CHAPTER 6

Multi Input FREN

Since FREN has only one input signal, in some control applications, one must use the pararell structure to handle multiple inputs. The drawback of this parallel structure is that the fuzzy inference cannot be completely emulated. In this chapter, a modification of FREN to handle multiple inputs called MIFREN is presented.

6.1 Structure of MIFREN

For a fuzzy inference system with n inputs and each input has r fuzzy states, thus the number of rules is r^n. Each fuzzy IF-THEN rules can be represented by

RULE k:
IF (I_1 is A_{k_1}) and (I_2 is A_{k_2}) ... and (I_n is A_{k_n})
THEN $O_k = B_k$

where I_j denote the j-th input variable. A_i denotes the i-th fuzzy set, since there are r possible fuzzy sets, then $k_i \in \{1, 2, \ldots, r\}$. The k-th rule is related 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.6.1. 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 $\mu_{A_{i,j}}$ where $i = 1, \ldots, r$; $j = 1, \ldots, n$.

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

$$f_k = f_{k_1,\ldots,k_n} = \prod_{j=1}^{n} \mu_{A_{k_j,j}} \tag{6.1}$$

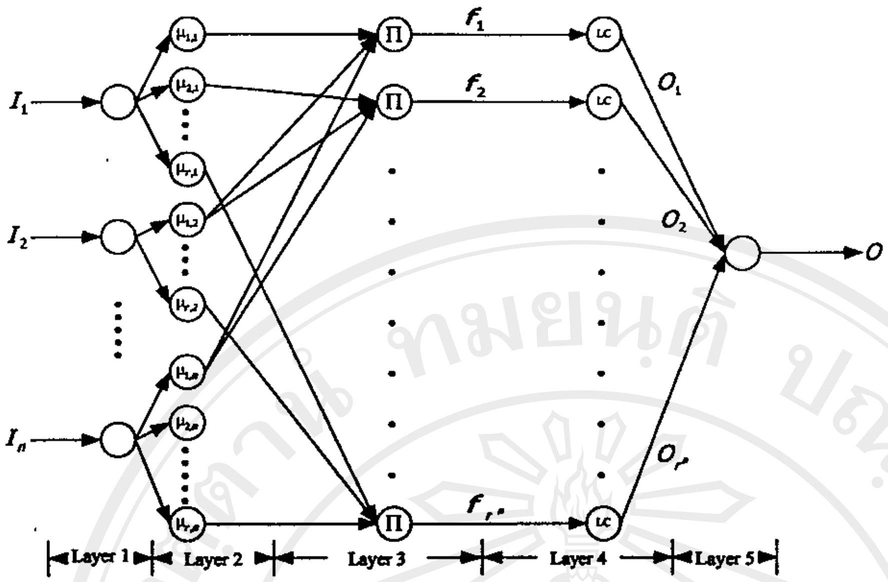


Figure 6.1: Structure of MIFREN

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

$$O_k = \beta_k f_k + b_k, \tag{6.2}$$

where β_k and b_k are parameters of k-th node. For simplicity, in this work, let define $b_k = 0$ for all k. Then Eq.(6.2) becomes

$$O_k = \beta_k f_k. \tag{6.3}$$

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

$$O = f_{\text{MF}}(I_1, \ldots, I_n) = \sum_{k=1}^{r^n} O_k = \boldsymbol{\beta}^T \boldsymbol{F}, \tag{6.4}$$

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

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.

6.1.1 Mamdani's Fuzzy Inference and MIFREN

In this subsection, the comparison between the conventional two inputs Mamdani's fuzzy inference and the output response of MIFREN is given.

6.1.1.1 Mamdani's Fuzzy Inference Surface Response

In [28], the example of Mamdani's fuzzy inference is presented with two inputs and one output. The membership functions of the inputs X and Y and the output Z are shown in Fig.6.2(a)-(c), respectively. And the four fuzzy rules are expressed as

RULE 1	IF X IS SMALL	AND Y IS SMALL	THEN Z IS NEGATIVE LARGE,
RULE 2	IF X IS SMALL	AND Y IS LARGE	THEN Z IS NEGATIVE SMALL,
RULE 3	IF X IS LARGE	AND Y IS SMALL	THEN Z IS POSITIVE SMALL,
RULE 4	IF X IS LARGE	AND Y IS LARGE	THEN Z IS POSITIVE LARGE.

With the max-min composition and the centroid defuzzification, the overall input-output surface response is obtained as shown in Fig.6.3.

6.1.1.2 MIFREN Surface Response

The fuzzy rules for two inputs MIFREN is similar to the rules employed in the case of Mamdani's. They are given by,

RULE 1	IF X IS SMALL	AND Y IS SMALL	THEN Z $= \beta_1 f_1$,
RULE 2	IF X IS SMALL	AND Y IS LARGE	THEN Z $= \beta_2 f_2$,
RULE 3	IF X IS LARGE	AND Y IS SMALL	THEN Z $= \beta_3 f_3$,
RULE 4	IF X IS LARGE	AND Y IS LARGE	THEN Z $= \beta_4 f_4$.

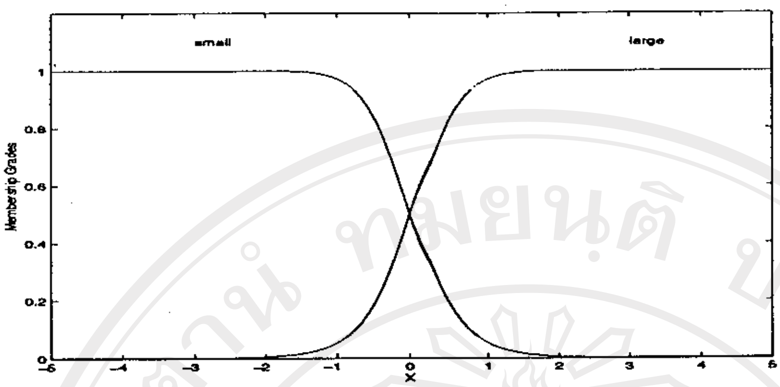
The initial value of β_i and f_i $(i = 1, 2, 3, 4)$ are defined as

$\beta_1 = -5$	$f_1 = \mu_{\text{small}X}(X)\mu_{\text{small}Y}(Y)$,
$\beta_2 = -1.75$	$f_2 = \mu_{\text{small}X}(X)\mu_{\text{large}Y}(Y)$,
$\beta_3 = 1.75$	$f_3 = \mu_{\text{large}X}(X)\mu_{\text{small}Y}(Y)$,
$\beta_4 = 5$	$f_4 = \mu_{\text{large}X}(X)\mu_{\text{large}Y}(Y)$,

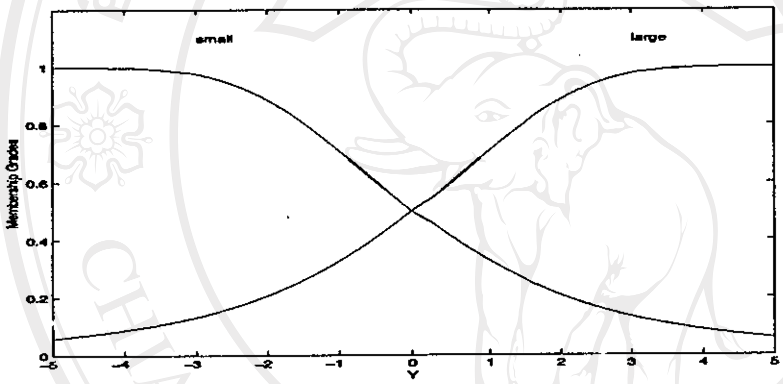
where $\mu_{\Box_X}$ and $\mu_{\Box_Y}$ are the corresponding membership functions of X and Y, respectively.

To define these linear parameters β_i, consider the output Z from Fig. 6.3, it has the maximum and the minimum value around 5 and -5, respectively. That means a linear parameter of Rule 1 (β_1) is set to -5 and a linear parameter of Rule 4 (β_4) is set to 5. β_2 can be set to any value between -5/2 and 0. And for β_3, it can be set to any value between 0 and 5/2. However, in Fig. 6.3, it can be seen that the gradient nearly zero is very high. It may be better when β_2 and β_3 are set nearly zero than -2.5 and 2.5. In this selection, -1.75 and 1.75 are selected for β_2 and β_3, respectively. These functions are illustrated in Fig. 6.4 (a) and (b) for X and Y, respectively. Finally, the overall input-output surface response is obtained as shown in Fig. 6.4.

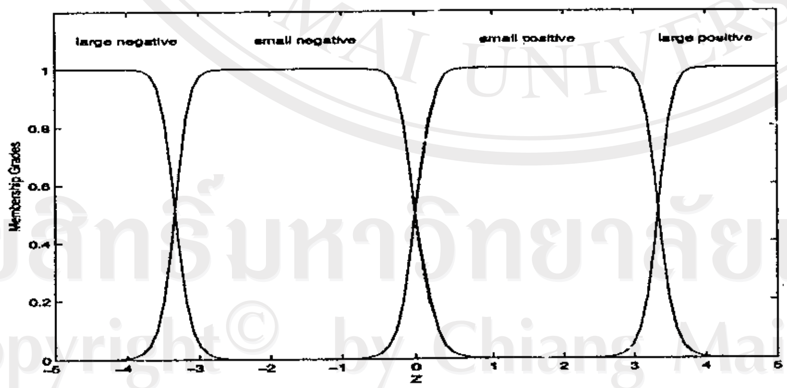
In this example, the surface response shown in Fig.6.3 and Fig.6.4 are similar which implies that the fuzzy behavior of the conventional Mamdani's fuzzy inference can also be obtained from the MIFREN. However, since the defuzzification step in MIFREN is included in the linear consequence layer, the initial parameter setting becomes easier and the computation complexity is reduced in MIFREN approach. Comparing to the well known TSK fuzzy system [60] which uses a linear function in the consequence part, but the number of parameters depends on the number of input signals, MIFREN needs only β_k and b_k in Eq.(6.2) for the consequence part of the k-th rule.



(a)



(b)



(c)

Figure 6.2: Membership function of (a) **X**, (b) Y and (c) Z for the two inputs Mamdani fuzzy inference system.

