NEURAL UNITS WITH HIGHER-ORDER SYNAPTIC OPERATIONS WITH APPLICATIONS TO EDGE DETECTION AND CONTROL SYSTEMS

A Thesis

Submitted to the College of Graduate Studies and Research in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in the

Intelligent Systems Research Laboratory

Department of Mechanical Engineering

University of Saskatchewan

Saskatoon, Saskatchewan

CANADA

By

KI-YOUNG SONG

July 2004

Permission To Use

The author has agreed that the library, University of Saskatchewan may make this thesis freely available for inspection. Moreover, the author has agreed that permission for extensive copying of this thesis for scholarly purpose may be granted by the professor who supervised the thesis work recorded herein or, in his absence, by the head of the department or the Dean of the college in which this thesis work was done. It is understood that due recognition will be given to the author of this thesis and to the University of Saskatchewan in any use of the material in the thesis. Copying for publication or any other use of this thesis for financial gain without approval by the University and the author's written permission is strictly prohibited.

Requests for permission to copy or make any other use of the material in this thesis in whole or part should be addressed to:

Head of the Department of Mechanical Engineering

University of Saskatchewan

57 Campus Drive

Saskatoon, Saskatchewan

Canada S7N 5A9

Acknowledgements

I wish to express my sincere gratitude to Dr. Madan M. Gupta for his valuable guidance, supervision and support throughout the course of this work. His encouragement and positive criticisms are mainly responsible for the success of the project. Also, I would like to thank Dr. Zeng-Guang Hou who has provided numerous valuable comments during this thesis.

The scholarships from the University of Saskatchewan as well as from the Natural Science and Engineering Research Council (NSERC) are gratefully acknowledged.

I would like to thank Drs. Bok-Hyun Yun, Il-Soo Kim and Hyung-Bae Jung of Mokpo National University in South Korea for having encouraged me to carry on the M.Sc. program at the University of Saskatchewan.

I would like to express my sincere thanks to my families in Saskatoon, Thailand and Korea for providing encouragement and moral support throughout this research endeavor.

Abstract

The biological sense organ contains infinite potential. The artificial neural structures have emulated the potential of the central nervous system; however, most of the researchers have been using the linear combination of synaptic operation. In this thesis, this neural structure is referred to as the neural unit with *linear synaptic operation* (LSO).

The objective of the research reported in this thesis is to develop novel neural units with *higher-order synaptic operations* (HOSO), and to explore their potential applications. The neural units with *quadratic synaptic operation* (QSO) and *cubic synaptic operation* (CSO) are developed and reported in this thesis. A comparative analysis is done on the neural units with LSO, QSO, and CSO. It is to be noted that the neural units with lower order synaptic operations are the subsets of the neural units with higher-order synaptic operations. It is found that for much more complex problems the neural units with higher-order synaptic operations are much more efficient than the neural units with lower order synaptic operations.

Motivated by the intensity of the biological neural systems, the dynamic nature of the neural structure is proposed and implemented using the neural unit with CSO. The dynamic structure makes the system response relatively insensitive to external disturbances and internal variations in system parameters. With the success of these dynamic structures researchers are inclined to replace the recurrent (feedback) *neural networks* (NNs) in their present systems with the neural units with CSO.

Applications of these novel dynamic neural structures are gaining potential in the areas of image processing for the machine vision and motion controls. One of the machine vision emulations from the biological attribution is edge detection. Edge detection of images is a significant component in the field of computer vision, remote sensing and image analysis. The neural units with HOSO do replicate some of the biological attributes for edge detection. Further more, the developments in robotics are gaining momentum in neural control applications with the introduction of mobile robots, which in turn use the neural units with HOSO; a CCD camera for the vision is implemented, and several photo-sensors are attached on the machine. In summary, it was demonstrated that the neural units with HOSO present the advanced control capability for the mobile robot with neuro-vision and neuro-control systems.

Table of Contents

List of Figures

Chapter 3

Chapter 4

List of Symbols

Symbol	Meaning
Chapter 1	
$x_1, x_2, \ldots, x_n$	Neural inputs
$w_1, w_2, \ldots, w_n$	Neural synaptic weights
$\mathbf{w}^T$	Synaptic weight vector
$\mathbf{x}$	Neural input vector
v	Synaptic output
x_0	Constant bias
w_0	Threshold (bias) weight
$\mathbf{w_a}^T$	Augmented synaptic weight vector
$\mathbf{x_a}$	Augmented neural input vector
$\Phi[\bullet]$	Nonlinear activation function
$\Re$	Set of real numbers
y_N	Neural output

Chapter 2

k	Discrete time index
$y_d(k)$	Desired output
$e(k)$	Error signal
$r(k)$	Reference signal
w_{ij}	Synaptic weight of the neural unit with QSO
w_{ijk}	Synaptic weight of the neural unit with CSO
$\Delta w_{ijk}(k)$	Adjustment in the synaptic weight of the neural unit with CSO
$w_{i_1 i_2 \cdots i_Z}(k)$	Synaptic weight of Z^{th} order neural unit
$\Delta w_{i_1 i_2 \cdots i_Z}(k)$	Adjustment in the synaptic weight of Z^{th} order neural unit
$J[\bullet]$	Error function (LMS)
$E[\bullet]$	Expectation function
M	Number of observation

μ	Learning rate
$\Phi'[\bullet]$	Slope of nonlinear activation function
OR	**OR** logic operation
AND	**AND** logic operation
XOR	**XOR** logic operation

Chapter 3

$d(k)$	Output of DNS
$G_{tf}(k)$	Transfer function of DNS
z^{-1}	Unit delay operator (memory)
$b(k)$	Feedback weight
$\nabla_b J[b(k)]$	Gradient with respect to feedback weight
$S_b(k)$	Sensitivity signal
E_{total}	Total error function for batch training
E	Error function for on-line training
A	Set of indices *j*

j	Index of neuron
$e_j(n)$	Error signal at the output
n_0	Start time of an epoch
n_1	End time of an epoch
$\delta_j(n)$	Local gradient
$v_j(n)$	Synaptic operation of the *j*-th neural unit with CSO
$w_{abc}^{(j)}$	Synaptic weight of the *j*-th neural unit with CSO

Chapter 4

$s(p)$	Color spatial function
p	Pixels of an image
$G(c)$	Ganglion signal
$R(c)$	Receptor signal
c	Spatial coordinate

$Ga(x)$	Gaussian function
α	Slope rate of the Gaussian function
$DG(x)$	1st order differentiation of the Gaussian function
$LG(x)$	2nd order differentiation of the Gaussian function
$D_n[\bullet]$	*n*-th order differential operator
X_n	Input of the differential operator at *n*-th time

Chapter 5

θ	Angle to point **P** from the origin
ρ	Distance to point **P** from the origin
$e_a(t)$	Armature voltage
R_m	Resistance
L_m	Inductance
$e_m(t)$	Back-EMF
K	Motor parameter

ϕ	Field flux
θ	Angle of the motor shaft
J	Moment of inertia
B	Friction
$G(s)$	Transfer function of servomotor
Ref	Reference input
C	Output of the motor
y_{ref}	Output of the reference model
E	Error

List of Tables

Chapter 1

Introduction

1.1 Biological Motivation

The mass of the vertebrate of the central nervous system, called the brain, executes incomprehensible functions such as cognition, intelligence, and emotions and many more. These recondite functions of the brain make researchers curious and study them. As a result of this research on the biological sensitive mass, which consists of biological neural networks, the concept of neural networks gained its importance. This concept has been applied for manipulating rather practical and useful devices applications like system identification, pattern recognition, and control systems. The contemporary neural network systems have proved their success so convincingly for they are now used for unpredictable problems such as weather forecasting and stock predictions [2].

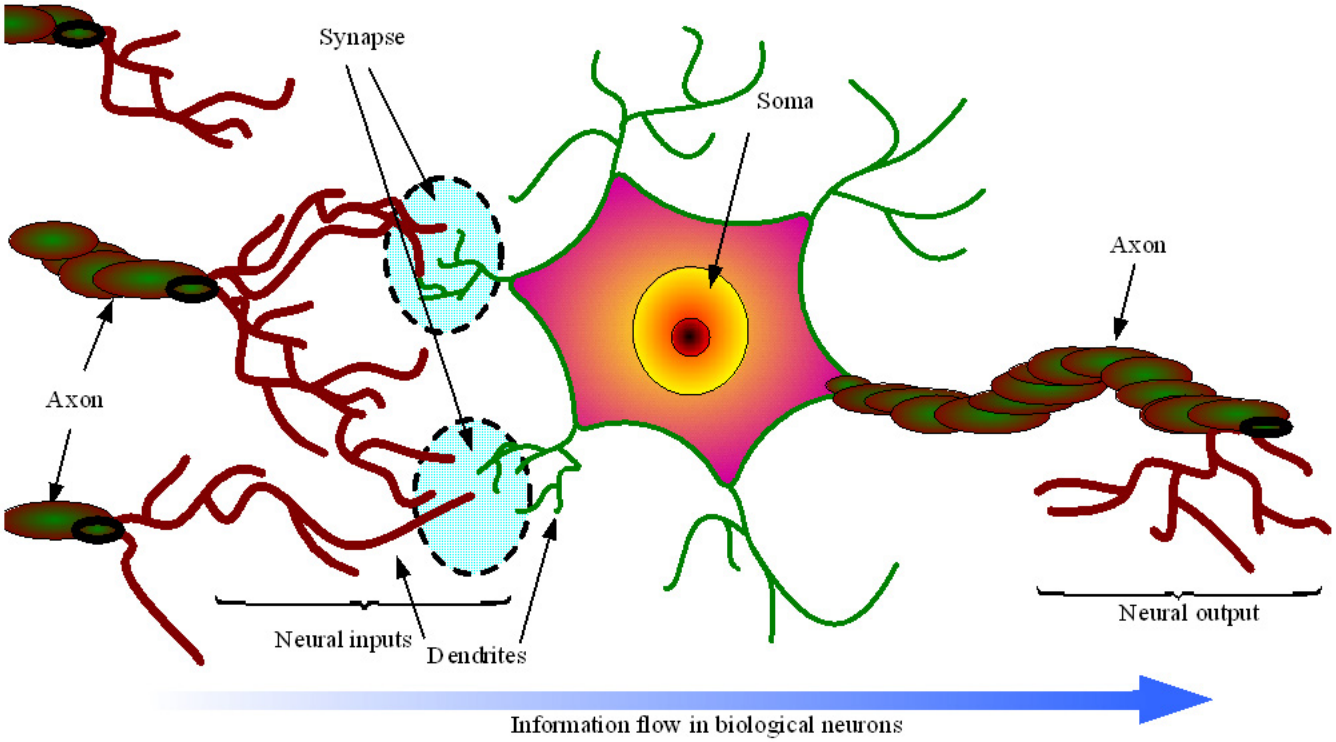
The biological sense organ contains infinite potential. In the literature, the neural structures have emulated the potential of the central nervous system to a great extent; however, most of the researchers have been using the linear combination of synaptic operation for generating this behaviour. Nevertheless, the neural networks with the linear neural structures are required to have numerous neural units to emulate the nonlinear functions of the biological neural networks. Additionally, these neural networks may not be able to present the superior performance of the natural neural networks. As a further research breakthrough, the present linear structure of the neuron is extended in this study to higher-order (nonlinear) formations which include both linear and nonlinear structure,

in order to effectively capture the prospective tasks of the brain. Furthermore, mathematical replication of this procedure of the central nervous system will help researchers to develop more robust mechanisms for real life complex solving problems. Based on this inspiration, this thesis describes novel numerical structures of neural units and neural networks in higher-order structures. These innovative neural units with *higher-order synaptic operations* (HOSO) may impressively and convincingly assist engineers and scientists for their application in their respective fields.

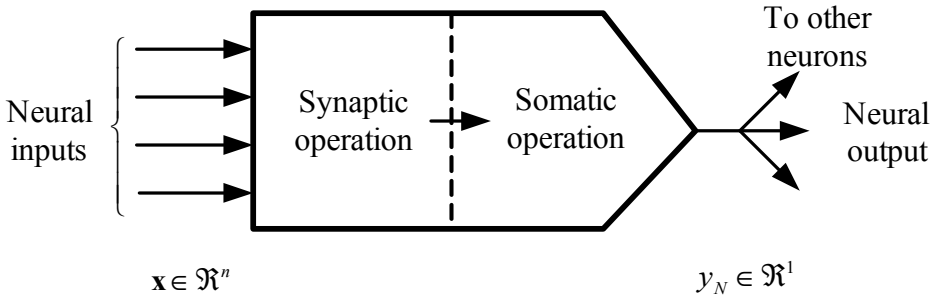
1.2 Basic Knowledge of Neural Units and Neural Networks

1.2.1 Biological Neurons and Neural Units: Neural Units with Linear Synaptic Operation (LSO)

In simple words, the neural unit replicates the task of a biological neuron. The biological neuron is shown in Fig. 1.1(a) and the model of the neuron is described in Fig. 1.1(b). The biological neuron primarily does not only consist of the synapse and soma, but also has dendrites. Information from stimuli flows from the dendrites of one neuron to that of another neuron. This procedure is called the synaptic operation. After synaptic processing, the information is passed through the soma to other neurons, and this operation is called the somatic operation. The artificial neural unit operates in a very similar way and process input signals. The biological neuron has numerous dendrites for receiving data and forward this processed information to other neurons. The neural unit also has multiple input terminals and a single output terminal, which may be considered as a *multiple-input/single-output* (MISO) system [3].



