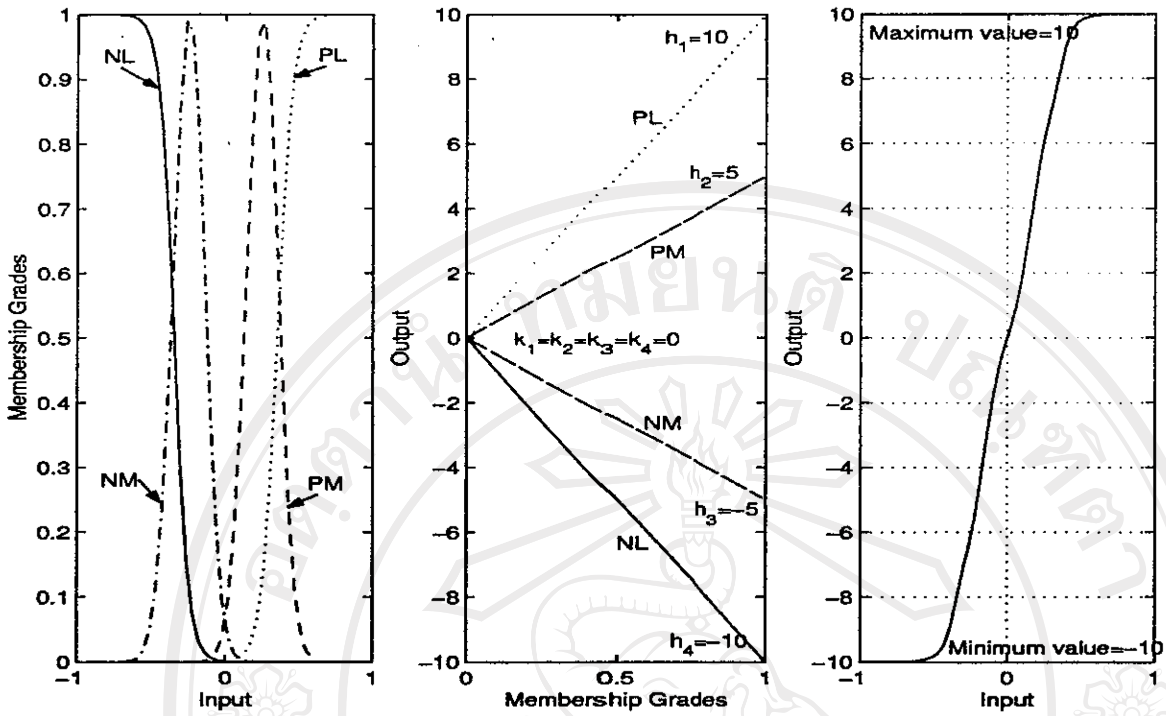

Figure 3.4: Example of FREN parameters setting.

be given as:

$$\text{RULE 1:}\quad A_1 = \mu_{A_1}(I) = \frac{1}{1+\exp\left[-20(I-0.35)\right]} \quad ; B_1 = 10A_1,$$

$$\text{RULE 2:}\quad A_2 = \mu_{A_2}(I) = \exp\left(-\left[\frac{I-0.25}{0.15}\right]^2\right) \quad ; B_2 = 5A_2,$$

$$\text{RULE 3:}\quad A_3 = \mu_{A_3}(I) = \exp\left(-\left[\frac{I+0.25}{0.15}\right]^2\right) \quad ; B_3 = -5A_3,$$

$$\text{RULE 4:}\quad A_4 = \mu_{A_4}(I) = \frac{1}{1+\exp\left[20(I+0.35)\right]} \quad ; B_4 = -10A_4.$$

The results of this setting is shown in Fig. 3.4. Note that these parameters are further adjusted using an on-line adaptive algorithm to fine tune the system performance.

3.2 Fren, RBF and Mamdani Fuzzy Logic

In this section, it will be shown that FREN is equivalent to the *Radial Basis Function* (RBF) network and the Mamdani fuzzy logic [28].

3.2.1 RBF

The structure of an RBF network with single input, single output and m nodes in its hidden layer is shown in Fig. 3.5. The network output, O_r, is obtained from

$$O_r = \sum_{i=1}^{m} \alpha_i \mu_i(I), \tag{3.6}$$

where I denotes the input signal, μ_i and α_i are the radial basis function and the weight parameter of i-th node, respectively. Consider Eq.(3.6) and Eq.(3.5), if all membership functions of FREN are selected as radial basis functions, $m = N$ and $\alpha_i = h_i$ then the output of FREN in Eq.(3.5) is equal to O_r.

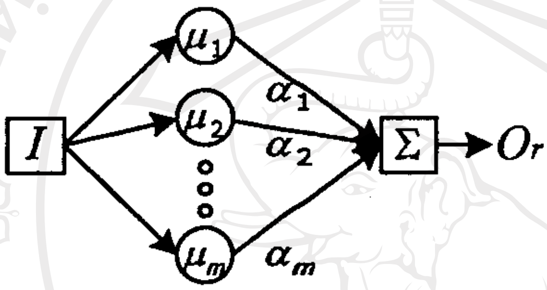

Figure 3.5: Radial basis function network structure.

3.2.2 Mamdani Fuzzy Logic

Consider the Mamdani fuzzy logic system [28] with single input, single output and m fuzzy control rules. Each rule is given by:

Rule i: IF I IS A_i THEN O_{fi} IS B_i

for $i = 1, 2, 3, \ldots, m$ and B_i and O_{fi} denote the output and a singleton value of the i-th rule, respectively.