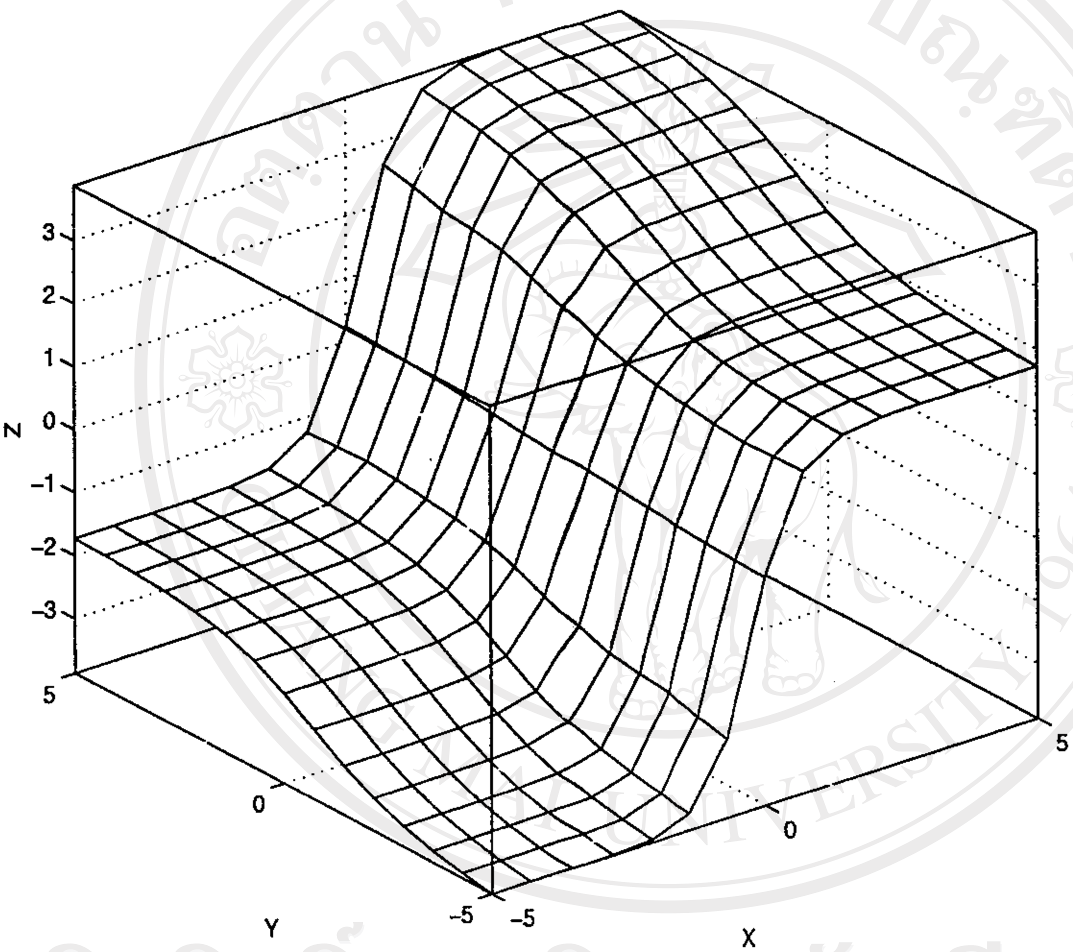
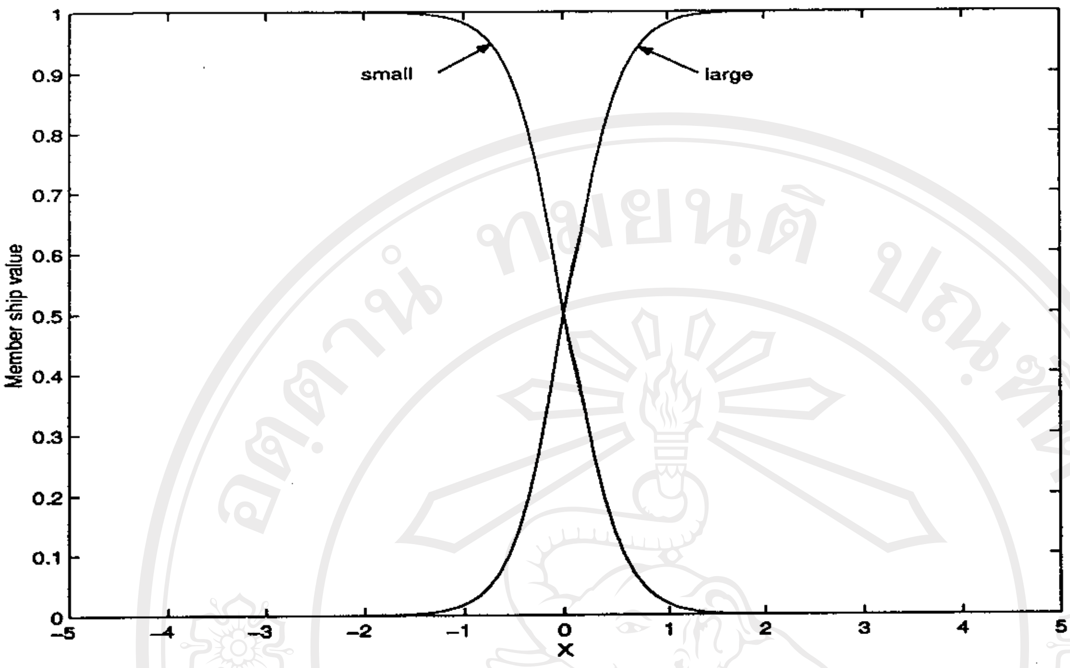
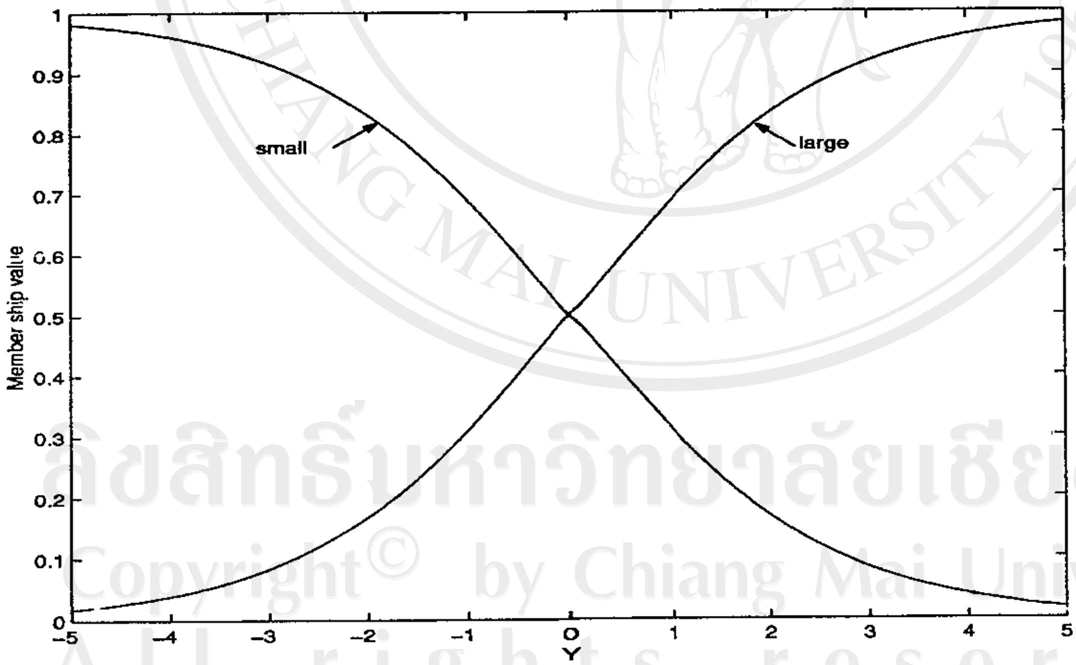


Figure 6.3: Output-input surface response obtained from Mamdani fuzzy system.



(a)



(b)

Figure 6.4: Membership functions of (a) X and (b) Y for MiFren.

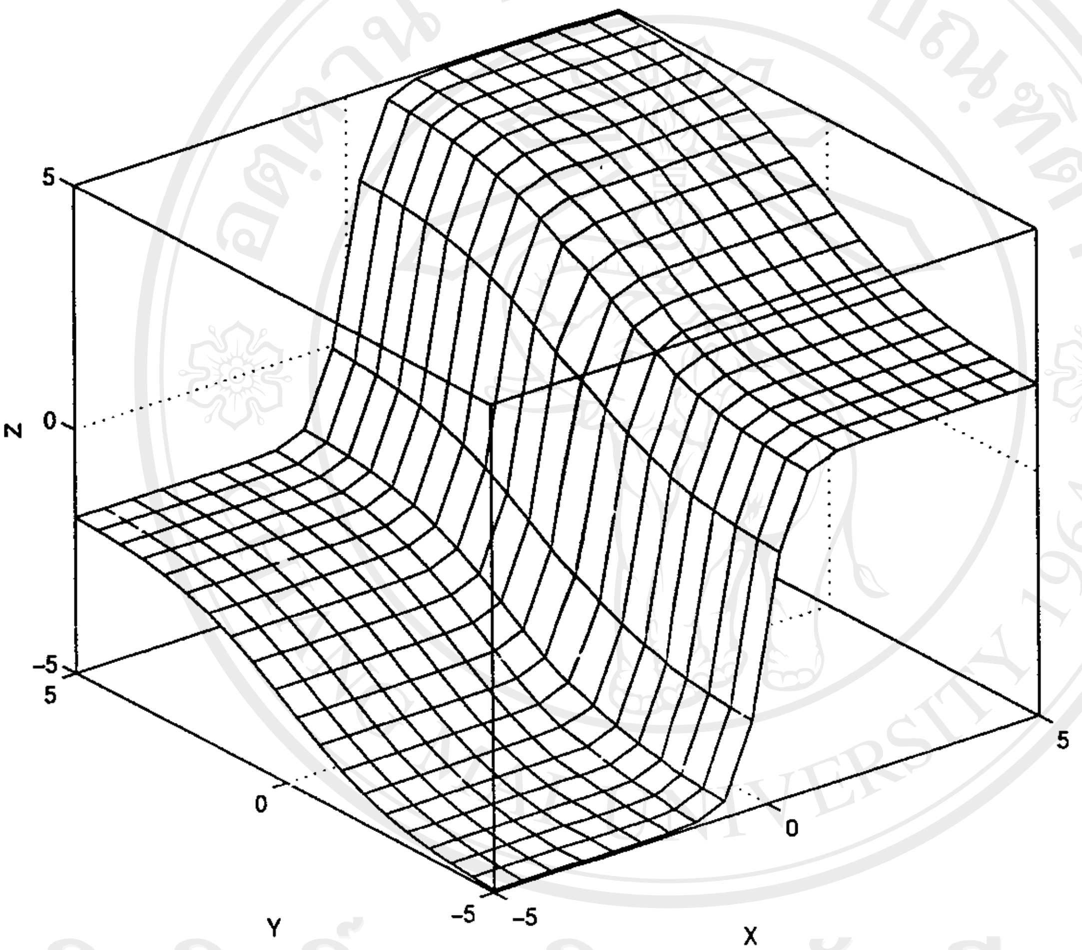


Figure 6.5: Surface response of MiFren.

6.1.2 MIFREN as Universal Function Approximation

In this subsection, it will be shown that MIFREN has the property of a universal function approximation using the Stone-Weierstrass theorem [4,65].

Theorem 6.1.1 ***Stone-Weierstrass Theorem***
"Let Ω be a compact space of N dimensions and let $\mathcal{F}$ be a set of real functions on a compact set Ω. If

(1) $\mathcal{F}$ is an algebra,

(2) $\mathcal{F}$ separates points on Ω, and

(3) $\mathcal{F}$ vanishes at no point on Ω,

then $\mathcal{F}$ is dense in $C(\Omega)$, the set of continuous real-valued function on Ω. In other words, for any $\hat{e} > 0$ and any function g in $C(\Omega)$, there is a function f in $\mathcal{F}$ such that $|g(x) - f(x)| < \hat{e}$ for all $x \in \Omega$".