(a) A schematic view of a biological neuron



(b) Model representation of a biological neuron with multiple inputs $\mathbf{x} \in \Re^n$, and a single input, $y_N \in \Re^1$

Figure 1.1 A biological neuron and its model

1.2.1.1 Synaptic Operation: Synaptic Weights and Threshold

The synapse of the biological neuron is considered as a storage element of the past experience or knowledge learned from the neuronal environment. That experience or knowledge continuously adapts its strength in synapse. There are over 1000 synapses in each biological neuron. In the neural structure, the past experience is called synaptic weight [3]. The connotation of the synaptic weights signifies the importance of the given neural inputs, $\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]^{\mathbf{T}} \in \Re^n$. The synaptic weights are expressed as $\mathbf{w} = [w_1 \quad w_2 \quad \cdots \quad w_n]^{\mathbf{T}} \in \Re^n$. In biological neurons, the new stimulus after the synapse is compared with the old stimulus which is called the threshold. The neural unit emulates the threshold as well. The neural inputs and synaptic weights with the threshold is defined as augmented neural input $\mathbf{x_a} = [x_0 \quad x_1 \quad \cdots \quad x_n]^{\mathbf{T}} \in \Re^{n+1}$ and augmented synaptic weights $\mathbf{w_a} = [w_0 \quad w_1 \quad \cdots \quad w_n]^{\mathbf{T}} \in \Re^{n+1}$.

The synaptic operation with the threshold is represented as

$$\begin{aligned} v &= \sum_{i=0}^{n} w_i x_i = w_0 x_0 + w_1 x_1 + \cdots + w_n x_n \\ &= \mathbf{w_a}^{\mathbf{T}} \mathbf{x_a} \\ &= |\mathbf{w_a}| \; |\mathbf{x_a}| \cos\theta \end{aligned} \tag{1.1}$$

where $v \in \Re^1$ is the neural synaptic output, w_0 is the threshold (bias) weight, $x_0 = 1$ is the constant bias and θ is the phase difference between neural inputs and synaptic weights.

The synaptic operation represents the difference between the memory (synaptic weights $\mathbf{w}_a$) and the new information (neural inputs $\mathbf{x}_a$). The geometric explanation of synaptic operation is shown in Fig. 1.2.

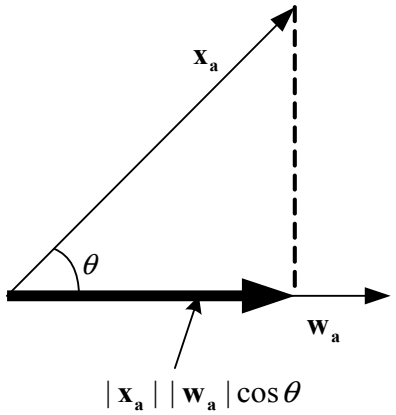
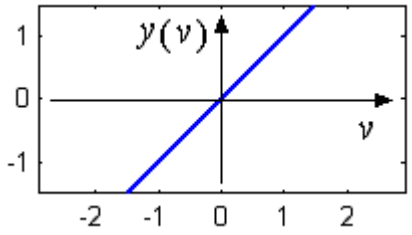


Figure 1.2 The geometric representation of synaptic operation

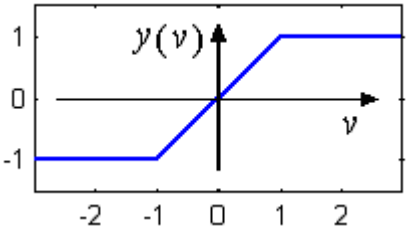
The linear characteristic of the synaptic operation guides the neural structure to the neural unit with *linear synaptic operation* (LSO).

1.2.1.2 Somatic Operation: Activation Functions

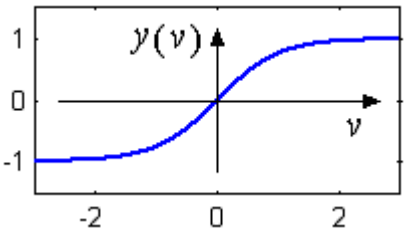
After the linear synaptic operation, the combination of the neural inputs and the neural synaptic weights, are applied to a nonlinear activation function ($\Phi[\bullet]$) which is considered as the somatic operation. Figure 1.3 shows the different kinds of activation functions which are currently used as linear or nonlinear mapping functions [3].



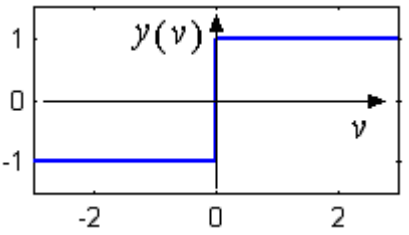
(a) Linear function: $y(v) = v$



(b) Linear with saturation: $\begin{cases} y(v) = -1 & (v < -1) \\ y(v) = v & (-1 \le v \le 1) \\ y(v) = 1 & (v > 1) \end{cases}$



(c) Sigmoid function: $y(v) = \dfrac{e^{v} - e^{-v}}{e^{v} + e^{-v}}$



(d) Saturation function: $\begin{cases} y(v) = -1 & (v < 0) \\ y(v) = 1 & (v \ge 0) \end{cases}$

Figure 1.3 Various somatic activation functions

For the neural structures the choice of somatic activation function depends on the performance and nature of its applications [2]. However, the sigmoid function is deemed to be a general activation function due to the differentiable nonlinear distinctiveness [3]. The entire process of the neural unit with LSO with *n* inputs passing through the somatic operation gives a neural output y_N as follows

$$y_N = \Phi[v] \in \Re^1 \tag{1.2}$$

The schematic diagram of the neural unit with LSO is shown in Fig. 1.4.

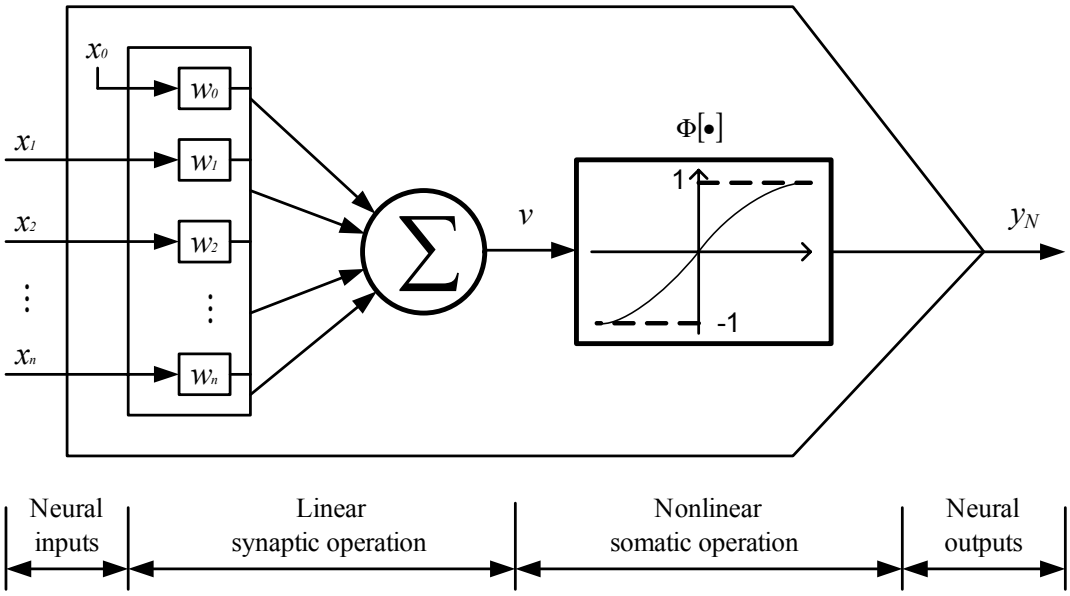


Figure 1.4 A neural unit with linear synaptic operation and nonlinear somatic operation

1.2.2 Neural Networks (NNs)

In the central nervous system cells are positioned in parallel layers [4]. The *neural networks* (NNs) are designed to emulate the properties of the biological neural systems. Generally, NNs are framed and portrayed as parallel distributed units [2, 5]. Their crucial

ability is their aptitude of learning and adaptation. Moreover, the parallel nature aids the overall procedure of the networks, although some artificial neurons fail to process the given signal [6, 7]. Typically, the structure of NNs consists of one input layer, several hidden layers and one output layer. This structure is called the *multilayered feedforward neural network* (MFNN). The static behaviour of the neural unit with LSO leads to this type of feedforward network with no dynamics (feedback) in the network. Figure 1.5 shows a MFNN with three layers, one input layer, one hidden layer and one output layer. The input layer is nothing but a set of the neural inputs [9]. Each neuron in the hidden and output layer is a neural unit with LSO.

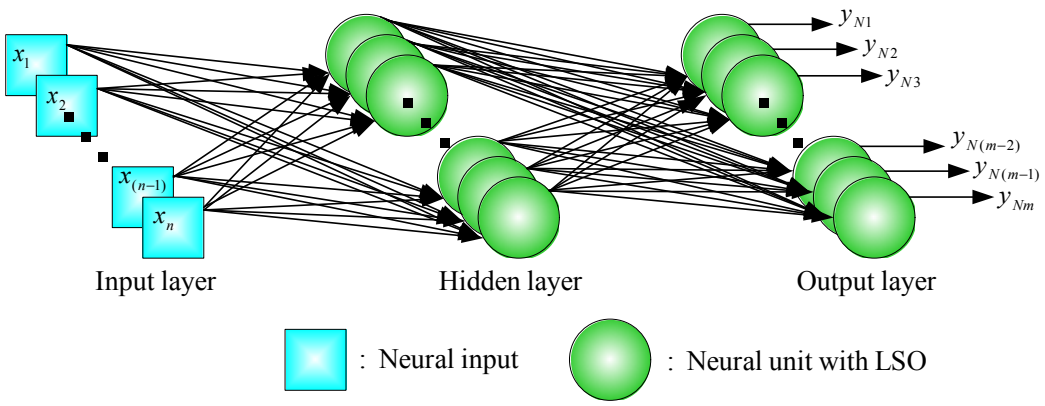


Figure 1.5 Parallel connections of MFNN with one input layer, one hidden layer and one output layer

Besides the MFNN, another type of NNs has been developed and is called the *dynamic (recurrent) neural networks* (DNNs) which is nothing but static neurons with feedback [8, 10, 11, 12]. DNNs find applications in the fields of system identification and control of unknown dynamic systems [11, 14, 15]. This feedback unit is considered as a memory of the system and extends more advanced computational methods for the network by

affecting the other neural units in excitatory or inhibitory mode [5, 7, 8]. The structure of a DNN is shown in Fig. 1.6. For the purpose of clarity, the connections for only one neural unit in each layer are shown. Every neuron is correlated with the feedback and has feedforward connections with other neural units.

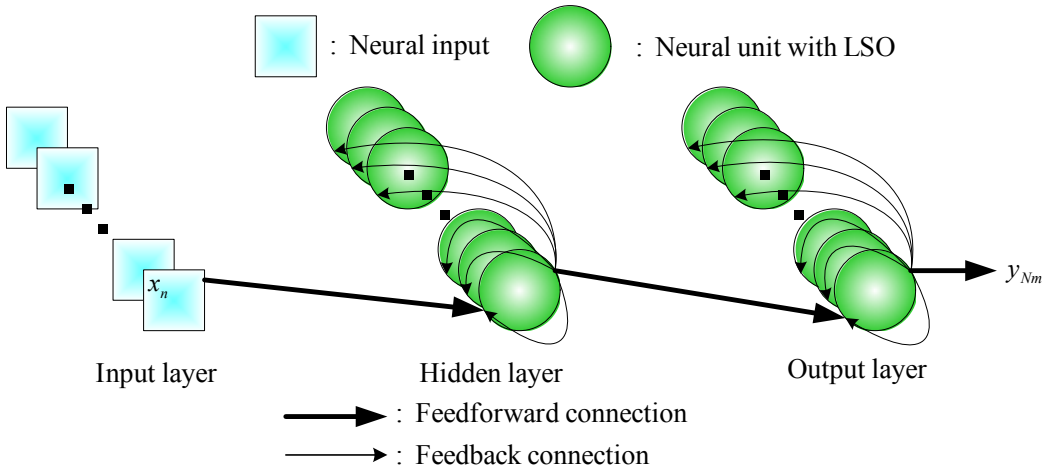


Figure 1.6 Multilayered NN by interconnecting neurons with feedforward and feedback

1.3 Thesis Objectives

In a word, the neural unit with LSO embraces a linear synaptic operation which is different from the biological neural synapse. In natural synapse, the association between the prior neuron and the posterior neuron is extraordinarily versatile as seen in Fig. 1.1. The dendrite connection is coupled in many ways. However, in the neural unit with LSO, there are only one-to-one correspondences in the synapse, which results in the linearity of the neural structures. In order to emulate the biological synapse more effectively, the synaptic operation of the neural unit with LSO is modulated. In the synapse, the outer stimuli from natural sensors such as eyes, ears, and skin etc. can be amplified and deleted

in conformity with the significance of the information. The multi-correspondence of the dendrites plays the principal role and adds to the importance of the stimuli. In this thesis, the linear synaptic operation of the neural unit with LSO is expanded to the nonlinear synaptic operation to build novel static and dynamic higher-order neural structures to prove superior capabilities of the neural units with *higher-order synaptic operations* (HOSO) over the neural unit with LSO. The nonlinearity of the synaptic operation characterizes the neural unit with HOSO. The higher-order amalgamation of neural inputs and neural weights therefore provides advanced neural performance. The objectives of this thesis are as follows:

- To develop a nonlinear synaptic operation for the neural structure and to be able to come up with an enhanced performance.
- To formulate dynamic neural units with HOSO based on the dynamic neural structure.
- To apply the neural units with HOSO and dynamic neural units with HOSO for image processing, model identification and controllers for the nonlinear system as static and dynamic approaches.

1.4 Layout of the Thesis

The novel neural units with HOSO with the second-order synaptic operation are described as the neural unit with *quadratic synaptic operation* (QSO) and that with the third order synaptic operation are described as the neural unit with *cubic synaptic operation* (CSO). In Chapter 2, the **XOR** logic problem is solved in order to prove the superior capability of the neural units with HOSO. In addition, the mathematical details

are explained to formulate the generalization of the neural units with HOSO with the *learning and adaptation rule* (LAR). In Chapter 3, a dynamic structure is applied to the neural unit with CSO based on the developed dynamic neural structure, and the dynamic neural unit with *cubic synaptic operation* (CSO) is introduced with LAR. As simulation results, model identifications with both of the dynamic neural unit with CSO and the static neural unit with CSO are compared, and the effect of the dynamic structure is explained. In Chapter 4, the neural unit with CSO is illustrated on how it performs as edge detection for image processing, and in Chapter 5, a neuro-control system driving a mobile robot, which is a control application, is developed. Finally, conclusions and future work are given in Chapter 6.