The system output O_f obtained using a discrete center of area defuzzification method can be written as

$$O_f = \frac{\sum_{i=1}^{m} B_i \mu_{A_i}(I)}{\sum_{i=1}^{m} \mu_{A_i}(I)}. \tag{3.7}$$

Let $\sum_{i=1}^{m} \mu_{A_i}(I) = 1$ then we obtain

$$O_f = \sum_{i=1}^{m} B_i \mu_{A_i}(I). \tag{3.8}$$

Consider Eq.(3.8) and Eq.(3.5) , if $m = N$ and $B_i = h_i$ then O_f is equal to the output of FREN in Eq. (3.5).

Thus the FREN, the RBF network, and the Mamdani fuzzy logic can, in principle, give the same performances. However, the initial setting of RBF is normally defined as a random small value set [4, 28], while the Mamdani fuzzy inference needs the more precise parameter values since it lacks the self adaptation. As shown earlier, for FREN, the initial setting of MF can be selected to cover the range of input variable and LC parameters can be roughly estimated from the range of the output variable together with expert's knowledge.

3.3 Parameter Adaptation

The adaptation of FREN parameters is presented in this section. Since the initial setting of FREN parameters are just rough estimation based on a human expert experience. It is necessary to fine tune these values in order to cope with environmental change and also to improve the system performance.

3.3.1 Adaptation Based on Steepest Descent Algorithm

In this work, an adaptive technique based on the steepest descent technique is proposed to adjust all parameters, i.e., the shapes of membership functions and linear consequences, during the system operation. Firstly, we define the objective function as

$$\xi(k) = \frac{1}{2}\Big(t(k) - o(k)\Big)^2, \tag{3.9}$$

where $t(k)$ and $o(k)$ are the target and the FREN's output signal at time k respectively. It is desired to adjust all FREN's parameters, i.e., MF and LC parameters, to minimize this objective function. Here, the value of parameter P_i is updated at each time step by

$$P_i^{\text{new}} = P_i + \Delta P_i = P_i - \eta_i \frac{\partial \xi}{\partial P_i}, \tag{3.10}$$

where η_i is called the learning rate of i-th parameter. The term $\partial\xi/\partial P_i$ is calculated from

$$\frac{\partial \xi}{\partial P_i} = \frac{\partial \xi}{\partial o}\frac{\partial o}{\partial P_i}, \tag{3.11}$$

and

$$\frac{\partial \xi}{\partial o} = -[t(k) - o(k)] = -E(k). \tag{3.12}$$

Finally, Eq (3.10) becomes

$$P_i^{\text{new}} = P_i + \eta_i E(k)\frac{\partial o}{\partial P_i}. \tag{3.13}$$

3.3.2 Learning Rate Selection

The difficulty of adaptive method based on the steepest descent technique is how to select an appropriate value of the learning rate. Too large value of the learning rate may cause the objective function oscillates around its global minimum whereas too small value reduces the learning performance. In this subsection, we discuss how to select an appropriate learning rate based on Lyapunov's stability condition. Note that a similar approach has been suggested in [20].

Consider the following Lyapunov function

$$V(k) = \frac{1}{2}\Big(t(k) - o(k)\Big)^2 = \frac{1}{2}E^2(k). \tag{3.14}$$

The change of Lyapunov function is given by

$$\begin{aligned} \Delta V(k) &= V(k+1) - V(k) \\ &= \frac{1}{2}\Big(E^2(k+1) - E^2(k)\Big) \\ &= \Delta E(k)\Big(E(k) + \frac{1}{2}\Delta E(k)\Big), \end{aligned} \tag{3.15}$$

where $\Delta E(k) = E(k+1) - E(k)$ is the change of error. This can be approximated by

$$\Delta E(k) = \frac{\Delta E(k)}{\Delta P_i}\Delta P_i \approx \frac{\partial E(k)}{\partial P_i}\Delta P_i, \tag{3.16}$$

for small ΔP_i.

The term $\partial E(k)/\partial P_i$ can be calculated by

$$\frac{\partial E(k)}{\partial P_i} = \frac{\partial E(k)}{\partial o}\frac{\partial o}{\partial P_i} = -\frac{\partial o}{\partial P_i}, \tag{3.17}$$

since $\partial E(k)/\partial o = -1$.

Using ΔP_i from Eq.(3.13), Eq. (3.16) can be rewritten as

$$\Delta E(k) = -\eta_i E(k)\left(\frac{\partial o}{\partial P_i}\right)^2, \tag{3.18}$$

and the change of the Lyapunov function can then be written as

$$\Delta V(k) = -\eta_i\left(E(k)\frac{\partial o}{\partial P_i}\right)^2\left\{1 - \frac{1}{2}\eta_i\left(\frac{\partial o}{\partial P_i}\right)^2\right\}. \tag{3.19}$$

According to the stability condition, $\Delta V(k)$ must be less than zero, this yields

$$0 < \eta_i < 2\left(\frac{\partial o}{\partial P_i}\right)^{-2}. \tag{3.20}$$

The learning rate η_i should lie in the range indicated by the above relation in order to guarantee system stability.

Note that for the direct control system, in this work, the output of FREN must be the control effort of the plant. Unfortunately, the perfect control effort, as the target of FREN, is not known in advance because of the system uncertainty, the nonlinear parameter drift, the disturbance effect and so on. This problem will be discussed and solved in the next chapter.