Let the output of MIFREN $y(\boldsymbol{x}) \in \mathcal{F}$ and the input $\boldsymbol{x} \in \Omega$. From Eq.(6.4), the output of two MIFREN networks can be rewritten as

$$\begin{aligned} y_1(\boldsymbol{x}) &= \sum_{k=1}^{N_1} \beta_{1,k} \prod_{j=1}^{n} \mu_{1_{k_j,j}}(x_j) \\ &= \sum_{k=1}^{N_1} \beta_{1,k} f_{1,k} \\ &= \boldsymbol{\beta}_1^T \boldsymbol{F}_1, \end{aligned}$$

and

$$\begin{aligned} y_2(\boldsymbol{x}) &= \sum_{k=1}^{N_2} \beta_{2,k} \prod_{j=1}^{n} \mu_{2_{k_j,j}}(x_j) \\ &= \sum_{k=1}^{N_2} \beta_{2,k} f_{2,k} \\ &= \boldsymbol{\beta}_2^T \boldsymbol{F}_2. \end{aligned}$$

Consider,

$$\begin{aligned} g(\boldsymbol{x}) &= y_1(\boldsymbol{x}) + y_2(\boldsymbol{x}) \\ &= \boldsymbol{\beta}_1^T \boldsymbol{F}_1 + \boldsymbol{\beta}_2^T \boldsymbol{F}_2 \\ &= \begin{bmatrix} \boldsymbol{\beta}_1^T & \boldsymbol{\beta}_2^T \end{bmatrix} \begin{bmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \end{bmatrix} \\ &= \bar{\boldsymbol{\beta}}^T \bar{\boldsymbol{F}}, \end{aligned}$$

where $\bar{\boldsymbol{\beta}}^T = [\boldsymbol{\beta}_1^T \quad \boldsymbol{\beta}_2^T]$ and $\bar{\boldsymbol{F}}^T = [\boldsymbol{F}_1^T \quad \boldsymbol{F}_2^T]$ Since $\beta_{i,j} \in \mathbb{R}$, then $g(\boldsymbol{x}) = y_1(\boldsymbol{x}) + y_2(\boldsymbol{x}) \in \mathcal{F}$.
Next, consider

$$\begin{aligned} g(\boldsymbol{x}) &= y_1(\boldsymbol{x})y_2(\boldsymbol{x}), \\ &= y_1(\boldsymbol{x})\beta_2^T \boldsymbol{F}_2, \\ &= \grave{\beta}_2^T \boldsymbol{F}_2, \end{aligned}$$

where $\grave{\boldsymbol{\beta}}_2^T = y_1(\boldsymbol{x})\boldsymbol{\beta}_2^T$ and each value of $\grave{\beta}_2$ or $\grave{\beta}_{2,i} \in \mathbb{R}$. Thus $g(\boldsymbol{x}) = y_1(\boldsymbol{x})y_2(\boldsymbol{x}) \in \mathcal{F}$.
For $a \in \mathbb{R}$

$$\begin{aligned} g(\boldsymbol{x}) &= a y_1(\boldsymbol{x}), \\ &= a\boldsymbol{\beta}_1^T \boldsymbol{F}_1, \\ &= \hat{\boldsymbol{\beta}}_1^T \boldsymbol{F}_1, \end{aligned}$$

where $\hat{\boldsymbol{\beta}}_1^T = a\boldsymbol{\beta}_1^T$. That is $g(\boldsymbol{x}) = a y_1(\boldsymbol{x}) \in \mathcal{F}$.
From these results, it can be concluded that $\mathcal{F}$ is an *algebra*.

Next it will be shown that for any two points $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$ in Ω there is y in $\mathcal{F}$ such that $y(\boldsymbol{x}_1) \neq y(\boldsymbol{x}_2)$.

Let $\boldsymbol{x}_1 = [x_{1,1}^o \quad x_{1,2}^o \quad \cdots \quad x_{1,n}^o]^T$, $\boldsymbol{x}_2 = [x_{2,1}^o \quad x_{2,2}^o \quad \cdots \quad x_{2,n}^o]^T$ and $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$.

Define two fuzzy rules of MiFren as

Rule 1 If I_1 is A_{11} and I_2 is A_{12} $\cdots$ and I_n is A_{1n} Then $y_1 = \beta_1 f_1$,
Rule 2 If I_1 is A_{21} and I_2 is A_{22} $\cdots$ and I_n is A_{2n} Then $y_2 = \beta_2 f_2$.

The output of MiFren is

$$y(\boldsymbol{I}) = \beta_1 f_1 + \beta_2 f_2, \tag{6.5}$$

where $\boldsymbol{I} \in \mathbb{R}^n$ denotes the input vector, and define

$$\begin{aligned} f_1 &= \prod_{i=1}^{n} \mu_{A_{1,i}}(I_i) = \prod_{i=1}^{n} \psi_{A_{1,i}}(I_i - x_{1,i}^o), \\ f_2 &= \prod_{i=1}^{n} \mu_{A_{2,i}}(I_i) = \prod_{i=1}^{n} \psi_{A_{2,i}}(I_i - x_{2,i}^o), \end{aligned}$$

here $\psi_{A_{k,i}}$ represents the function used to implement the membership, i.e. $\mu_{A_{k,i}}(x) = \psi_{A_{k,i}}(x - x^o_{k,i})$, $x^o_{1,i}$ and $x^o_{2,i}$ are the center of $\mu_{A_{1,i}}$ and $\mu_{A_{2,i}}$, respectively.

Thus

$$\begin{aligned} y(\boldsymbol{x}_1) &= \beta_1 \prod_{i=1}^{n} \psi_{A_{1,i}}(x^o_{1,i} - x^o_{1,i}) + \beta_2 \prod_{i=1}^{n} \psi_{A_{2,i}}(x^o_{1,i} - x^o_{2,i}) \\ &= \beta_1 \prod_{i=1}^{n} \psi_{A_{1,i}}(0) + \beta_2 \prod_{i=1}^{n} \psi_{A_{2,i}}(x^o_{1,i} - x^o_{2,i}), \end{aligned} \tag{6.6}$$

and

$$\begin{aligned} y(\boldsymbol{x}_2) &= \beta_1 \prod_{i=1}^{n} \psi_{A_{1,i}}(x^o_{2,i} - x^o_{1,i}) + \beta_2 \prod_{i=1}^{n} \psi_{A_{2,i}}(x^o_{2,i} - x^o_{2,i}) \\ &= \beta_1 \prod_{i=1}^{n} \psi_{A_{1,i}}(x^o_{2,i} - x^o_{1,i}) + \beta_2 \prod_{i=1}^{n} \psi_{A_{2,i}}(0) \end{aligned} \tag{6.7}$$

From the characteristic of the membership function one has

$$\prod_{i=1}^{n} \psi_{A_{1,i}}(0) \neq \prod_{i=1}^{n} \psi_{A_{1,i}}(x^o_{2,i} - x^o_{1,i}), \tag{6.8}$$

and

$$\prod_{i=1}^{n} \psi_{A_{2,i}}(0) \neq \prod_{i=1}^{n} \psi_{A_{2,i}}(x^o_{1,i} - x^o_{2,i}). \tag{6.9}$$

Thus if β_1 and β_2 are selected such that

$$\frac{\beta_1}{\beta_2} \neq \frac{\prod_{i=1}^{n} \psi_{A_{2,i}}(0) - \prod_{i=1}^{n} \psi_{A_{2,i}}(x^o_{1,i} - x^o_{2,i})}{\prod_{i=1}^{n} \psi_{A_{1,i}}(0) - \prod_{i=1}^{n} \psi_{A_{1,i}}(x^o_{2,i} - x^o_{1,i})} \tag{6.10}$$

then it can be shown that $y(\boldsymbol{x}_1) \neq y(\boldsymbol{x}_2)$. It can be concluded that $\mathcal{F}$ processes *separability.*

From Eq.(6.4), $y(\boldsymbol{x}) = \sum_{k=1}^{r^n} \beta_k f_k$ and $\forall \boldsymbol{x} \in \mathbb{R}^n$. The membership functions are also defined such that $f_k > 0$. If we also choose $\beta_k > 0$ then $y > 0$ for any $\boldsymbol{x} \in \mathbb{R}^n$. That is, any $y \in \mathcal{F}$ with $\beta_k > 0$ can serve as the required y. It can be concluded that $\mathcal{F}$ *vanishes at no point.*

Therefore, the muti-input FREN or MIFREN is a universal function approximator from the Stone-Weierstrass theorem.

6.2 MIFREN with SMC

In this section, the MIFREN is applied as a direct adaptive controller. As in the case of FREN the sliding bounds proposed in Chapter 5 is employed to the control effort generated by MIFREN.

6.2.1 Controller Design

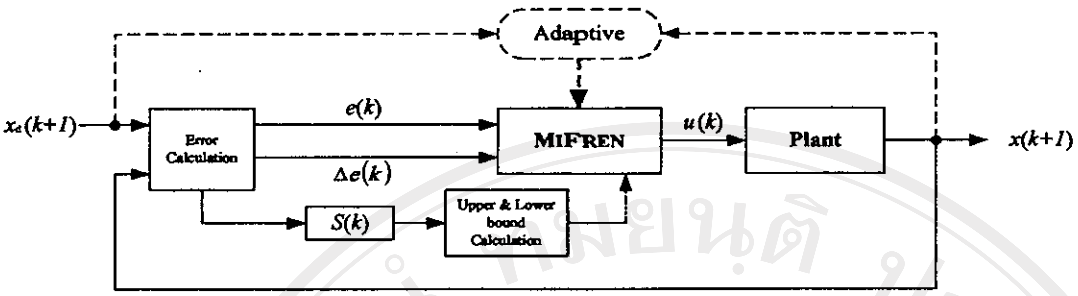


Figure 6.6: Control system using MIFREN

To improve the proposed control algorithm in 5.2, here the two-input MIFREN is used as a controller. The structure of the control system is illustrated in Fig. 6.6. Here, MIFREN receives the error signal $e(k)$ and its change $\Delta e(k)$ and computes the control signal $u(k)$,

$$u(k) = f_{\text{MF}}\left(e(k), \Delta e(k)\right), \tag{6.11}$$

where $f_{\text{MF}}(.)$ denotes the MIFREN's operation in Eq.(6.4).