Chapter 2

Development of Neural Units with Higher-Order Synaptic Operations (HOSO)

2.1 Introduction

In the literature, the neural units are considered to have linear synaptic connection that leads to the neural unit with LSO, and this differs from the biological neurons. The natural junction, called the synapse, contains complex correlation between the pre-synaptic nerve cells and the post-synaptic nerve cells [4]. This characteristic biological linkage is focussed and being emulated to escalate the performance of the neural units. However, the performance of the neural unit with LSO has several disadvantages when compared to a biological neuron. For example, the neural unit with LSO is inadequate in solving complex problems and has difficulty in learning process like translation, rotation and scale-invariant pattern recognition, motion detection and so forth [3]. The urge to replicate the multi-correspondence of the natural synapse motivated this research to come up with neural units with HOSO to replace the neural unit with LSO. The architectures of the neural units with HOSO are accomplished by capturing the higher-order association as well as the linear association between the elements of the input patterns [3]. The second-order association of the neural inputs with the synaptic weights leads to the neural unit with *quadratic synaptic operation* (QSO) and the third order neural association results in the neural unit with *cubic synaptic operation* (CSO) [16, 34]. The properties of the neural units with HOSO depend on the cross correlation and the inter-dependence of

the neural inputs. Hence, the neural units with HOSO are more sensitive than the neural unit with LSO. Additionally, the biological evidence accumulated in the past supports the presence of multiplicative-like operations in the brain, which can be considered as the simplest fundamental nonlinear operation.

In this chapter, the innovative neural structures namely, the neural units with QSO and CSO are introduced and discussed first. Later, the focus shifts to second and third order neural unit structures and finally the general formula for the neural unit with Z^{th} order synaptic operation is established. In addition, in order to prove the potential of the neural units with HOSO, the **XOR** logic problem is solved with both the neural unit with LSO and the neural units with HOSO.

2.2 Neural Unit with Quadratic Synaptic Operation (QSO)

The synaptic operation of the neural unit with *quadratic synaptic operation* (QSO) embraces both the first and second-order neural input combinations with the synaptic weights. A neural unit with QSO with *n*-dimensional neural inputs and the association of the neural weights and the neural inputs is depicted in Fig. 2.1. The augmented neural inputs are defined as $\mathbf{x_a} = [x_0 \quad x_1 \quad \cdots \quad x_n]^{\mathbf{T}} \in \Re^{n+1}$. Neural inputs and synaptic weights are amalgamated, and the nonlinear synaptic output of the neural unit with QSO is given in Eqn. (2.1a) and (2.1b).

$$v = \sum_{i=0}^{n}\sum_{j=i}^{n} w_{ij}x_i x_j = w_{00}x_0x_0 + w_{01}x_0x_1 + \ldots \\ \ldots + w_{n(n-1)}x_n x_{n-1} + w_{nn}x_n x_n \tag{2.1a}$$

or

$$v = \mathbf{x_a}^{\mathrm{T}} \mathbf{w_a} \mathbf{x_a} \tag{2.1b}$$

where w_{00} is the threshold (bias) weight and $x_0 = 1$ is the constant bias.

The augmented synaptic weights for the neural unit with QSO are defined in the matrix form as

$$\mathbf{w_a} = \begin{bmatrix} w_{00} & w_{01} & \cdots & w_{0n} \\ 0 & w_{11} & \cdots & w_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{nn} \end{bmatrix} \in \Re^{(n+1)\times(n+1)} \tag{2.2}$$

The upper triangle matrix elements are considered as the synaptic weights of the neural unit with QSO. The index number (0, 1, …, *n*) of each synaptic weight indicates that the synaptic weight is compounded with the same index numbered neural inputs. The processed inputs pass through the nonlinear activation function followed by the synaptic operation, and the output of the neural unit with QSO is given as

$$y_N = \Phi[v] \in \Re^1 \tag{2.3}$$

The schematic view of the neural unit with QSO is shown in Fig. 2.1 with nonlinear synaptic operation and somatic operation.

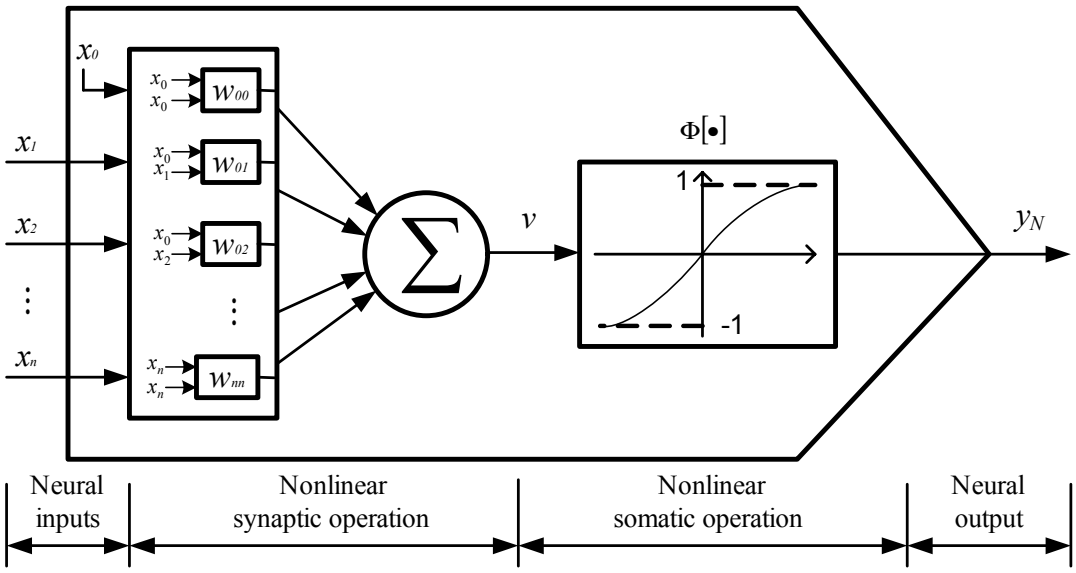


Figure 2.1 The structure of the neural unit with QSO with the higher-order computation of the neural inputs and the synaptic weights

Consider the first row in the weight matrix of the neural unit with QSO. The first row of the matrix is $[w_{00} \quad w_{01} \quad \cdots \quad w_{0n}]$. The elements in the first row are multiplied with the neural inputs corresponding to the index number as given in Eqn. (2.4). The sum of the product of the first row in the weight matrix of the neural unit with QSO shows the same combination of the synaptic operation of the neural unit with LSO which is in linear arrangement.

$$\begin{aligned} \sum_{i=0}^{0}\sum_{j=i}^{n} w_{ij}x_i x_j &= w_{00}x_0x_0 + w_{01}x_0x_1 + \cdots + w_{0n}x_0x_n \\ &= w_{00}x_0 + w_{01}x_1 + \cdots + w_{0n}x_n, \; x_0 = 1 \end{aligned} \tag{2.4}$$

The linear element in the synaptic operation combination of the neural unit with QSO tells that the neural unit with QSO contains the property of the neural unit with LSO. Thus, the neural unit with LSO is a subset of the neural unit with QSO [16].

2.3 Neural Unit with Cubic Synaptic Operation (CSO)

2.3.1 Structure and Mathematical Development of Neural Unit with CSO

The synaptic mechanism of the neural unit with *cubic synaptic operation* (CSO) is that the correlation of the synaptic operation is more intricate with the first, second and third order connections algebraically with neural inputs and synaptic weights. Figure 2.2 describes the structure of the neural unit with CSO. The synaptic operation of the neural unit with CSO is defined as

$$v = \sum_{i=0}^{n}\sum_{j=i}^{n}\sum_{k=j}^{n} w_{ijk}x_i x_j x_k = w_{000}x_0x_0x_0 + w_{001}x_0x_0x_1 + \cdots \\ \cdots + w_{nn(n-1)}x_n x_n x_{n-1} + w_{nnn}x_n x_n x_n \tag{2.5}$$

where w_{000} is the threshold (bias) weight and $x_0 = 1$ is the constant bias.

The higher-order combination of the neural inputs and synaptic weights shows that the synaptic operation of the neural unit with CSO has the property of nonlinearity. The somatic operation of the neural unit with CSO with a nonlinear activation function is defined as

$$y_N = \Phi[v] \in \Re^1 \tag{2.6}$$

From Eqn. (2.5), it is discovered that the synaptic operation of the neural unit with CSO encloses the first and second-order numerical combination as well as the third order numerical computation of neural inputs and synaptic weights. The association corresponding to the lower order neural units represents the linear and quadratic

combination of the first and second-order numerical mixtures which are similar to the synaptic operations of the neural units with LSO and QSO. Thus, the neural unit with LSO and QSO are subsets of the neural unit with CSO [34].

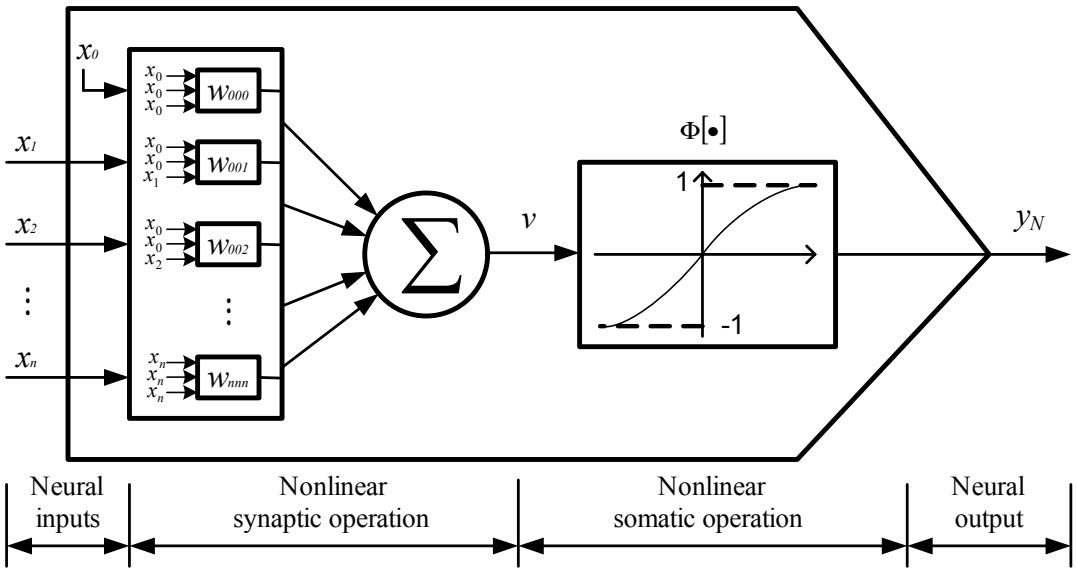


Figure 2.2 The structure of the neural unit with CSO with the higher-order computation of the neural inputs and the synaptic weights

These figures can represent any of the neural units with LSO, QSO and CSO. The main distinction of those units is the synaptic combination with neural inputs and neural weights. The higher-ordered synaptic operation takes more neural inputs and higher computations. These higher computations make the neural units more adaptable for the environments.

2.3.2 Learning and Adaptation for Weight Elements of Neural Unit with CSO

Learning and adaptation in neural structures assist the neurons to perform more efficiently. The *learning and adaptation rule* (LAR) makes the neural units memorize the

information and act on the new data corresponding to the previous data as the biological neurons do. LAR optimizes the synaptic weights in order to process the neural inputs. The LAR for the neural units with LSO and QSO can be inferred from the LAR of the neural unit with CSO. Therefore, in this section, the LAR is implemented for a neural unit with CSO.

The synaptic weights play a significant role in tracking the desired output. Thus, the error between desired system and the neural structure could be reduced with the LAR. Figure 2.3 shows the learning procedure of the neural unit with CSO with LAR.

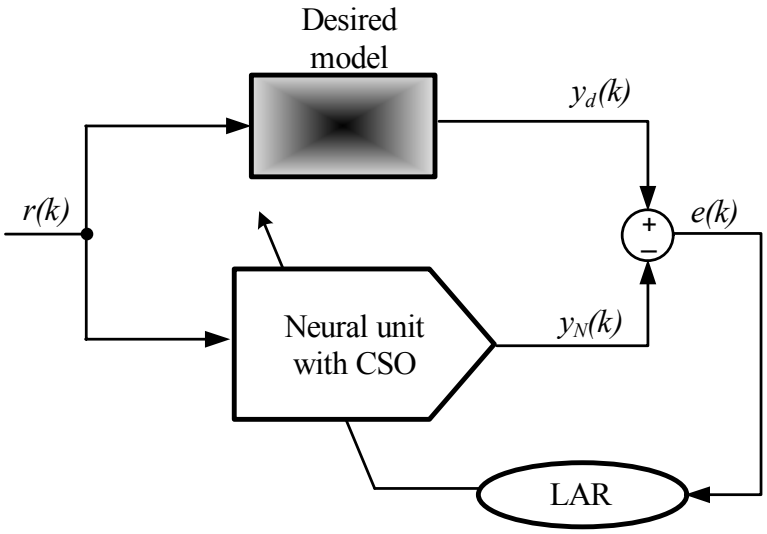


Figure 2.3 Optimization of the synaptic weights of the neural unit with CSO by the *learning and adaptation rule* (LAR)

The error is defined as

$$e(k) = y_d(k) - y_N(k) \tag{2.7}$$

where $y_d(k)$ is the desired output and $y_N(k)$ is the output of the neural unit with CSO and *k* represents discrete time.

The error computed by LAR is applied to change the synaptic weights and thresholds optimally [9]. The objective of the LAR is to reduce the error, so that the neural output $y_N(k)$ approaches the desired output $y_d(k)$. In order to reduce the error, an error function $J[\bullet]$ is applied as

$$J[e\{k)] = \frac{1}{2} E[e^2(k)] \tag{2.8}$$

where $E[\bullet]$ is an expectation function which is regarded as the average of the error squares [3].

The error function $J[\bullet]$ is called the *least mean square* (LMS). The average function $E[\bullet]$ is defined as

$$E[\bullet] = \frac{1}{M} \sum_{l=1}^{M} e_l{}^2(k) \tag{2.9}$$

where *M* is number of observations.

The adaptation for the synaptic weights is defined as

$$w_{ijk}(k+1) = w_{ijk}(k) + \Delta w_{ijk}(k) \tag{2.10}$$

where *i*, *j*, and *k* are indices of the synaptic weights.

A new weight $w_{ijk}(k+1)$ is then defined by summing the previous weight $w_{ijk}(k)$ and the adjustment $\Delta w_{ijk}(k)$, where $\Delta w_{ijk}(k)$ is defined by the error and the gradient of the error function as

$$\Delta w_{ijk}(k) = -\mu \frac{\partial J[e(k)]}{\partial w_{ijk}(k)} \tag{2.11}$$

where μ is the learning rate and it is a constant and determines the step size of the synaptic weights to find out their optimal value.

The range of the step size is recommended from 10^{-3} to 10 throughout the experiment [2]. The error function with respect to the synaptic weight is derived as

$$\frac{\partial J(e(k))}{\partial w_{ijk}(k)} = \frac{1}{2} E\left[\frac{\partial e^2(k)}{\partial w_{ijk}(k)}\right]$$

$$= E\left[e(k)\left\{\frac{\partial e(k)}{\partial w_{ijk}(k)}\right\}\right]$$

$$= E\left[e(k)\left\{\frac{\partial (y_d(k) - y_N(k))}{\partial w_{ijk}(k)}\right\}\right]$$

$$= E\left[e(k)\left\{-\frac{\partial y_N(k)}{\partial w_{ijk}(k)}\right\}\right] \tag{2.12}$$

The output $y_N(k)$ of the neural unit with CSO is obtained after a nonlinear somatic operation is carried out. This neural output $y_N(k)$ depends on the steepness of the slope of the nonlinear activation function. The threshold and the synaptic weights influence the

slope of the activation function. Error function is derived in terms of the neural inputs in Eqn. (2.13).

$$E\left[e(k)\left\{-\frac{\partial y_N(k)}{\partial w_{ijk}(k)}\right\}\right] = E\left[e(k)\left\{-\frac{\partial \Phi[v(k)]}{\partial w_{ijk}(k)}\right\}\right]$$

$$= E\left[e(k)\left\{-\frac{\partial \Phi[v(k)]}{\partial v(k)}\frac{\partial v(k)}{\partial w_{ijk}(k)}\right\}\right]$$

$$= E\left[e(k)\left\{-\Phi'[v(k)]\frac{\partial v(k)}{\partial w_{ijk}(k)}\right\}\right]$$

$$= E\left[e(k)\left\{-\Phi'[v(k)]\frac{\partial\left(\sum_{i=0}^{n}\sum_{j=i}^{n}\sum_{k=j}^{n} w_{ijk}(k)x_i(k)x_j(k)x_k(k)\right)}{\partial w_{ijk}(k)}\right\}\right]$$

$$= -E[e(k)x_i(k)x_j(k)x_k(k)\Phi'\{v(k)\}] \tag{2.13}$$

where $\Phi'[\bullet]$ is the slope of the somatic operation and is obtained by the differentiation of the nonlinear activation function.

Further, Eqn. (2.11) can be rewritten as

$$\Delta w_{ijk}(k) = \mu E[e(k)x_i(k)x_j(k)x_k(k)\Phi'\{v(k)\}] \tag{2.14}$$

The LAR for the synaptic weights and the threshold for the neural unit with CSO can be derived from Eqn. (2.10) and Eqn. (2.14) as

$$w_{ijk}(k+1) = w_{ijk}(k) + \mu E[e(k)x_i(k)x_j(k)x_k(k)\Phi'\{v(k)\}] \quad (2.15)$$

2.4 Neural Pattern Classifiers

Pattern is defined as the quantitative sketch of an object, event, or phenomenon. The objective of pattern classification is to assign a physical object, event or phenomenon to one of several pre-specified classes or categories [2]. Traditionally, the neural structures have been used as pattern classifiers and have established an outstanding capability to pigeonhole the given information. The logical operations such as **NOT, OR, AND, NOR, NAND** and **XOR** are typical examples of pattern classification. Logical operations except for the **XOR** are linearly separable problems. The prototype of the exclusive-or function (**XOR**) is not linearly separable and includes many different local minima in the performance surface. The local minima easily attract the synaptic weights and the neural output zero, which makes the error constant. These neural structures may not work properly with constant error. The **XOR** is regarded as an excellent standard for evaluating the potential of pattern classifiers due to its non-linear features [9]. In this chapter, the typical neural classifiers performing **OR**, **AND** and **XOR** operations are described. Additionally, the performances of the neural units with HOSO with the **XOR** problem demonstrate the prospect and need of the novel higher-order structures.

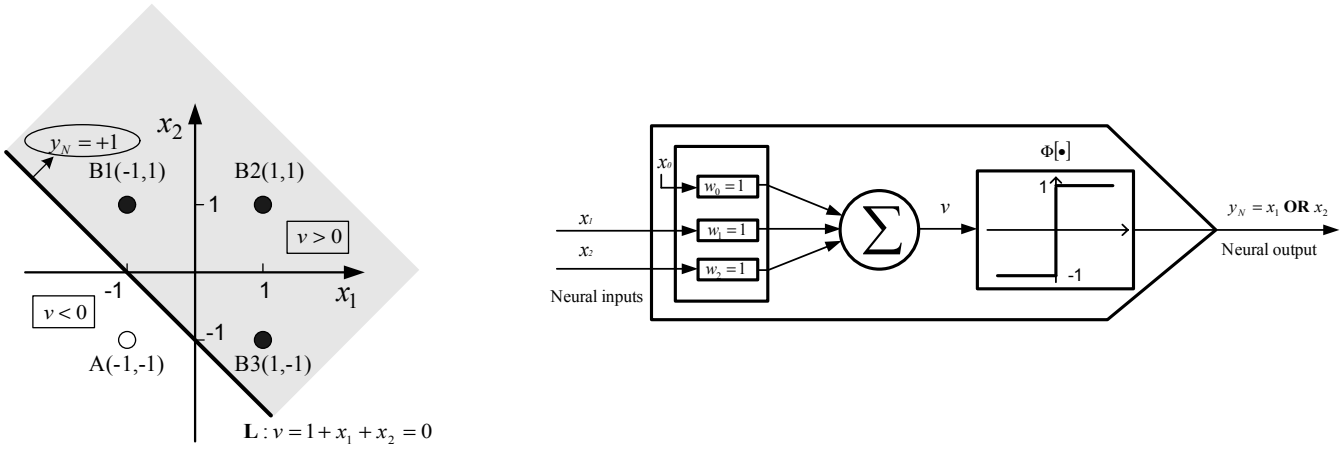
2.4.1 OR, AND and XOR Logic Operations with Neural Pattern Classifier with LSO

The **OR,** and **AND** logic operations are defined with neural inputs and neural outputs in Table 2.1 and 2.2. The two logic operations can be classified in two classes, Class A and

Class B according to the neural outputs. A neural pattern classifier with LSO generates one linear discriminant line between different classes of outputs. Figure 2.4 and 2.5 show the classification of Class A and Class B with neural **OR** and **AND** logic classifier with LSO. In the figures, one discriminant line can categorize the two classes.

Table 2.1 **OR** logic operation

Neural inputs		Neural Outputs
x_1	x_2	$y_N = x_1$ **OR** x_2
-1	-1	-1: A
-1	1	1: B1
1	-1	1: B2
1	1	1: B3
Class A = {A} Class B = {B1, B2, B3}		

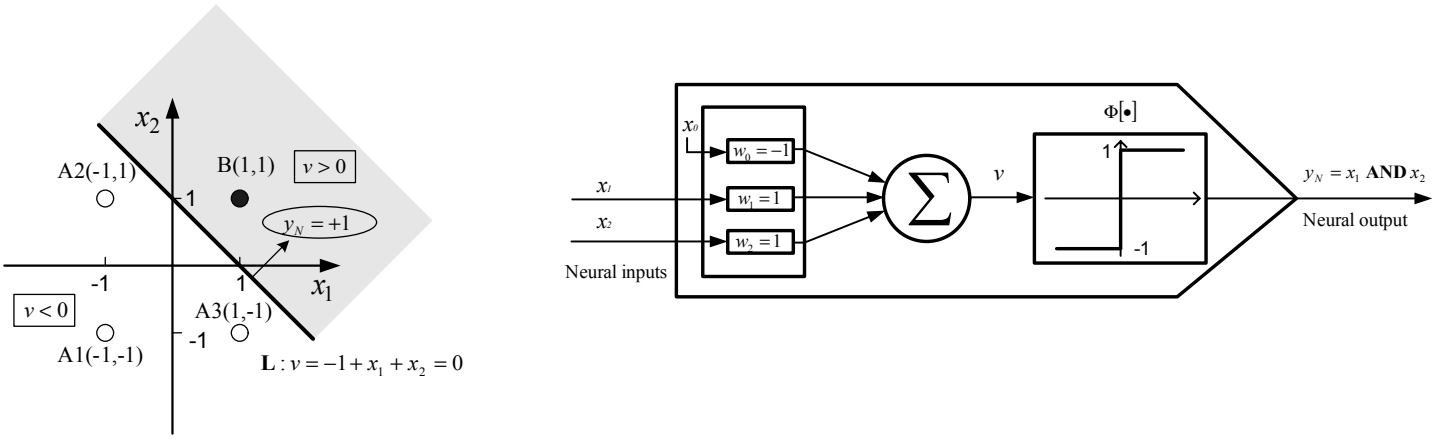


(a) A linear discriminant line (b) Neural **OR** classifier with LSO

Figure 2.4 A linear synaptic operation for **OR** logic operation

Table 2.2 **AND** logic operation

Neural inputs		Neural Outputs
x_1	x_2	$y_N = x_1$ **AND** x_2
-1	-1	-1: A1
-1	1	-1: A2
1	-1	-1: A3
1	1	1: B
Class A = {A1, A2, A3} Class B = {B}		



(a) A linear discriminant line (b) Neural **AND** classifier with LSO

Figure 2.5 A linear synaptic operation for **AND** logic operation

The neural pattern classifiers with LSO for the **OR** and **AND** logic problems are described in each figure. The synaptic weights of neural units with LSO are optimized for

these logic problems. A saturation function is applied as the somatic activation function for obtaining optimal neural outputs.

The **XOR** logic operation is defined in Table 2.3.

Table 2.3 XOR logic operation

Neural inputs		Neural Outputs
x_1	x_2	$y_N = x_1$ **XOR** x_2
-1	-1	-1: A1
-1	1	1: B1
1	-1	1: B2
1	1	-1: A2
Class A = {A1, A2} Class B = {B1, B2}		

The **XOR** logic operation can be expressed as

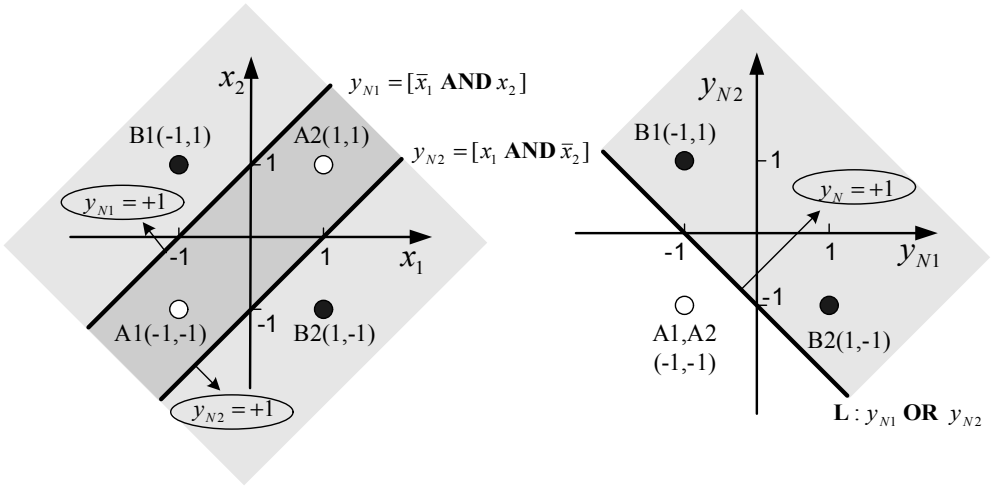
$$y_N = x_1 \textbf{ XOR } x_2 = [\bar{x}_1 \textbf{ AND } x_2] \textbf{ OR } [x_1 \textbf{ AND } \bar{x}_2] \tag{2.16a}$$

or

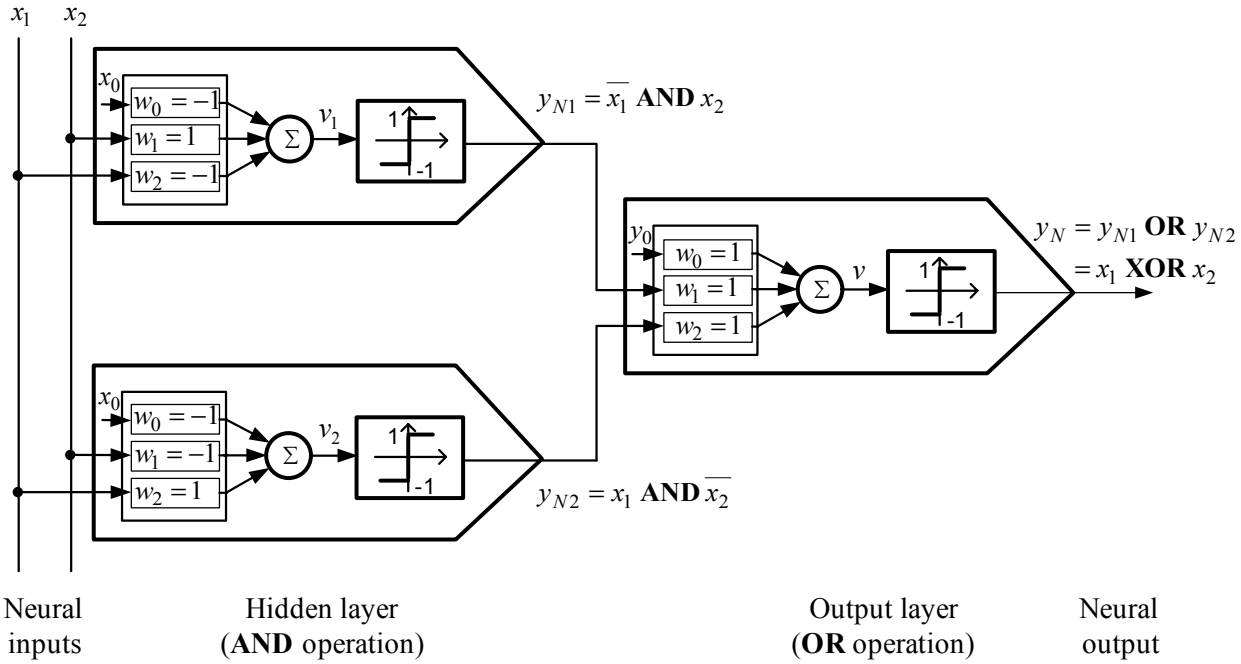
$$y_N = x_1 \textbf{ XOR } x_2 = [x_1 \textbf{ OR } x_2] \textbf{ AND } [\bar{x}_1 \textbf{ OR } \bar{x}_2] \tag{2.16b}$$