As an example of how the initial value of FREN's parameters are selected, consider the following fuzzy control rules,

RULE 1	IF $e(k)$ IS N	AND $\Delta e(k)$ IS N	THEN $u_1(k) = \beta_1 f_1(k)$,
RULE 2	IF $e(k)$ IS N	AND $\Delta e(k)$ IS P	THEN $u_2(k) = \beta_2 f_2(k)$,
RULE 3	IF $e(k)$ IS P	AND $\Delta e(k)$ IS N	THEN $u_3(k) = \beta_3 f_3(k)$,
RULE 4	IF $e(k)$ IS P	AND $\Delta e(k)$ IS P	THEN $u_4(k) = \beta_4 f_4(k)$,

here N and P denote the negative and the positive linguistic level respectively. Assume that both e and $\Delta e \in [-1, 1]$, and the minimum and the maximum values of control effort are -1 and 1, respectively, i.e. $u(k) \in [-1, 1]$. The initial value of β_i and f_i $(i = 1, 2, 3, 4)$ are defined as

$\beta_1 = -1$	$f_1 = \mu_{\text{N}e}(e_k)\mu_{\text{N}\Delta e}(\Delta e_k)$,
$\beta_2 = -0.5$	$f_2 = \mu_{\text{N}e}(e_k)\mu_{\text{P}\Delta e}(\Delta e_k)$,
$\beta_3 = 0.5$	$f_3 = \mu_{\text{P}e}(e_k)\mu_{\text{N}\Delta e}(\Delta e_k)$,
$\beta_4 = 1$	$f_4 = \mu_{\text{P}e}(e_k)\mu_{\text{P}\Delta e}(\Delta e_k)$,

Then, MF parameters are selected to cover the error range. The initial setting of all parameters are given as:

$$\begin{aligned} \mu_{\text{N}e}(x) &= \mu_{\text{N}\Delta e}(x) &= \left(1 + \exp\left[20(x + 0.5)\right]\right)^{-1}, \\ \mu_{\text{P}e}(x) &= \mu_{\text{P}\Delta e}(x) &= \left(1 + \exp\left[-20(x - 0.5)\right]\right)^{-1}. \end{aligned}$$

The adaptation of all MF and LC parameters are performed by using the algorithm in subsection 3.3.

6.2.2 Computer Simulation Results

6.2.2.1 Robotic Control

The characteristic of the nonlinear discrete-time robotic plant [50] is given as

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ f(\boldsymbol{x}(k)) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k), \tag{6.12}$$

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

$$f(\boldsymbol{x}(k)) = (2T-1)x_1(k) + 2(1-T)x_2(k) + 10T^2 \sin(x_1(k)).$$

where T is the sampling interval and equals to 0.01 second. The reference signal $\boldsymbol{x}_d$ is set as

$$\boldsymbol{x}_d(k+1) = \begin{bmatrix} x_{d1}(k+1) \\ x_{d2}(k+1) \end{bmatrix} = \frac{\pi}{2} \begin{bmatrix} \sin(\frac{\pi}{5}kT) \\ \sin(\frac{\pi}{5}(k+1)T) \end{bmatrix}. \tag{6.13}$$

Denote the error $e(k) = x_2(k) - x_{d2}(k)$ and $\boldsymbol{c}$ in Eq.(5.3) is defined as $\boldsymbol{c} = \begin{bmatrix} 1 & 2 \end{bmatrix}$. The fuzzy control rules are given by,

RULE 1	IF $e(k)$ IS N	AND $\Delta e(k)$ IS N	THEN $u_1(k) = \beta_1 f_1$,
RULE 2	IF $e(k)$ IS N	AND $\Delta e(k)$ IS Z	THEN $u_2(k) = \beta_2 f_2$,
RULE 3	IF $e(k)$ IS N	AND $\Delta e(k)$ IS P	THEN $u_3(k) = \beta_3 f_3$,
RULE 4	IF $e(k)$ IS Z	AND $\Delta e(k)$ IS N	THEN $u_4(k) = \beta_4 f_4$,
RULE 5	IF $e(k)$ IS Z	AND $\Delta e(k)$ IS Z	THEN $u_5(k) = \beta_5 f_5$,
RULE 6	IF $e(k)$ IS Z	AND $\Delta e(k)$ IS P	THEN $u_6(k) = \beta_6 f_6$,
RULE 7	IF $e(k)$ IS P	AND $\Delta e(k)$ IS N	THEN $u_7(k) = \beta_7 f_7$,
RULE 8	IF $e(k)$ IS P	AND $\Delta e(k)$ IS Z	THEN $u_8(k) = \beta_8 f_8$,
RULE 9	IF $e(k)$ IS P	AND $\Delta e(k)$ IS P	THEN $u_9(k) = \beta_9 f_9$,

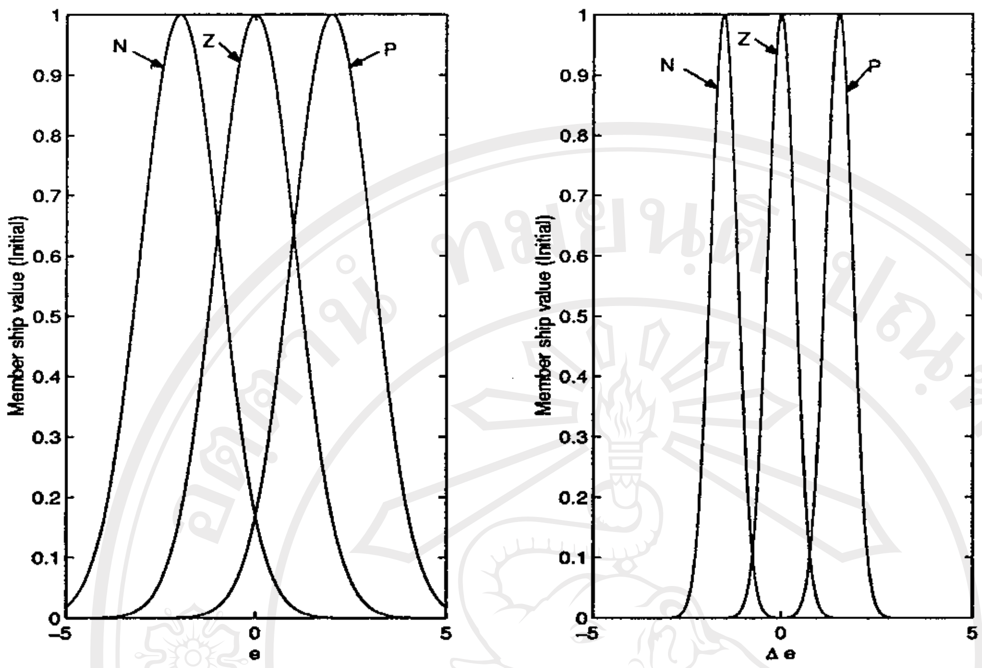
where N, Z and P denote the negative, the zero and the positive linguistic levels respectively. The initial values of β_i and f_i $(i = 1, 2, \cdots, 9)$ are defined as

$\beta_1 = 3.0$	$f_1 = \mu_{\mathrm{N}e}(e_k)\mu_{\mathrm{N}\Delta e}(\Delta e_k)$,
$\beta_2 = 2.1$	$f_2 = \mu_{\mathrm{N}e}(e_k)\mu_{\mathrm{Z}\Delta e}(\Delta e_k)$,
$\beta_3 = 1.5$	$f_3 = \mu_{\mathrm{N}e}(e_k)\mu_{\mathrm{P}\Delta e}(\Delta e_k)$,
$\beta_4 = 0.6$	$f_4 = \mu_{\mathrm{Z}e}(e_k)\mu_{\mathrm{N}\Delta e}(\Delta e_k)$,
$\beta_5 = 0.0$	$f_5 = \mu_{\mathrm{Z}e}(e_k)\mu_{\mathrm{Z}\Delta e}(\Delta e_k)$,
$\beta_6 = -0.6$	$f_6 = \mu_{\mathrm{Z}e}(e_k)\mu_{\mathrm{P}\Delta e}(\Delta e_k)$,
$\beta_7 = -1.5$	$f_7 = \mu_{\mathrm{P}e}(e_k)\mu_{\mathrm{N}\Delta e}(\Delta e_k)$,
$\beta_8 = -2.1$	$f_8 = \mu_{\mathrm{P}e}(e_k)\mu_{\mathrm{Z}\Delta e}(\Delta e_k)$,
$\beta_9 = -3.0$	$f_9 = \mu_{\mathrm{P}e}(e_k)\mu_{\mathrm{P}\Delta e}(\Delta e_k)$,

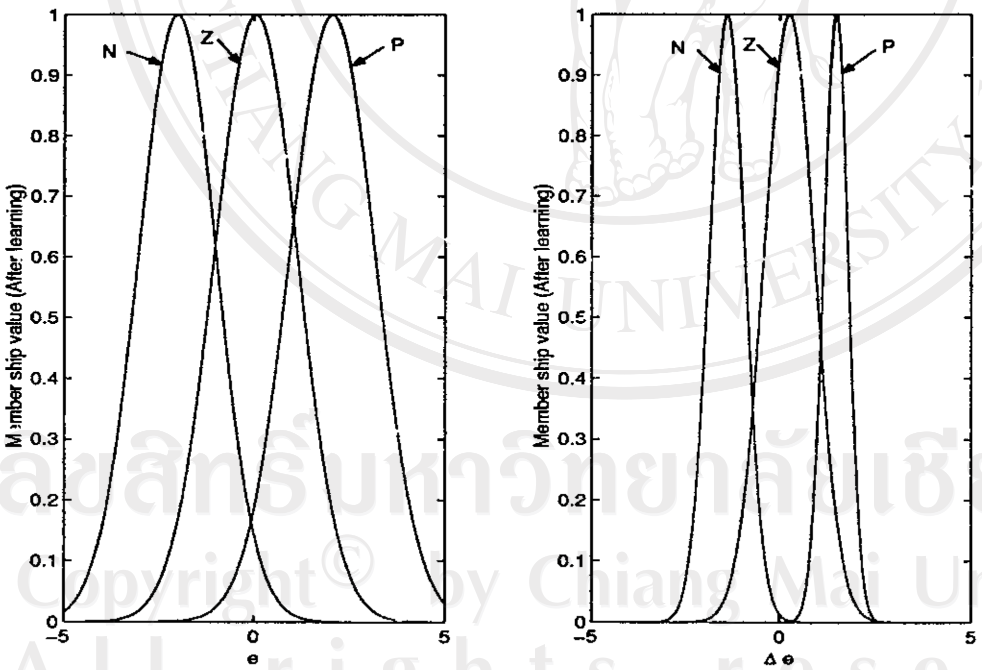
The MF parameters are selected as shown in Fig. 6.7 (a). The parameter learning process is finished at $k = 2,000$ and the final MF parameters are shown in Fig. 6.7 (b). The linear parameters β_i after learning process become

$\beta_1 = 3.0309$	$\beta_2 = 2.2080$	$\beta_3 = 1.4933,$
$\beta_4 = 0.7146$	$\beta_5 = 0.0712$	$\beta_6 = -0.6653,$
$\beta_7 = -1.4870$	$\beta_8 = -2.1605$	$\beta_9 = -3.0179.$

The waveforms of the output signal $x_2(k)$, the control effort $u(k)$ and the error $e(k)$ are shown in Fig. 6.8 (a)-(c), respectively.