Unlike the **OR** and **AND** logic problems, the **XOR** logic problem contains nonlinear property. For the solution of the **XOR** logic problem, there are two possibilities to classify the Class A and Class B with linear discriminant lines. The geometric

representation of these methods is shown in Fig. 2.6 and 2.7 corresponding to Eqn. (2.16a) and (2.16b).

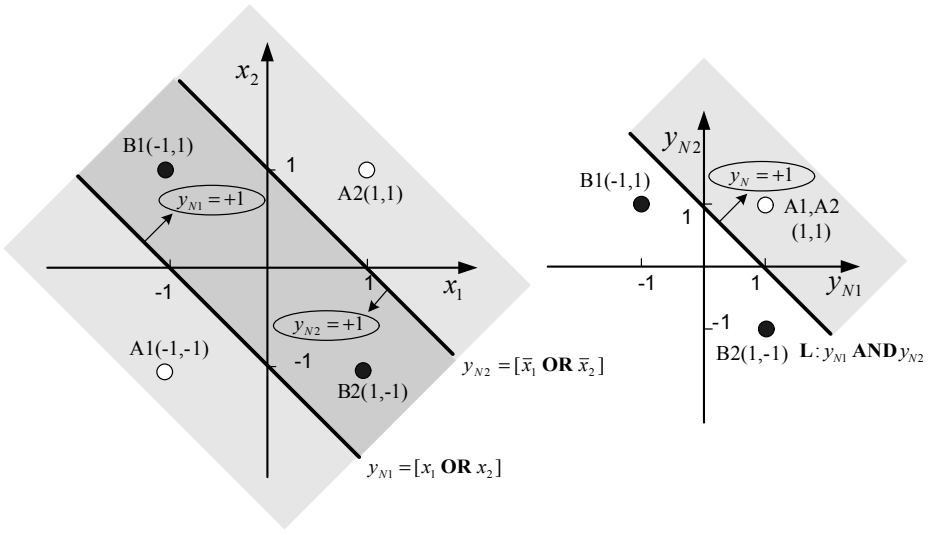


(a) $x_1 - x_2$ plane (b) $y_{N1} - y_{N2}$ plane

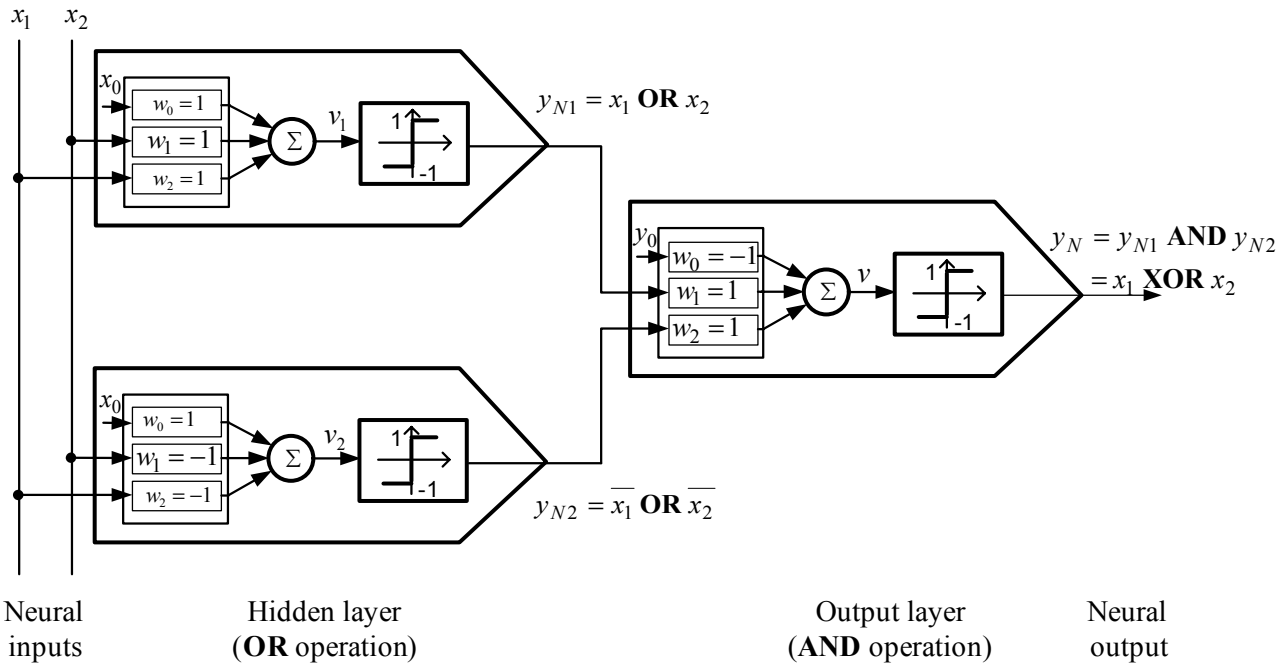


(c) Neural **XOR** classifier with LSO

Figure 2.6 Method 1: Geometric view of the mapping functions for **XOR** logic using two neural **AND**s and one neural **OR** operation, Eqn. (2.16a)



(a) $x_1 - x_2$ plane (b) $y_{N1} - y_{N2}$ plane



(c) Neural **XOR** classifier with LSO

Figure 2.7 Method 2: Geometric view of the mapping functions for **XOR** logic using two neural **OR**s and one neural **AND** operation, Eqn. (2.16b)

For the **OR** and **AND** operations, one neural unit with LSO was able to organize the outputs in Class A and Class B; however, for the **XOR** operation, one neural unit with

LSO is clearly incapable of categorizing the classes. In order to classify the **XOR** operation, three neural units with LSO in two layers should be applied to tackle the nonlinearity of the classes.

2.4.2 XOR Logic Operation with Neural Pattern Classifiers with HOSO

In the previous section, three neural units with LSO emulated the **XOR** operation. The neural unit with LSO is built with linear structure whereas the **XOR** operation has nonlinear property. The novel structures of the neural units with HOSO have the nonlinearity embedded in them and therefore these neural units have the capability to categorize the nonlinear properties with just one neuron. This is a more efficient and powerful way as compared to the neural unit with LSO. In this section, these novel neural units with QSO and CSO are focussed and their application in solving the **XOR** logic operation is illustrated.

2.4.2.1 Neural XOR Classification with QSO

The synaptic operation of the neural unit with QSO is described in Eqn. (2.2) and the neural input of the **XOR** operation is given in Table 2.3. With the neural inputs, the synaptic operation is derived as

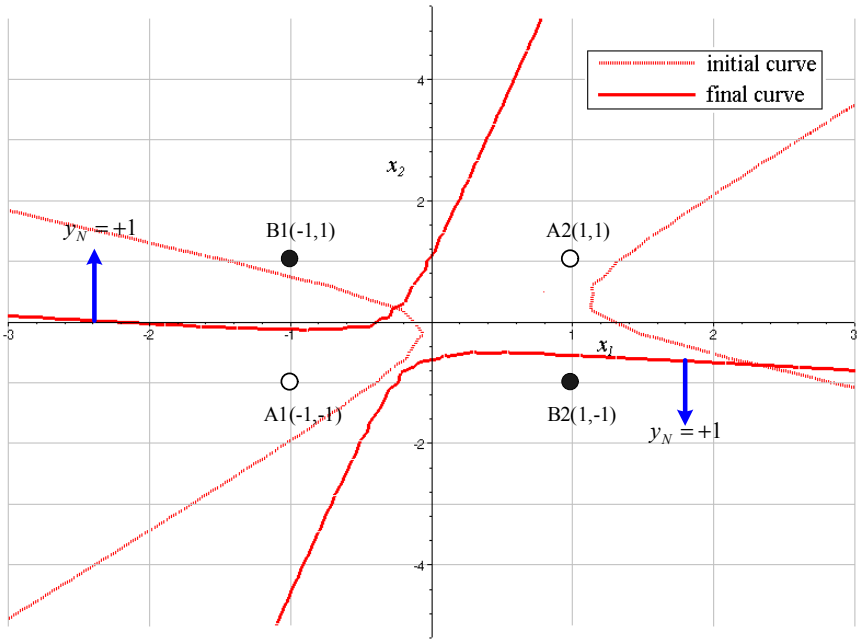
$$v = w_{00}x_0^2 + w_{01}x_0x_1 + w_{02}x_0x_2 + w_{11}x_1^2 + w_{12}x_1x_2 + w_{22}x_2^2, \quad x_0 = 1 \tag{2.17}$$

The nonlinearity of the synaptic operation of a neural unit with QSO makes the discriminant line curved corresponding to the initial values of the synaptic weights, and the discriminant curve equations can be represented by a polynomial. One discriminant

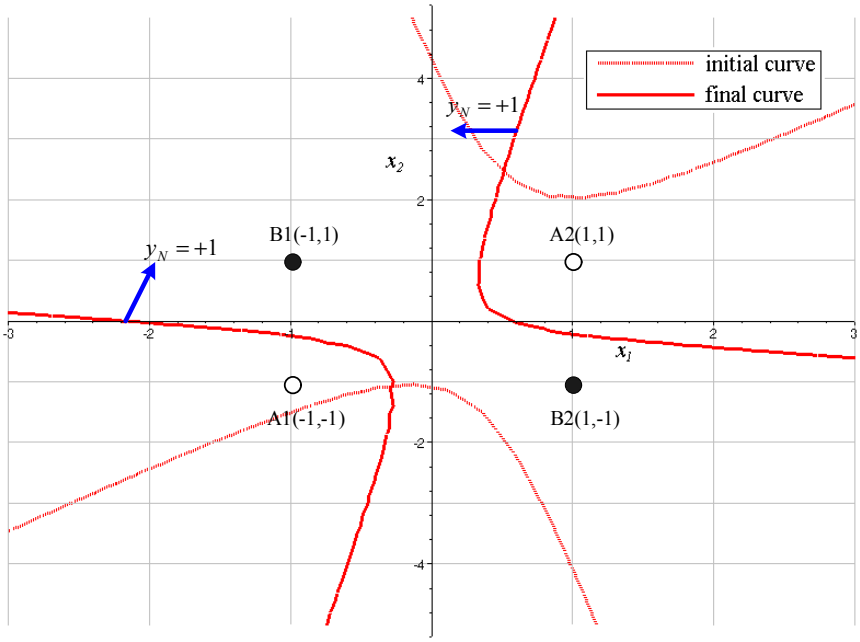
curve was able to classify the two classes of outputs. Table 2.4 gives the details of the initial and final synaptic weights. Three different results are provided, and these results are illustrated in the different figures in Fig. 2.8. The simulation was stopped when the error between the desired output and the neural output was less than 0.05.

Table 2.4 Initial and final synaptic weights in a neural unit with QSO for **XOR** operation

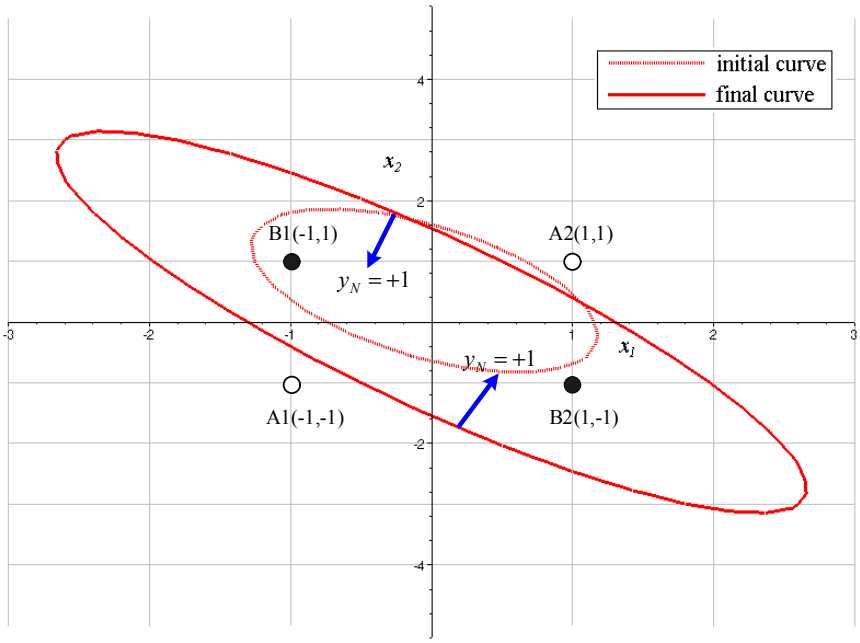
Case		Synaptic weights					
		w_{00}	w_{01}	w_{02}	w_{11}	w_{12}	w_{22}
I	Initial	-0.0784	-0.8245	-0.2674	0.7037	0.8995	-0.9715
	Final	-0.5079	-1.3828	-0.3589	-0.5338	-3.6217	0.7111
II	Initial	-0.6393	0.4679	-0.4479	-0.9742	0.7320	0.1390
	Final	0.5155	-0.6984	0.3299	-0.4425	-3.0623	0.4882
III	Initial	0.8471	0.3968	0.7790	-0.9661	-0.9609	-0.8028
	Final	1.1637	-0.00006	0.00006	-0.6782	-0.9990	-0.4849



(a) Case I $\begin{cases} \text{initial curve}: v = -0.0784 - 0.8245x_1 - 0.2674x_2 + 0.7037x_1^2 + 0.8995x_1x_2 - 0.9715x_2^2 \\ \text{final curve}: v = -0.5079 - 1.3828x_1 - 0.35899x_2 - 0.5338x_1^2 - 3.6217x_1x_2 + 0.7111x_2^2 \end{cases}$



(b) Case II $\begin{cases} \text{initial curve}: v = -0.6393 + 0.4679x_1 - 0.4479x_2 - 0.9742x_1^2 + 0.7320x_1x_2 + 0.1390x_2^2 \\ \text{final curve}: v = 0.5155 - 0.6984x_1 + 0.3299x_2 - 0.4425x_1^2 - 3.0623x_1x_2 + 0.4882x_2^2 \end{cases}$



(c) Case III $\begin{cases} \text{initial curve}: v = 0.8471 + 0.3968x_1 + 0.7790x_2 - 0.9661x_1^2 - 0.9609x_1x_2 - 0.8028x_2^2 \\ \text{final curve}: v = 1.1642 - 0.000068x_1 + 0.000061x_2 - 0.6785x_1^2 - 0.9995x_1x_2 - 0.4852x_2^2 \end{cases}$

Figure 2.8 Neural **XOR** classification with QSO and the discriminant curves from the polynomial equations of synaptic operation in different cases

2.4.2.2 Neural XOR Classification with CSO

The synaptic operation of the neural unit with CSO is described in Eqn. (2.5), and the synaptic operation is derived with the neural inputs as

$$\begin{aligned} v = {} & w_{000}x_0^3 + w_{001}x_0^2x_1 + w_{002}x_0^2x_2 + w_{011}x_0x_1^2 + w_{012}x_0x_1x_2 \\ & + w_{022}x_0x_2^2 + w_{111}x_1^3 + w_{112}x_1^2x_2 + w_{122}x_1x_2^2 + w_{222}x_2^3, \quad x_0 = 1 \end{aligned} \tag{2.18}$$

Owing to the nonlinearity of the neural unit with CSO, the generated discriminant curves were polynomials. One discriminant curve was able to categorize the two classes efficiently. Table 2.5 gives details of the initial and final values of synaptic weights.

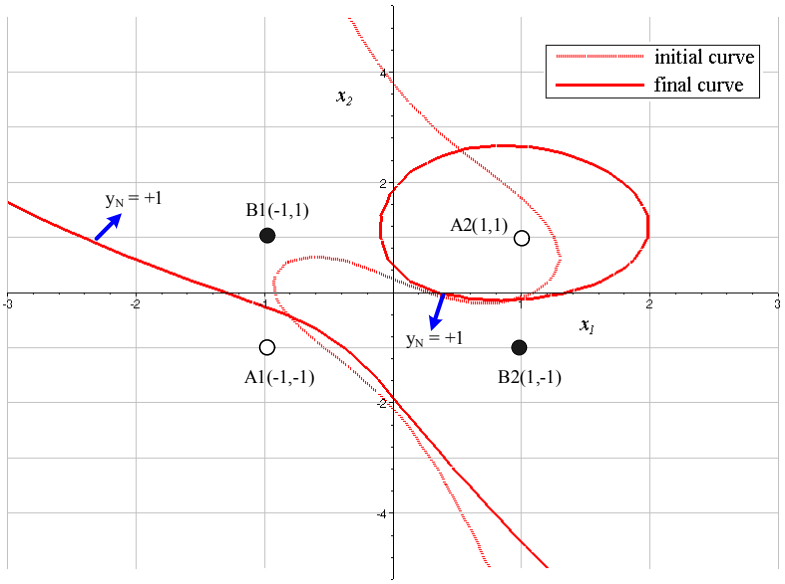
Table 2.5 Initial and final synaptic weights in a neural unit with CSO for **XOR** operation

Case		Synaptic weights									
		w_{000}	w_{001}	w_{002}	w_{011}	w_{012}	w_{022}	w_{111}	w_{112}	w_{122}	w_{222}
I	Initial	0.2548	-0.9144	-0.9553	-0.4055	-0.8561	-0.2457	0.9143	0.5029	0.8288	0.1262
	Final	0.3967	-1.1713	-0.8464	-0.2637	-2.5382	-0.1039	0.6575	0.6118	0.5720	0.2351
II	Initial	0.5067	-0.4188	0.8796	0.4825	-0.6803	-0.5687	0.4789	0.9668	-0.5397	0.5077
	Final	0.3762	-0.2396	0.0951	0.3520	-2.5422	-0.6992	0.6582	0.1823	-0.3605	-0.2769
III	Initial	-0.9160	-0.6001	-0.3219	0.0908	-0.9981	0.6703	0.5833	0.8747	0.9249	0.8266
	Final	-0.8546	-0.8834	-0.7815	0.1522	-2.5389	0.7316	0.3000	0.4151	0.6416	0.3670

When the error is less than 0.05 after several iteration times, the simulation was stopped. Three cases of discriminant curves from the neural unit with CSO were generated. The geometric representation is shown in Fig. 2.9. The discriminant curves of the neural unit with CSO were also determined by the initial values of the synaptic weights.

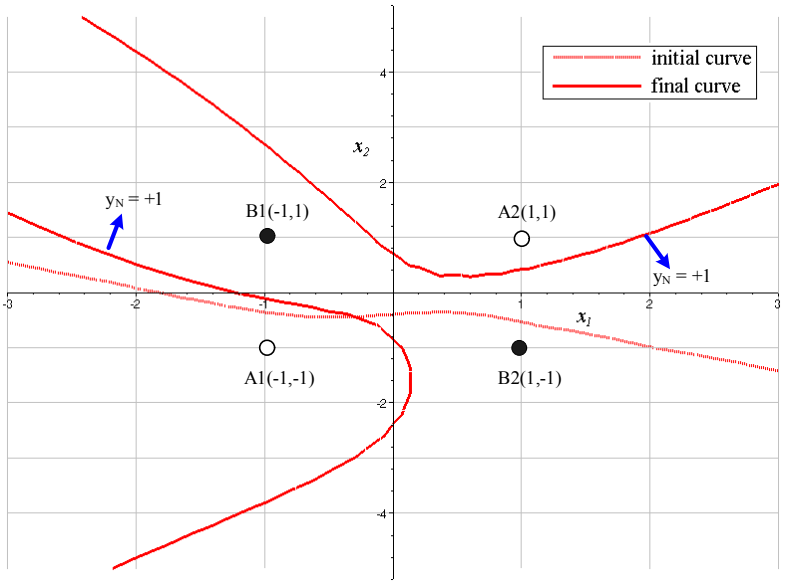
From the simulation results, it is shown that a single neural unit with QSO or CSO was able to classify the nonlinear discriminant classes of **XOR** logic problem. The final discriminant curves were generated corresponding to the initial synaptic weights. The initial and final curves from the initial and final synaptic weights are illustrated in the figures. From the figure, it is discovered how the initial curves are revolutionized to the final curves for classifying the nonlinear separable problem successfully.

In general, there may be many different neural classifiers for the **XOR** logic classification, but these neural units with QSO and CSO create more efficient discriminant curves. However, the neural units with HOSO may have the difficulty of consuming more time to solve the given problem as these neural units with HOSO have more complicated polynomial equations. Thus, the order of the synaptic operation should be considered in relation to the given problems.



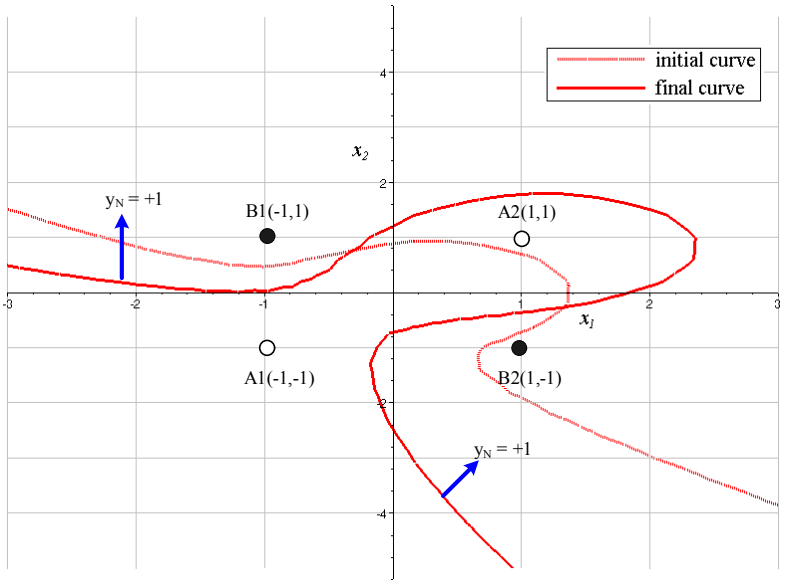
(a) Case I

$$\begin{cases} \text{initial curve}: v = 0.2548 - 0.9144x_1 - 0.9553x_2 - 0.4055x_1^2 - 0.8561x_1x_2 \\ \qquad\qquad - 0.2457x_2^2 + 0.9143x_1^3 + 0.5029x_1^2x_2 + 0.8288x_1x_2^2 + 0.1262x_2^3 \\ \text{final curve}: v = 0.3967 - 1.1713x_1 - 0.8464x_2 - 0.2637x_1^2 - 2.5382x_1x_2 \\ \qquad\qquad - 0.1039x_2^2 + 0.6575x_1^3 + 0.6118x_1^2x_2 + 0.5720x_1x_2^2 + 0.2351x_2^3 \end{cases}$$



(b) Case II

$$\begin{cases} \text{initial curve}: v = 0.5067 - 0.4188x_1 + 0.8796x_2 + 0.4825x_1^2 - 0.6803x_1x_2 \\ \qquad\qquad - 0.5687x_2^2 + 0.4789_1^3 + 0.9668x_1^2x_2 - 0.5397x_1x_2^2 + 0.5077x_2^3 \\ \text{final curve}: v = 0.3762 - 0.2396x_1 + 0.0951x_2 + 0.3520x_1^2 - 2.5422x_1x_2 \\ \qquad\qquad - 0.6992x_2^2 + 0.6582x_1^3 + 0.1823x_1^2x_2 - 0.3605x_1x_2^2 - 0.2769x_2^3 \end{cases}$$



(c) Case III

$$\begin{cases} \text{initial curve}: v = -0.9160 - 0.6001x_1 - 0.3219x_2 + 0.0908x_1^2 - 0.9981x_1x_2 \\ \qquad\qquad + 0.6703x_2^2 + 0.5833_1^3 + 0.8747x_1^2x_2 + 0.9249x_1x_2^2 + 0.8266x_2^3 \\ \text{final curve}: v = -0.8546 - 0.8834x_1 - 0.7815x_2 + 0.1522x_1^2 - 2.5389x_1x_2 \\ \qquad\qquad + 0.7316x_2^2 + 0.3000x_1^3 + 0.4151x_1^2x_2 + 0.6416x_1x_2^2 + 0.3670x_2^3 \end{cases}$$

Figure 2.9 Neural **XOR** classification with CSO and the discriminant curves from the polynomial equations of synaptic operation in different cases

2.5 Generalization of Neural Units with Higher-Order Synaptic Operations (HOSO)

With the derivation of the neural units with QSO and CSO, this concept can be generalized and formulated in mathematical representation for the neural units with HOSO. The synaptic and somatic operations of the neural unit with Z^{th} order synaptic operation can be written as

$$v = \sum_{i_1=0}^{n} \sum_{i_2=i_1}^{n} \cdots \sum_{i_Z=i_{Z-1}}^{n} w_{i_1 i_2 \cdots i_Z} x_{i_1} x_{i_2} \cdots x_{i_Z} \tag{2.19}$$

and

$$y_N = \Phi[v] \in \Re^1 \tag{2.20}$$

The LAR of the generalized neural unit with HOSO should be modified and developed for their respective synaptic weights. The LAR of the synaptic weight ($w_{i_1 i_2 \cdots i_Z}(k)$) in the neural units with HOSO can be expressed in discrete time as

$$w_{i_1 i_2 \cdots i_Z}(k+1) = w_{i_1 i_2 \cdots i_Z}(k) + \Delta w_{i_1 i_2 \cdots i_Z}(k) \tag{2.21}$$

Additionally, the adjustment of the synaptic weight $\Delta w_{i_1 i_2 \cdots i_Z}(k)$ in the generalized neural units with HOSO is derived as

$$\begin{aligned}
\Delta w_{i_1 i_2 \cdots i_Z}(k) &= -\mu \frac{\partial J[e(k)]}{\partial w_{i_1 i_2 \cdots i_Z}(k)} \\
&= \mu E\left[e \frac{\partial y_N(k)}{\partial w_{i_1 i_2 \cdots i_Z}(k)} \right] \\
&= \mu E\left[e(k) \left\{ \Phi'[v(k)] \frac{\partial v(k)}{\partial w_{i_1 i_2 \cdots i_Z}(k)} \right\} \right] \\
&= \mu E\left[e(k) \Phi'[v(k)] \frac{\partial \left(\sum_{i_1=0}^{n} \sum_{i_2=i_1}^{n} \cdots \sum_{i_Z=i_{Z-1}}^{n} w_{i_1 i_2 \cdots i_Z}(k) x_{i_1}(k) x_{i_2}(k) \cdots x_{i_Z}(k) \right)}{\partial w_{i_1 i_2 \cdots i_Z}(k)} \right] \\
&= \mu E[e(k) x_{i_1}(k) x_{i_2}(k) \cdots x_{i_Z}(k) \Phi'\{v(k)\}]
\end{aligned} \tag{2.22}$$

where $E[\bullet]$ is an expectation function.