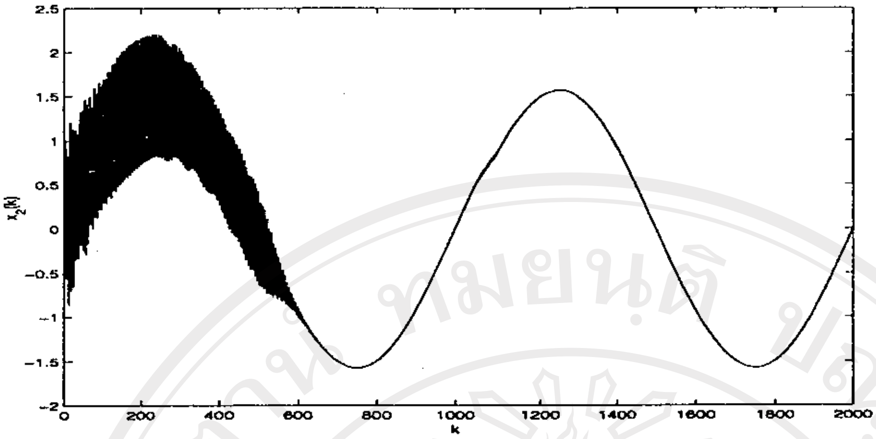


(a) Initial setting ($k = 1.$)

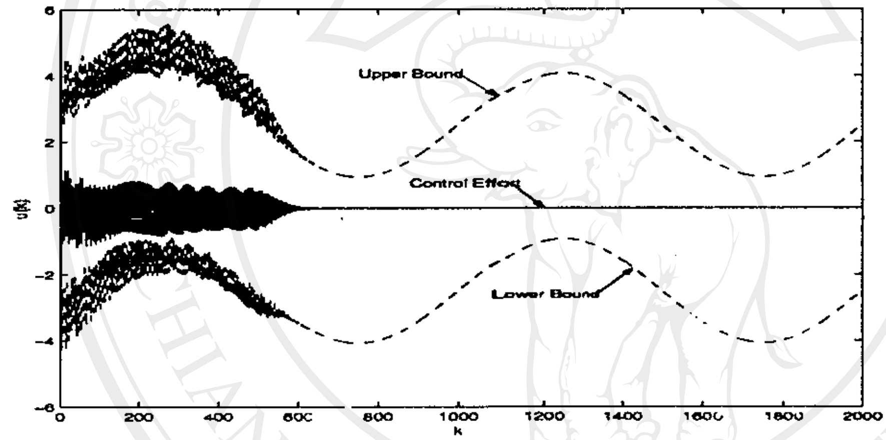


(b) Final setting ($k = 2000.$)

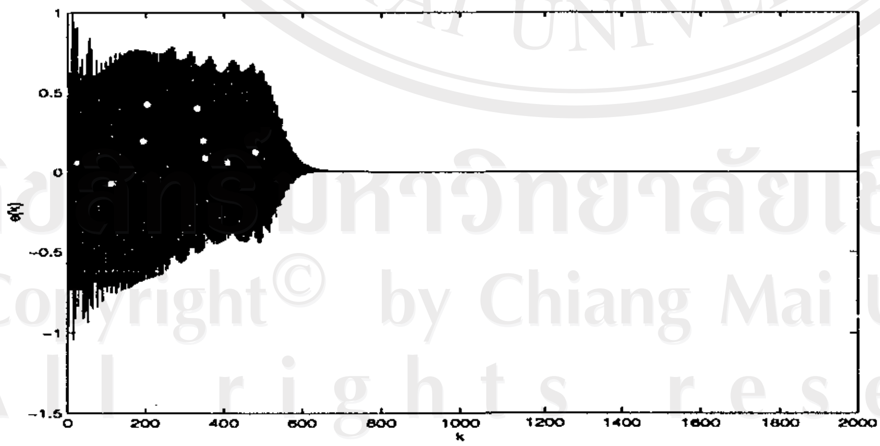
Figure 6.7: MF of MiFREN for robotic control.



(a) Output $x_2(k)$.



(b) Control effort $u(k)$ and its bounds.



(c) Error $e(k)$.

Figure 6.8: Simulation results of robotic control using MiFren.

6.2.2.2 Hénon Map Control

The discrete-time Hénon map from subsection 5.3.2 is tested with MIFREN controller to verify its performances. From Eq.(5.35), the characteristic of the controlled Hénon map is given by,

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ f(\boldsymbol{x}(k)) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k), \tag{6.14}$$

where $u(k)$ is the control signal and

$$f(\boldsymbol{x}(k)) = A - x_2^2(k) + Bx_1(k), \tag{6.15}$$

when $A = 1.29$ and $B = 0.3$. Let $\boldsymbol{c} = \begin{bmatrix} 1 & 2 \end{bmatrix}$ for the sliding bounds setting parameter in Eq.(5.1). The fixed point y_F is set again to 0.838486, then $\boldsymbol{x}_d$ can be written as

$$\boldsymbol{x}_d(k+1) = \begin{bmatrix} x_{d1}(k+1) \\ x_{d2}(k+1) \end{bmatrix} = \begin{bmatrix} y_F \\ y_F \end{bmatrix}. \tag{6.16}$$

Denoting the error $e = x_2(k) - y_F$, the fuzzy control rules are given by,

RULE 1 IF $e(k)$ IS N AND $\Delta e(k)$ IS N THEN $u_1(k) = \beta_1 f_1(k)$,
RULE 2 IF $e(k)$ IS N AND $\Delta e(k)$ IS Z THEN $u_2(k) = \beta_2 f_2(k)$,
RULE 3 IF $e(k)$ IS N AND $\Delta e(k)$ IS P THEN $u_3(k) = \beta_3 f_3(k)$,
RULE 4 IF $e(k)$ IS Z AND $\Delta e(k)$ IS N THEN $u_4(k) = \beta_4 f_4(k)$,
RULE 5 IF $e(k)$ IS Z AND $\Delta e(k)$ IS Z THEN $u_5(k) = \beta_5 f_5(k)$,
RULE 6 IF $e(k)$ IS Z AND $\Delta e(k)$ IS P THEN $u_6(k) = \beta_6 f_6(k)$,
RULE 7 IF $e(k)$ IS P AND $\Delta e(k)$ IS N THEN $u_7(k) = \beta_7 f_7(k)$,
RULE 8 IF $e(k)$ IS P AND $\Delta e(k)$ IS Z THEN $u_8(k) = \beta_8 f_8(k)$,
RULE 9 IF $e(k)$ IS P AND $\Delta e(k)$ IS P THEN $u_9(k) = \beta_9 f_9(k)$,

where N, Z and P denote the negative, the zero and the positive linguistic level respectively. The initial value of β_i and f_i $(i = 1, 2, \cdots, 9)$ are defined as

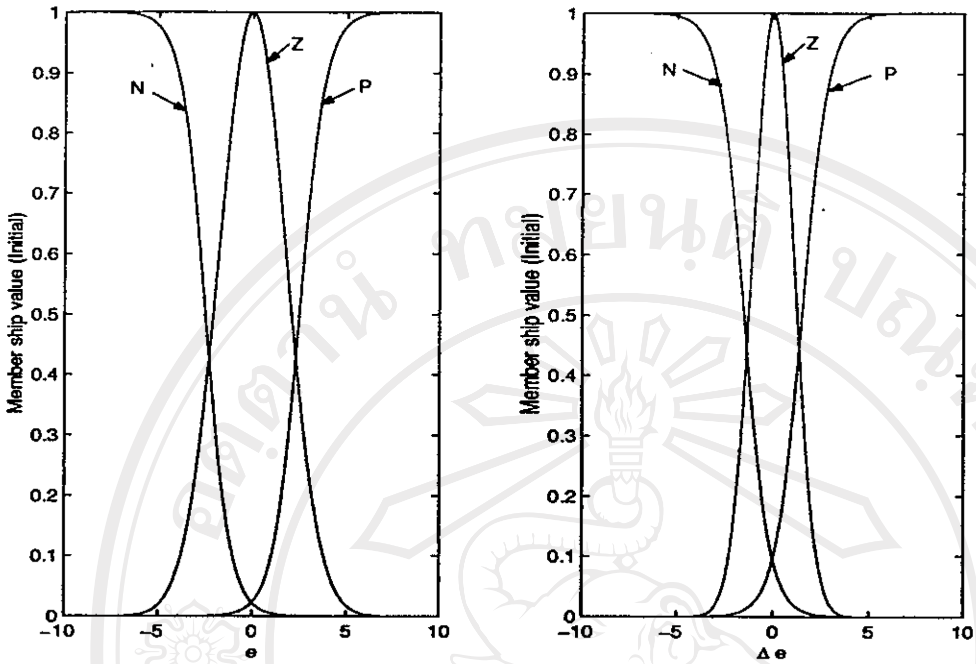
$\beta_1 = -1.5$	$f_1 = \mu_{\mathrm{N}e}(e_k)\mu_{\mathrm{N}\Delta e}(\Delta e_k)$,
$\beta_2 = -1.0$	$f_2 = \mu_{\mathrm{N}e}(e_k)\mu_{\mathrm{Z}\Delta e}(\Delta e_k)$,
$\beta_3 = -0.5$	$f_3 = \mu_{\mathrm{N}e}(e_k)\mu_{\mathrm{P}\Delta e}(\Delta e_k)$,
$\beta_4 = -0.2$	$f_4 = \mu_{\mathrm{Z}e}(e_k)\mu_{\mathrm{N}\Delta e}(\Delta e_k)$,
$\beta_5 = 0.0$	$f_5 = \mu_{\mathrm{Z}e}(e_k)\mu_{\mathrm{Z}\Delta e}(\Delta e_k)$,
$\beta_6 = 0.2$	$f_6 = \mu_{\mathrm{Z}e}(e_k)\mu_{\mathrm{P}\Delta e}(\Delta e_k)$,
$\beta_7 = 0.5$	$f_7 = \mu_{\mathrm{P}e}(e_k)\mu_{\mathrm{N}\Delta e}(\Delta e_k)$,
$\beta_8 = 1.0$	$f_8 = \mu_{\mathrm{P}e}(e_k)\mu_{\mathrm{Z}\Delta e}(\Delta e_k)$,
$\beta_9 = 1.5$	$f_9 = \mu_{\mathrm{P}e}(e_k)\mu_{\mathrm{P}\Delta e}(\Delta e_k)$,

The MF parameters are selected as shown in Fig. 6.9 (a). The parameter learning process is finished at $k = 150$ and the final MF parameters are shown in Fig. 6.9 (b). The LC parameters β_i after learning process become,

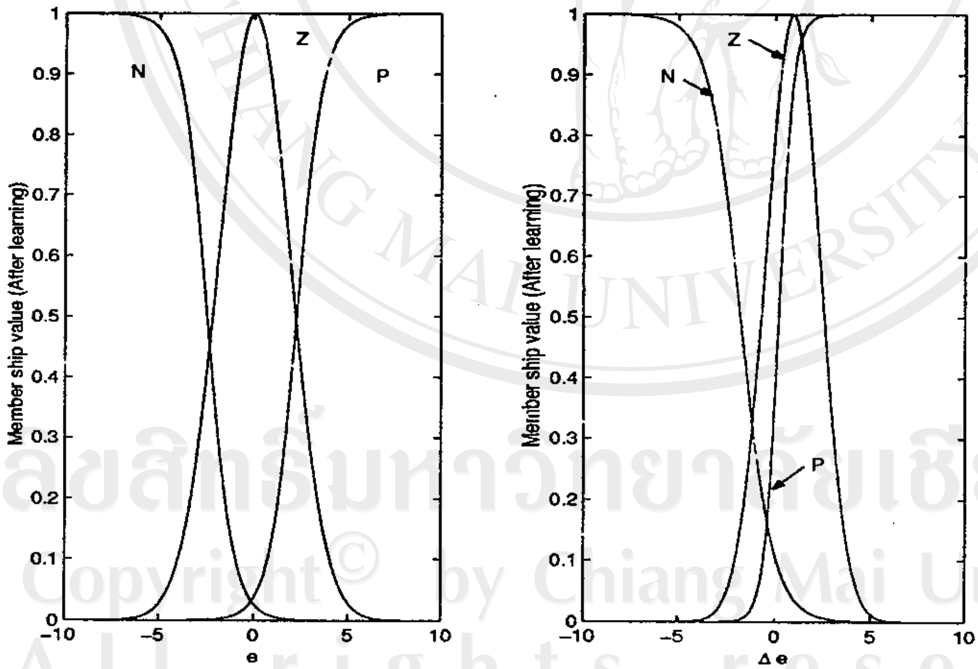
$\beta_1 = -1.5769$	$\beta_2 = -1.0791$	$\beta_3 = -0.4926,$
$\beta_4 = -1.3966$	$\beta_5 = -0.6083$	$\beta_6 = 1.7764,$
$\beta_7 = 0.4778$	$\beta_8 = 1.0808$	$\beta_9 = 1.6655.$

The output signal $x_2(k)$, the control effort $u(k)$ and the error $e(k)$ are shown in Fig. 6.10 (a)-(c), respectively. These signals are illustrated again in Fig. 6.11 when $\boldsymbol{c} = \begin{bmatrix} 1 & 4 \end{bmatrix}$.