Finally, the adapted synaptic weight of the generalized neural unit with HOSO is then calculated as

$$w_{i_1 i_2 \cdots i_Z}(k+1) = w_{i_1 i_2 \cdots i_Z}(k) + \mu E[e(k) x_{i_1}(k) x_{i_2}(k) \cdots x_{i_Z}(k) \Phi'\{v(k)\}] \tag{2.23}$$

The basic structure of the neural unit with HOSO is shown in Fig. 2.10.

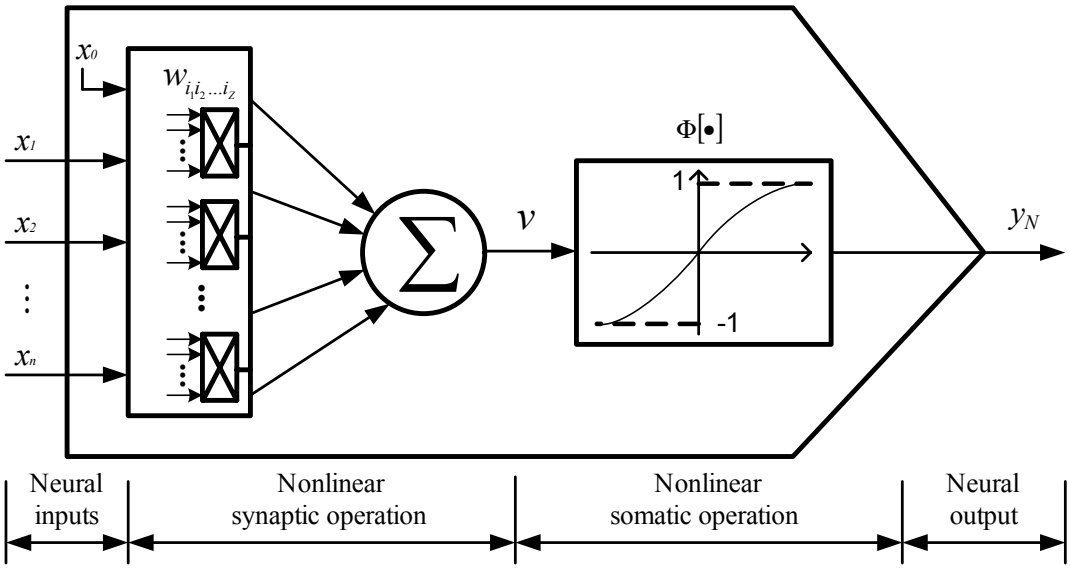


Figure 2.10 The schematic view of the generalized neural unit with HOSO with complex incorporation of multiple synaptic weights and neural inputs

From the generalized representation of neural units with HOSO, the neural units with LSO, QSO, and CSO can be derived. The numbers of the inputs that correspond to a neural weight decide the order of the synaptic operation.

2.6 Summary

The novel neural units with HOSO have been introduced in this chapter, demonstrating their profound potential. The major difference between the neural units with LSO and HOSO is in the order of synaptic operation. The higher-order association of the synaptic operation with neural inputs supplements the neural units with HOSO getting more information about of the neural inputs and making the neural system more efficient and sensitive. The correlation between the neural inputs is handled and taken care of by the synaptic operation. One synaptic weight affected by the multiple inputs calls for a nonlinear relation. Clearly the neural unit with LSO is not able to compute this nonlinearity due to the linear structural property. The superior capability of the neural units with HOSO comes from the nonlinearity of the synaptic operation. The given **XOR** logic operation with neural structure had been unresolved for several decades due to its nonlinear property. The existing methods for solving the **XOR** problem apply several layers of linear neurons stacked in layers. However, one neural unit with HOSO has the capability to replace the complex neural networks consisting of several neural units with LSO. These neural units with HOSO may offer great effects to resolve the numerous enigmatic intricacies just as the **XOR** problem. Despite the enormous improvement of the neural units with HOSO, the complexity of the synaptic weights of a neural unit with HOSO is inevitable. This complex computation of the neural units with HOSO may consume more time to achieve the required neural output. Hence, the order of the synaptic operation of the neural units with HOSO should be selected conforming to the complexity of the problem to get optimal and effective solution.

Chapter 3

Dynamic Neural Units and Neural Networks with Cubic Synaptic Operation (CSO)

3.1 Introduction

The neural units described in the previous chapters consist of a combination of two computational operations. One is the synaptic operation and the other is the somatic operation. These two operations are fundamental processes that add the weighted inputs and fire out the sum of inputs when the total result of the synaptic operation exceeds a certain threshold. These neural structures are referred to as static or feedforward structures [8]. In these static or feedforward neural structures, neither feedback connections nor memories are considered. The outputs of these static or feedforward neural structures are determined by the current inputs and values of the synaptic weights.

Biological neural systems are composed of dynamic recurrent connections and dynamic memories [10, 11, 17]. In the literature, feedback is broadly used in the system to make it a dynamic structure. The dynamic structure makes the system response relatively insensitive to external disturbances and internal variations in the system's parameters. Thus, feedback is very advantageous for the system when the system faces unpredictable disturbances [21]. This feedback can be implemented in two ways for the neural units and networks. When it is built within the neural unit, this feedback is referred to as local feedback. The other type of feedback is called global feedback and encircles the entire networks in which the neural units are static. In comparison to local feedback, global

feedback implementation provides a more profound effect on the networks [25]. A neural unit with local feedback is called a dynamic neural unit [18, 19]. In this chapter, the current dynamic structure of a dynamic neural unit is extended to the neural unit with CSO. The new structure is regarded as a dynamic neural unit with CSO. Furthermore, the present recurrent (feedback) NNs with global feedback can be extended with the neural units with CSO. This introduces neural networks built up of the neural unit with CSO. The recurrent networks are regarded as nonlinear dynamic systems [24]. Thus, the evolved networks built of neural units with CSO and dynamic structure are regarded as the novel *dynamic neural networks* (DNNs).

A review of the architectural details of the dynamic structure is described in Section 3.2. The modification of a dynamic neural unit with the neural unit with CSO is presented with the LAR in Section 3.3. The NNs consisting of neural units with CSO with the modified back-propagation are discussed in Section 3.4. Section 3.5 introduces an industrial motion control system as an application for the model identification. The static neural unit and the dynamic neural units with CSO are used as identifiers of the system. Finally, the properties of the dynamic structure in the dynamic neural unit are also summarized after the simulation studies in Section 3.5.

3.2 A Review of the Architectural Details of the Dynamic Neural Structure (DNS)

A biological neuron contains dynamic features such as feedback, memory, and delay used in the processing of the input signals are shown in Fig. 3.1. The input signals are associated in the synapse and if the combined signal exceeds a threshold, the neuron fires

the output. Meanwhile, the merged signal is fed back to the synapse and delayed during the synaptic and somatic procedures.

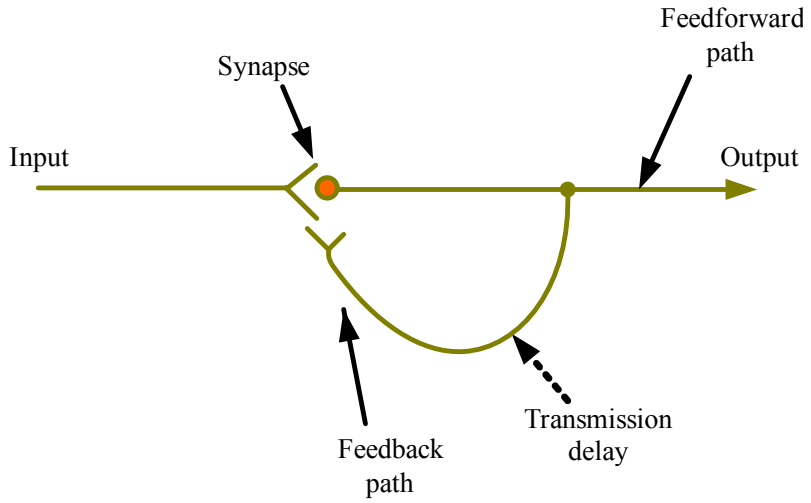


Figure 3.1 Biological neuron with reverberating signals

Based on the biological model, an artificial neural structure, called the *dynamic neural structure* (DNS), can be developed. This DNS is assumed to have feedback with a delay. This dynamic structure is placed between the synaptic operation and the somatic operation of the neural unit emulating the biological neuron shown in Fig. 3.1. The mathematical expression of the DNS in a neural unit which has the synaptic and somatic operations is expressed in Eqn. (3.1). Figure 3.2 describes the schematic view of this dynamic structure.

$$G_{tf}(k) = \frac{d(k)}{v(k)} = \frac{1}{1 + bz^{-1}} = \frac{z}{z + b} \tag{3.1}$$

where $v(k)$ is the synaptic output, $d(k)$ is the output of the DNS as well as the input to the somatic operation, b is the feedback weight of the DNS, z^{-1} is the unit delay operator, and *k* represents discrete-time.

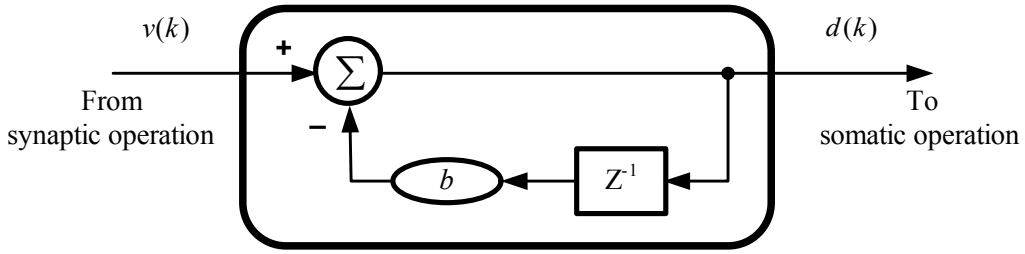


Figure 3.2 Basic structure of DNS with a feedback weight and a delay in a neural unit

The output of the DNS is derived according to Eqn. (3.1) as follows

$$v(k) = d(k) + bd(k)z^{-1} \tag{3.2a}$$

$$v(k) = d(k) + bd(k-1) \tag{3.2b}$$

or

$$d(k) = v(k) - bd(k-1) \tag{3.3}$$

The DNS is chosen for implementation with the neural unit with CSO from the group of neural units with HOSO due to its superior performance. A neural unit with CSO and DNS is called a dynamic neural unit with CSO and is illustrated in Fig. 3.3.

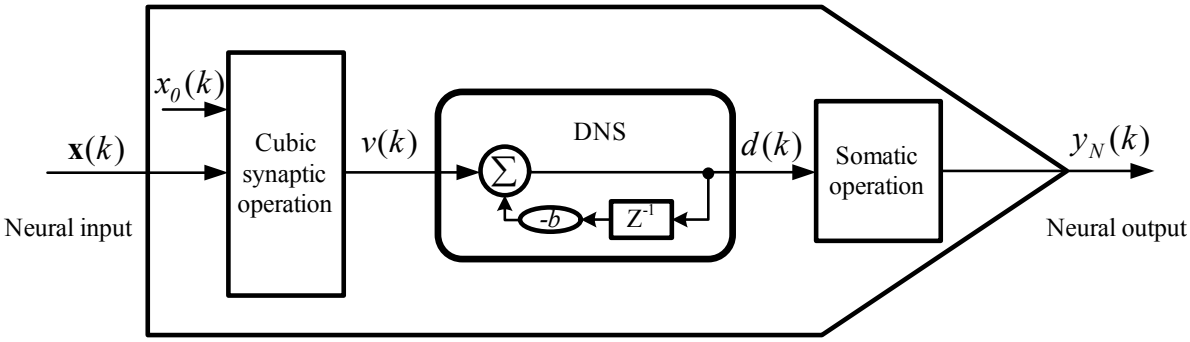


Figure 3.3 The structure of the dynamic neural unit with CSO

3.3 Learning and Adaptation Rule (LAR) of the Dynamic Neural Unit with CSO

The concept of LAR was introduced in the previous chapter. It helps the neural unit to learn and adapt the given information using the error between the neural output and the desired output. The LAR also influences the determination of the feedback weight which reduces the error for the optimal neural output. The adaptation of the feedback weight of the DNS is explained in this section.

The error between the neural output and the desired output is defined in Eqn. (3.4). If the error is reduced to an infinitesimally small value as the number of learning iterations increases, the learning scheme is said to be convergent [20].

$$\begin{aligned} e(k) &= y_d(k) - y_N(k) \\ \lim_{k\to\infty} e(k) &= 0 \end{aligned} \tag{3.4}$$

The error function $J[\bullet]$, called the least mean square, is used to discern the global minimum for optimal neural outputs. The feedback weight of the DNS is adapted to make

the dynamic structure more effective and optimal for the neural unit, although this may not guarantee global optimization [12]. This error function is represented as

$$J[b(k)] = \frac{1}{2} E[e^2(k;b)] \tag{3.5}$$

The DNS feedback weight is adapted based on Eqn. (3.6) in order to minimize the error

$$b(k+1) = b(k) - \mu \nabla_b J[b(k)] \tag{3.6}$$

where $b(k+1)$ and $b(k)$ are parameters at the *(k+1)*th and *(k)*th sampling instants, μ is the learning rate, and $\nabla_b J[b(k)]$ is the gradient of $J[\bullet]$ with respect to the DNS feedback weight *b*.