In these simulation tests, the convergent speed of the proposed controller is improved from FREN in Chapter 5. By using MIFREN as the controller, good results are obtained in the first testing epoch or process. While the control system based on FREN and sliding bounds gives good results after 2,000 testing epochs.

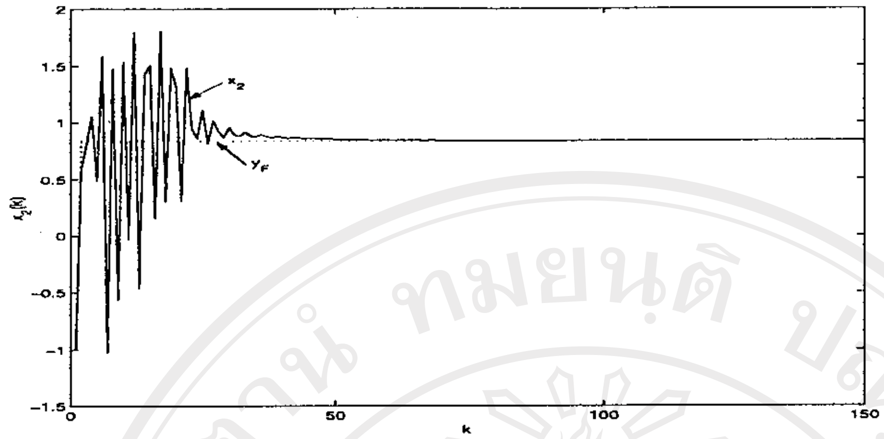


(a) Initial setting ($k = 1.$)

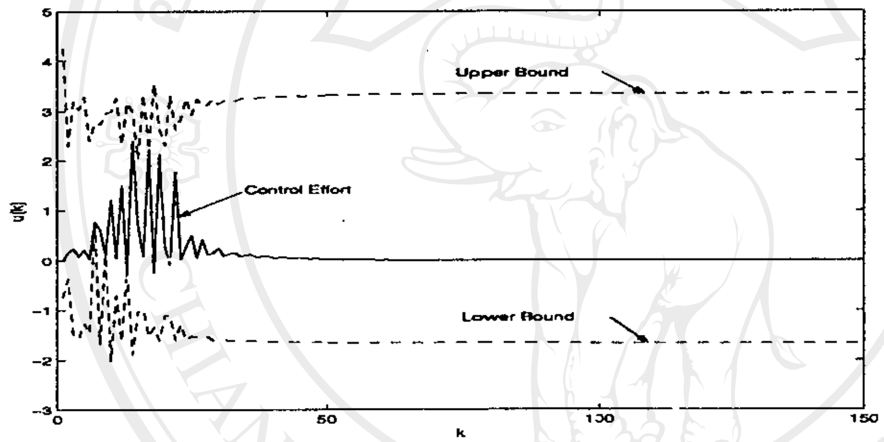


(b) Final setting ($k = 150.$)

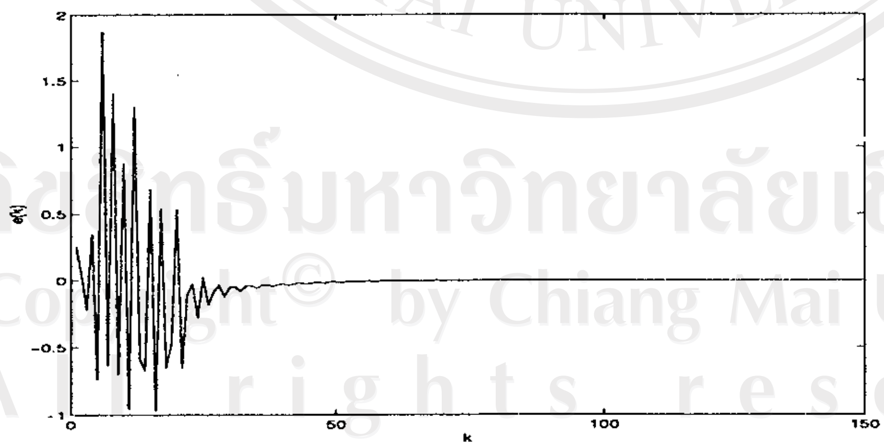
Figure 6.9: MF's parameter of Hénon map control



(a) Output $x_2(k)$.

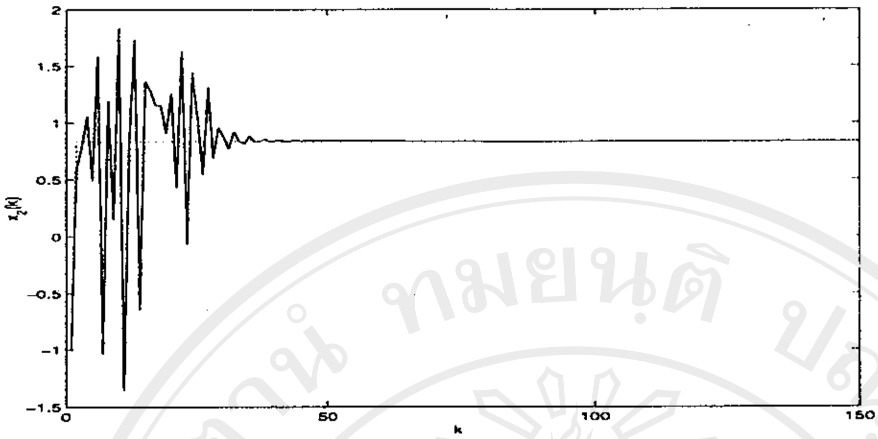


(b) Control effort $u(k)$ and its bounds.

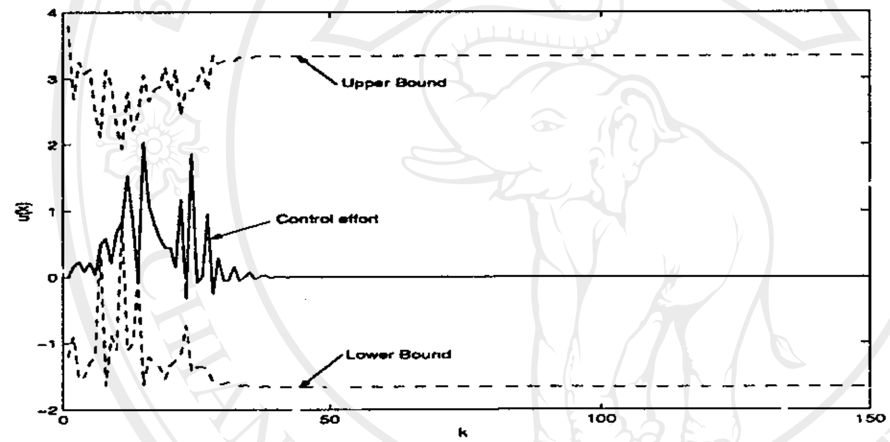


(c) Error $e(k)$.

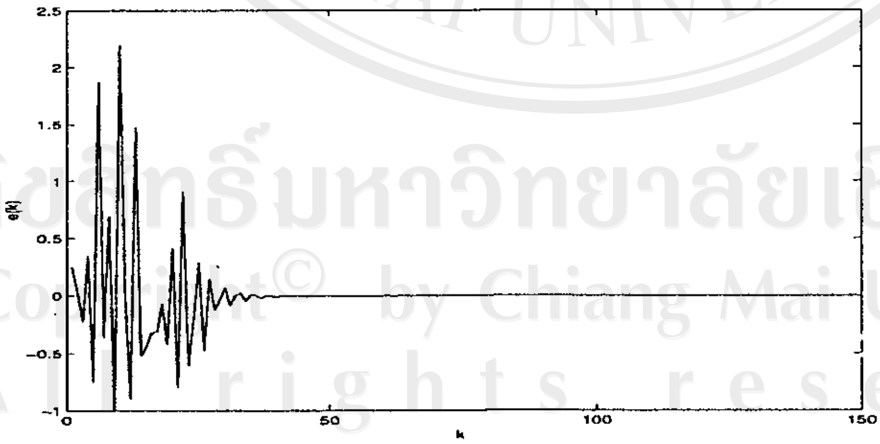
Figure 6.10: Simulation results of Hénon map control using MIFREN.



(a) Output $x_2(k)$.



(b) Control effort $u(k)$ and its bounds.



(c) Error $e(k)$.

Figure 6.11: Simulation results of Hénon map control using MiFren at $c = [1 \quad 4]$.

6.3 Hybrid Learning Algorithm for MIFREN

In this section, a hybrid algorithm to adjust all MIFREN's parameters is proposed. The normalized least-mean square and the error backpropagation are modified to adjusted LC and MF parameters, respectively. The performance of the identification based on MIFREN and its hybrid learning technique is presented by using the computer simulation test. The comparison between the proposed method and the well known ANFIS is discussed.

6.3.1 Adaptation of LC Parameters

Here, an adaptive technique based on adaptive filter [55] is proposed to adjust all LC's parameters i.e. $\boldsymbol{\beta}$ during the system operation. We define the error signal as,

$$\hat{e}(k) = d(k) - y(k), \tag{6.17}$$

where $d(k)$ is the output from the nonlinear system under determine and $y(k)$ is the estimated value of $d(k)$ obtained from MIFREN at time index k.

From Eq.(6.4), the output from MIFREN is given by

$$y(k) = \boldsymbol{\beta}(k)^T \boldsymbol{F}(k), \tag{6.18}$$

and Eq.(6.17) can be written as

$$\hat{e}(k) = d(k) - \boldsymbol{\beta}(k)^T \boldsymbol{F}(k). \tag{6.19}$$

Next define the cost function as

$$J(k) = \Big[\boldsymbol{\beta}(k+1) - \boldsymbol{\beta}(k)\Big]^T \Big[\boldsymbol{\beta}(k+1) - \boldsymbol{\beta}(k)\Big] + \lambda[d(k) - \boldsymbol{\beta}(k+1)^T \boldsymbol{F}(k)], \tag{6.20}$$

the value of *Lagrange multiplier* λ will be determined shortly. The local minimum point of $J(k)$ occurs when

$$\frac{\partial J(k)}{\partial \boldsymbol{\beta}(k+1)} = 2\Big[\boldsymbol{\beta}(k+1) - \boldsymbol{\beta}(k)\Big] - \lambda \boldsymbol{F}(k) = 0, \tag{6.21}$$

which gives

$$\boldsymbol{\beta}(k+1) = \boldsymbol{\beta}(k) + \frac{1}{2}\lambda \boldsymbol{F}(k). \tag{6.22}$$

Thus λ can be obtained as the following:

$$\begin{aligned} d(k) &= \beta(k+1)^T \boldsymbol{F}(k), \\ &= \left[\beta(k) + \frac{1}{2}\lambda \boldsymbol{F}(k)\right]^T \boldsymbol{F}(k), \\ &= \beta(k)^T \boldsymbol{F}(k) + \frac{1}{2}\lambda \|\boldsymbol{F}(k)\|^2, \\ \frac{1}{2}\lambda\|\boldsymbol{F}(k)\|^2 &= d(k) - \beta(k)^T \boldsymbol{F}(k), \\ &= \hat{e}(k), \\ \lambda &= \frac{2\hat{e}(k)}{\|\boldsymbol{F}(k)\|^2}. \end{aligned} \tag{6.23}$$

Thus the LC parameter adaptation becomes

$$\beta(k+1) = \beta(k) + \frac{\hat{e}(k)}{\|\boldsymbol{F}(k)\|^2}\boldsymbol{F}(k), \tag{6.24}$$

where $\|\boldsymbol{F}(k)\|$ is not zero because of its membership functions property.

6.3.2 Adaptation of MF Parameters

The technique of error back-propagation is applied to tune MF parameters. Here the cost function is defined as

$$\hat{\xi}(k) = \frac{1}{2}[d(k) - y(k)]^2. \tag{6.25}$$

The MF parameters of the i-th fuzzy set for the j-th input a_{ij} are adjusted by

$$a_{ij}(k+1) = a_{ij}(k) - \eta_0 \frac{\partial \hat{\xi}(k)}{\partial a_{ij}(k)}, \tag{6.26}$$

where $\eta_0 > 0$ is a pre-defined constant. And

$$\frac{\partial \hat{\xi}(k)}{\partial a_{ij}} = -\hat{e}(k)\frac{\partial y(k)}{\partial a_{ij}} = -\hat{e}(k)\beta(k)^T \frac{\partial \boldsymbol{F}(k)}{\partial a_{ij}}. \tag{6.27}$$

Consider the l-th element of $\boldsymbol{F}(k)$, then from Eq.(6.1) we have

$$\begin{aligned} \frac{\partial f_l(k)}{\partial a_{ij}} &= \frac{\partial}{\partial a_{ij}} f_{l_1, l_2, \ldots, l_n} \\ &= \frac{\partial}{\partial a_{ij}} \prod_{t=1}^{n} \mu_{A_{l_t},t}(I_t(k)) \\ &= \frac{f_l}{\mu_{A_j,j}(I_j(k))} \frac{\partial \mu_{A_j,j}(I_j(k))}{\partial a_{ij}}. \end{aligned}$$

Thus

$$\frac{\partial \boldsymbol{F}(k)}{\partial a_{ij}} = \frac{1}{\mu_{A_j,j}(I_j(k))} \frac{\partial \mu_{A_j,j}(I_j(k))}{\partial a_{ij}} \boldsymbol{F}(k). \tag{6.28}$$

And the update equation becomes

$$a_{ij}(k+1) = a_{ij}(k) + \eta_0 \delta_{ij}(k), \qquad (6.29)$$

where

$$\begin{aligned}\delta_{ij}(k) &= \frac{\hat{e}(k)}{\mu_{A_j,j}(I_j(k))}\frac{\partial \mu_{A_j,j}(I_j(k))}{\partial a_{ij}}\beta(k)^T \boldsymbol{F}(k), \\ &= \frac{\hat{e}(k)}{\mu_{A_j,j}(I_j(k))}\frac{\partial \mu_{A_j,j}(I_j(k))}{\partial a_{ij}} y(k). \end{aligned} \qquad (6.30)$$

More details of this adaptation will be discussed later in the computer simulation subsection.

6.3.3 Nonlinear System Identification Results

In this simulation test, the comparison between the MiFren and the well known ANFIS [28] are performed. ANFIS is selected because all of its parameters, linear and nonlinear, are adjusted and its performance is well known. Both networks are used to identify a two-dimensional sinc function defined by

$$z(x,y) = \frac{\sin(x)}{x}\frac{\sin(y)}{y}, \qquad (6.31)$$

its characteristic is illustrated in Fig. 6.12.

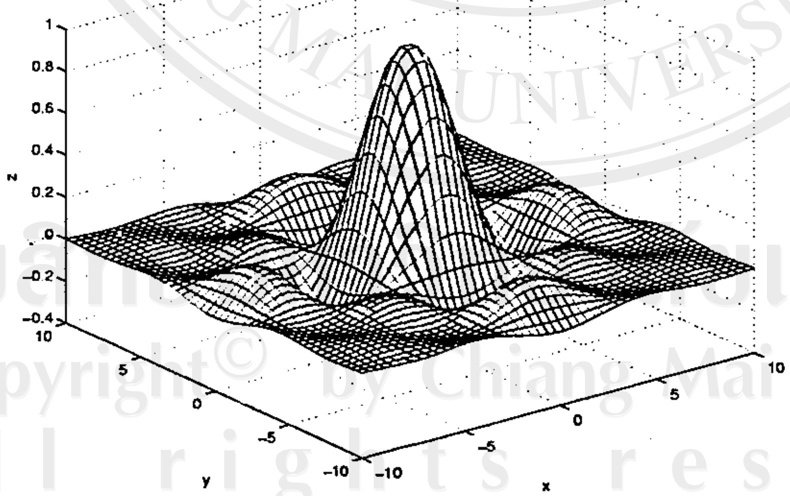


Figure 6.12: Two-dimensional sinc function.

At first, the two-input ANFIS is used to model the sinc function. Each input is classified into 3 linguistic values ,i.e., Negative, Zero and Positive. The training procedure is performed around 50 epochs. The initial and the final parameters are

are obtained as shown in Fig. 6.13. The result of ANFIS model is shown in Fig. 6.14. In this simulation, two 50×50 points random data sets are selected. One is used to adjust parameters in the learning phase and the other is used to test the results.

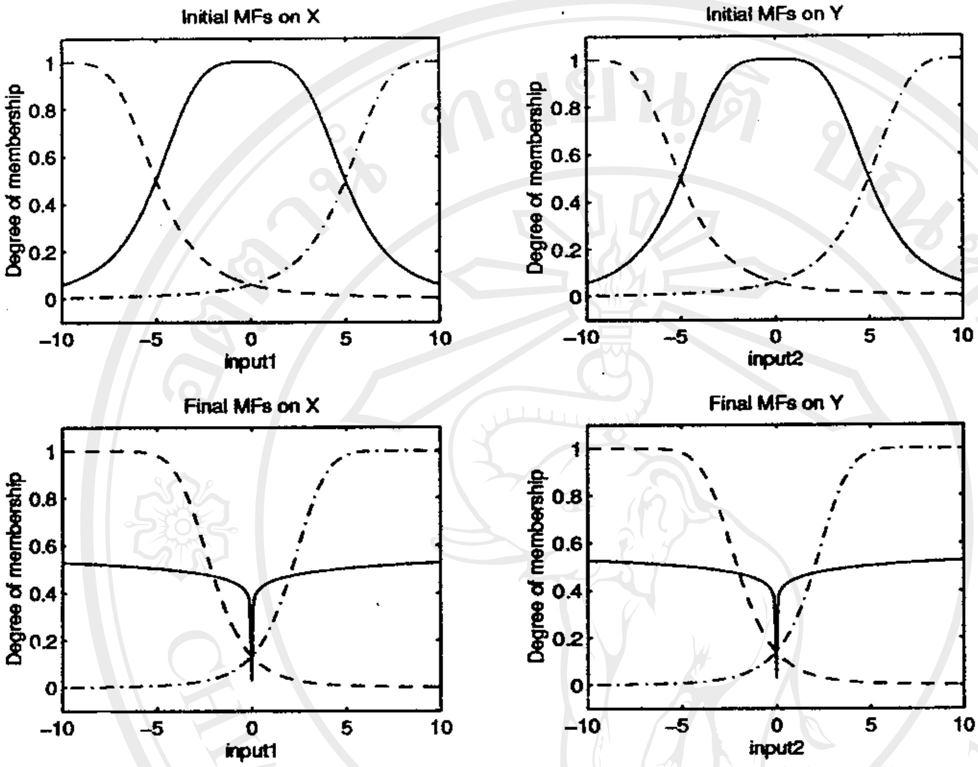


Figure 6.13: Initial and final parameters setting of ANFIS.

Next MiFren and its hybrid learning algorithm are used to indentify the sinc function. The estimated sinc output z is obtained as

$$\hat{z} = f_{\text{MF}}(x, y). \tag{6.32}$$

The If-Then rules are defined as the following:

Rule 1	If x is N	and y is N	Then $\hat{z}_1 = \beta_1 f_1(k)$,
Rule 2	If x is N	and y is Z	Then $\hat{z}_2 = \beta_2 f_2(k)$,
Rule 3	If x is N	and y is P	Then $\hat{z}_3 = \beta_3 f_3(k)$,
Rule 4	If x is Z	and y is N	Then $\hat{z}_4 = \beta_4 f_4(k)$,
Rule 5	If x is Z	and y is Z	Then $\hat{z}_5 = \beta_5 f_5(k)$,
Rule 6	If x is Z	and y is P	Then $\hat{z}_6 = \beta_6 f_6(k)$,
Rule 7	If x is P	and y is N	Then $\hat{z}_7 = \beta_7 f_7(k)$,
Rule 8	If x is P	and y is Z	Then $\hat{z}_8 = \beta_8 f_8(k)$,
Rule 9	If x is P	and y is P	Then $\hat{z}_9 = \beta_9 f_9(k)$,

here N, Z and P denote negative, zero, and positive linguistic level respectively. The initial value of β_i and f_i $(i = 1, 2, \cdots, 9)$ are defined as

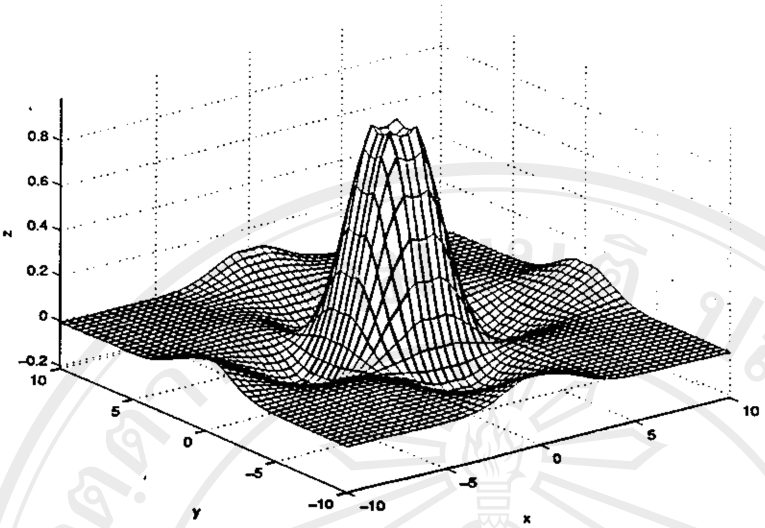


Figure 6.14: Estimated two-dimensional sinc using ANFIS.

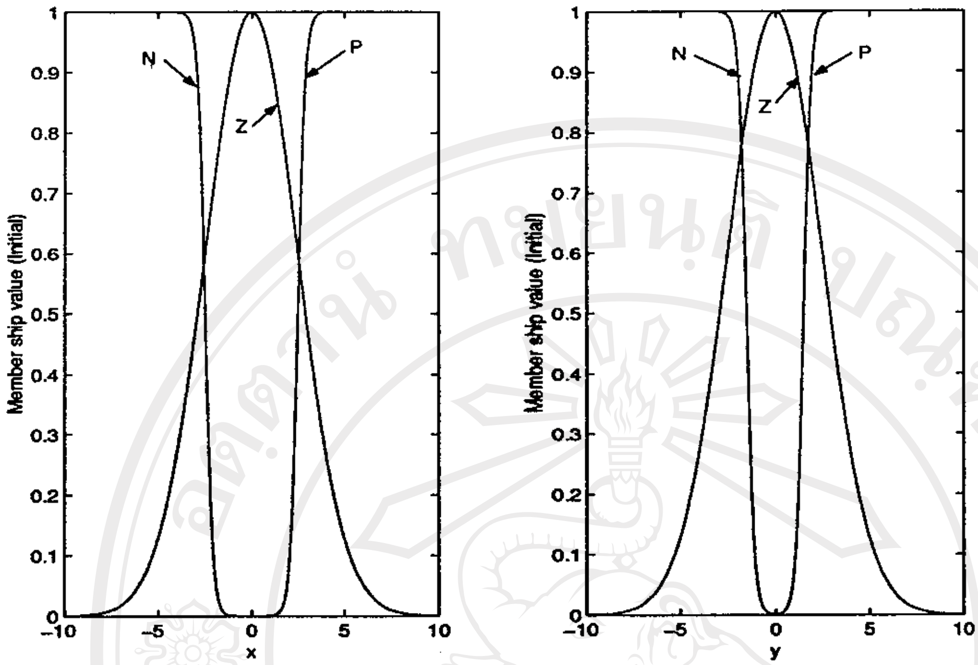
$\beta_1 = -1.0$	$f_1 = \mu_{\mathrm{N}x}(x)\mu_{\mathrm{N}y}(y)$,
$\beta_2 = -0.8$	$f_2 = \mu_{\mathrm{N}x}(x)\mu_{\mathrm{Z}y}(y)$,
$\beta_3 = -0.5$	$f_3 = \mu_{\mathrm{N}x}(x)\mu_{\mathrm{P}y}(y)$,
$\beta_4 = -0.2$	$f_4 = \mu_{\mathrm{Z}x}(x)\mu_{\mathrm{N}y}(y)$,
$\beta_5 = 0.0$	$f_5 = \mu_{\mathrm{Z}x}(x)\mu_{\mathrm{Z}y}(y)$,
$\beta_6 = 0.2$	$f_6 = \mu_{\mathrm{Z}x}(x)\mu_{\mathrm{P}y}(y)$,
$\beta_7 = 0.5$	$f_7 = \mu_{\mathrm{P}x}(x)\mu_{\mathrm{N}y}(y)$,
$\beta_8 = 0.8$	$f_8 = \mu_{\mathrm{P}x}(x)\mu_{\mathrm{Z}y}(y)$,
$\beta_9 = 1.0$	$f_9 = \mu_{\mathrm{P}x}(x)\mu_{\mathrm{P}y}(y)$.

The MF parameters are selected as shown in Fig. 6.15 (a). The parameter learning process is finished after 50 epochs and the final MF parameters are shown in Fig. 6.15 (b). The LC parameters β_i after learning process become,

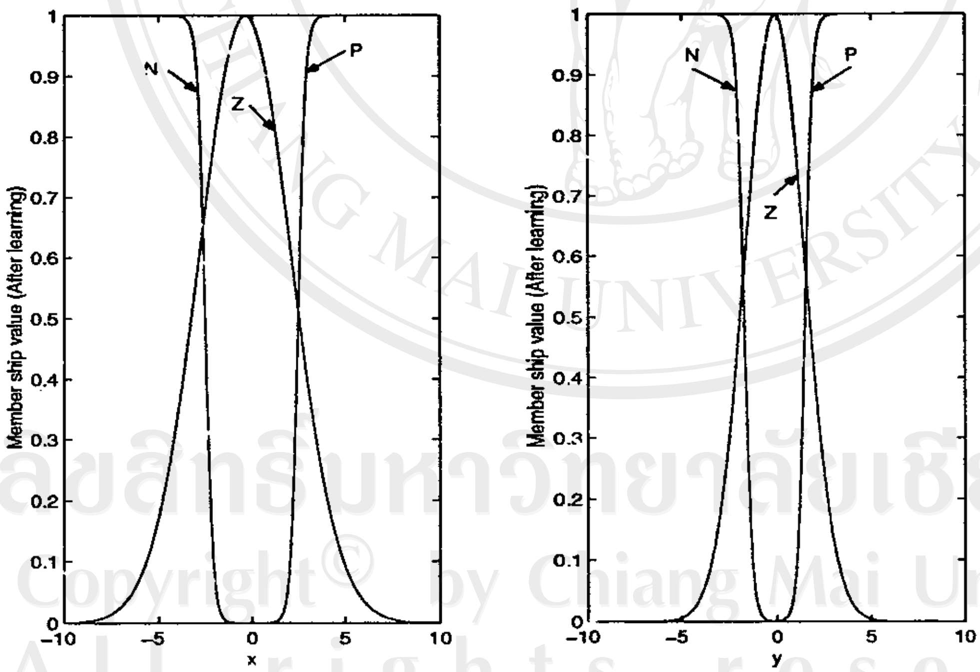
$\beta_1 = 0.2614$	$\beta_2 = 0.1149$	$\beta_3 = -0.1607$,
$\beta_4 = -0.6449$	$\beta_5 = -0.5237$	$\beta_6 = -0.6022$,
$\beta_7 = -0.0149$	$\beta_8 = -0.0397$	$\beta_9 = 0.0027$.

The result of MiFren model is shown in Fig. 6.16. Fig. 6.17 shows the RMSE (root mean squared error) curves for both ANFIS and MiFren. Here RMSE is definded as $\frac{\sqrt{\sum_{i=1}^{n} \hat{e}_i^2}}{n}$.

In this simulation experiment, these results show that MiFren has the impressive performance over ANFIS when the numbers of adjustable parameters are equal. The convergent speed of MiFren is faster and the RMSE of MiFren is less than ANFIS's.



(a) Initial setting.



(b) Final setting.

Figure 6.15: MF's parameter of MiFREN for the sinc function identification.

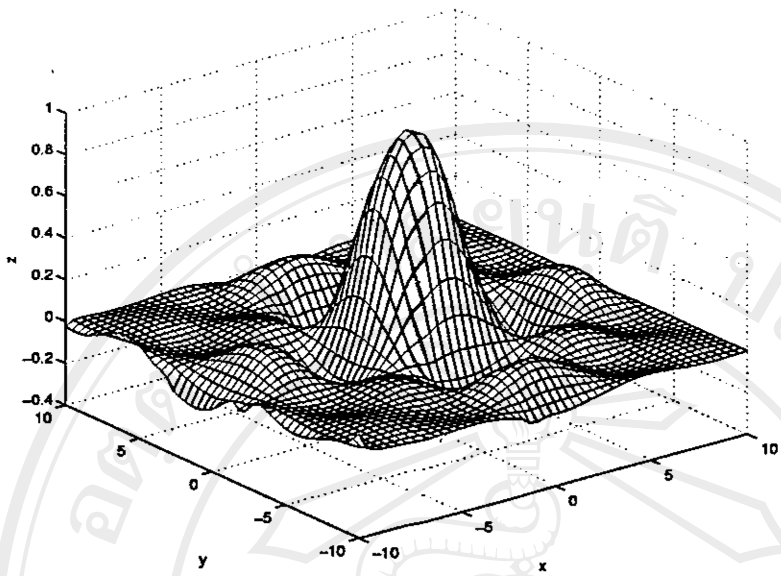


Figure 6.16: MiFren result of two-dimensional sinc identification.

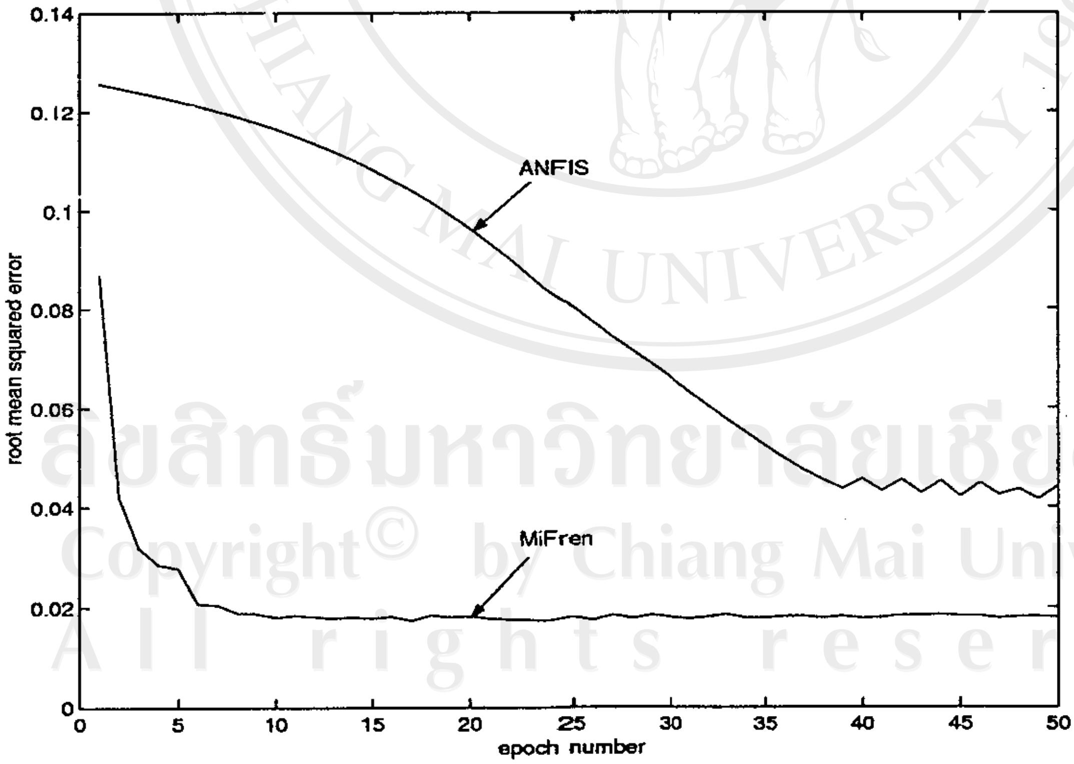


Figure 6.17: RMSE curves of two-dimensional sinc identification.