The error function is then rewritten as

$$\nabla_b J[b(k)] = \frac{\partial J[b(k)]}{\partial b(k)} \tag{3.7}$$

From the definition of $J[\bullet]$ and $e(k)$, Eqn. (3.7) is derived as

$$\frac{\partial J[b(k)]}{\partial b(k)} = \frac{1}{2} E\left[\frac{\partial (y_d(k) - y_N(k))^2}{\partial b(k)} \right]$$

$$= E\left[e(k) \left\{ \frac{\partial (y_d(k) - y_N(k))}{\partial b(k)} \right\} \right]$$

$$= E\left[e(k) \left\{ -\frac{\partial y_N(k)}{\partial b(k)} \right\} \right]$$

$$= E\left[e(k)\left\{ -\frac{\partial \Phi[d(k)]}{\partial b(k)} \right\} \right]$$

$$= E\left[e(k)\left\{ -\frac{\partial \Phi[d(k)]}{\partial d(k)} \frac{\partial d(k)}{\partial b(k)} \right\} \right]$$

$$= E\left[e(k)\left\{ -\Phi'[d(k)] \frac{\partial d(k)}{\partial b(k)} \right\} \right]$$

$$= E\left[-e(k)\left\{ \Phi'[d(k)] S_b(k) \right\} \right] \tag{3.8}$$

where $\Phi[\bullet]$ is the nonlinear activation function for the somatic operation, $d(k)$ is the output of the DNS, and $S_b(k)$ is defined as a sensitivity signal [8].

Hence, the LAR of the DNS feedback weight is rewritten as

$$b(k+1) = b(k) + \mu E\left[e(k)\left\{ \Phi'[d(k)] S_b(k) \right\} \right] \tag{3.9}$$

The sensitivity signal is derived from Eqn. (3.3) as

$$\begin{aligned} S_b(k) &= \frac{\partial d(k)}{\partial b(k)} = \frac{\partial}{\partial b(k)} [v_a(k) - b(k) d(k-1)] \\ &= -d(k-1) \end{aligned} \tag{3.10}$$

The block diagram for the sensitivity signal is shown in Fig. 3.4 Also, the entire scheme of the LAR for the neural unit with CSO, DNS and sensitivity is shown in Fig. 3.5.

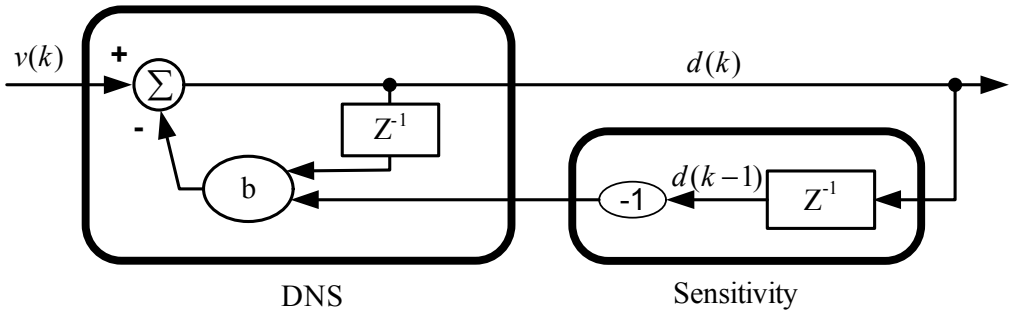


Figure 3.4 DNS with sensitivity to obtain an optimal feedback weight

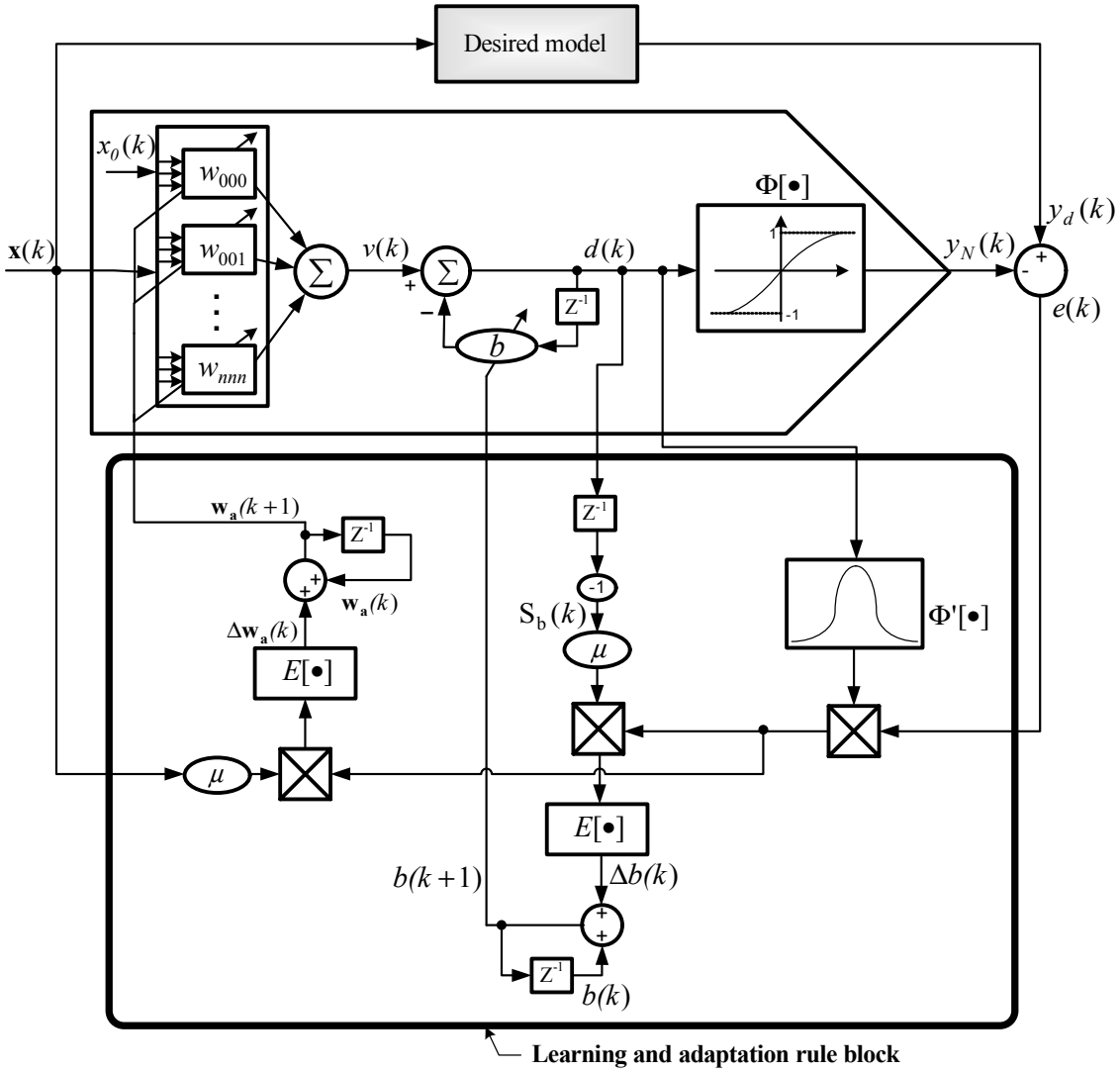
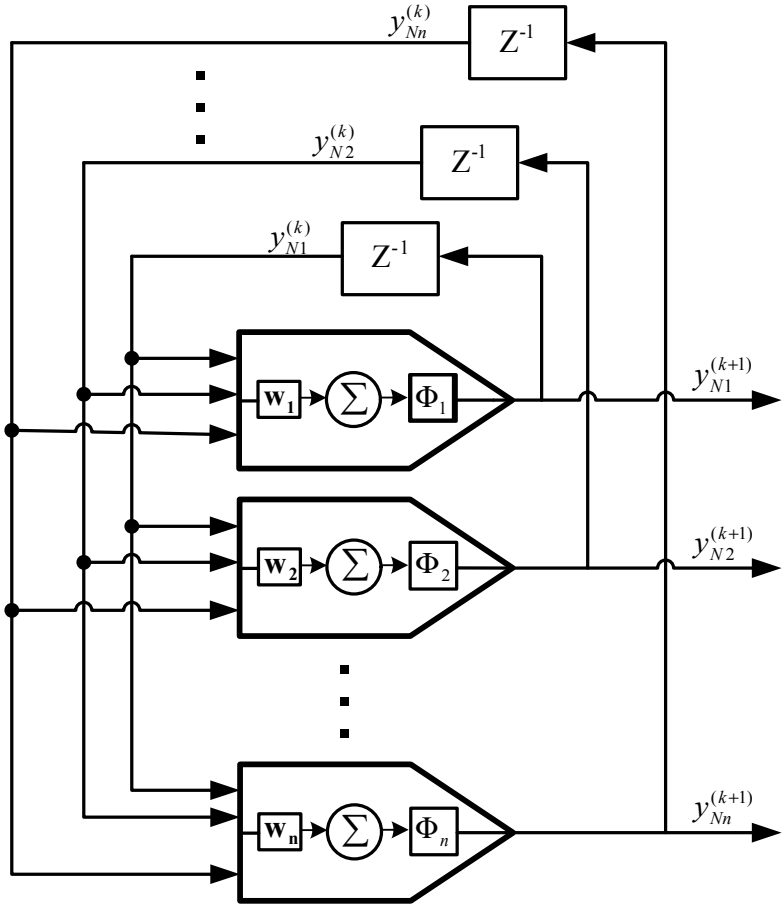


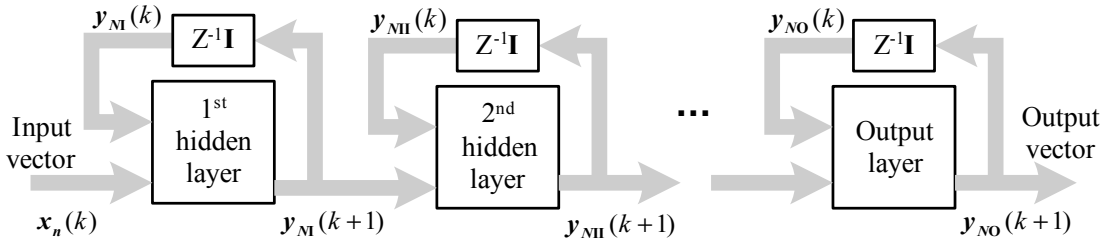
Figure 3.5 The implementation scheme of the LAR for the dynamic neural unit with CSO and sensitivity

3.4 Dynamic Neural Networks (DNNs) Consisting of Neural Units with CSO

In the literature, neural networks with one or more feedback loops are referred to as recurrent networks, and networks with the global feedback as dynamic neural networks. Global feedback can make the networks perform better and has the potential of reducing the memory requirement significantly [25]. Figure 3.6 shows the architecture of global feedback with multiple hidden layers and one output layer. In this section, a neural unit with CSO is used as the basic element of the DNNs.



(a) A recurrent layer



(b) Multilayer with global feedback (**I** : identity matrix **)**

Figure 3.6 Recurrent multilayer neural networks

3.4.1 Back-Propagation Through Time (BPTT)

A significant breakthrough in the field of neural networks was the discovery of the *back-propagation* (BP) algorithm, an integral function in multilayered networks, which makes the synaptic weights learn and adapt to the surroundings. This algorithm is based on the error-correction learning rule. The *back-propagation through time* (BPTT) algorithm is an extension of BP for a recurrent multilayer network. The BPTT consists of two procedures which are *epochwise back-propagation through time* (EBPTT) and *truncated back-propagation through time* (TBPTT) [25]. The BPTT was developed with linear neural structures by Williams and Peng [25, 26]. In this section, the BPTT with the neural units with CSO is used by replacing the structures of BPTT with that of the neural unit with CSO. The resultant modified EBPTT is introduced in this section.

3.4.1.1 Epochwise Back-Propagation Through Time (EBPTT) with Neural units with CSO

The term ‘epoch’ of the EBPTT means the period or process during which the entire data are processed from the input layer to the output layer. The synaptic weights of the neural

units with CSO are adapted after every epoch with the EBPTT algorithm. The modified algorithm proceeds as follows:

- Given the epoch, the total error function (cost function) from the output layer is defined as

$$\mathrm{E}_{total}(n_0, n_1) = \frac{1}{2}\sum_{n=n_0}^{n_1}\sum_{j\in \mathrm{A}} e_j^2(n) \tag{3.11}$$

  where A is the set of indices j, $e_j(n)$ is the error signal at the output, and n_0 denotes the start time of an epoch, and n_1 denotes its end time.

- A single forward pass of the data through the network for the interval (n_0, n_1) is performed. The complete record of the input data, synaptic weights of the networks, and desired responses over this interval is saved.
- A single backward pass for the prior neural layer errors over this past record is performed to compute the values of the local gradients of the total error

$$\delta_j(n) = -\frac{\partial \mathrm{E}_{total}(n_0, n_1)}{\partial v_j(n)} \tag{3.12}$$

  for all $j \in \mathrm{A}$ and $n_0 < n \le n_1$. The local gradient is executed as follows

$$\delta_j(n) = \begin{cases} \Phi'[v_j(n)]e_j(n) & \text{for } n = n_1 \\ \Phi'[v_j(n)]\left[e_j(n) + \sum_{k\in \mathrm{A}} w_{jjj}^{(k)}\delta_k(n+1)\right] & \text{for } n_0 < n < n_1 \end{cases} \tag{3.13}$$

where $\Phi'[\bullet]$ is the derivative of a nonlinear activation function, $v_j(n)$ is the synaptic operation of the *j*-th neural unit with CSO. $w_{jjj}^{(k)}$ is a synaptic weight, whose index number is *jjj* of the *k*-th neural unit with CSO, and computed by the output of the *j*-th neural unit with CSO. The synaptic weight affects the *j*-th neural unit with CSO.

- Once the computation of back-propagation has been performed to time n_0+1, the following adjustment is applied to the synaptic weight $w_{abc}^{(j)}$ of the *j*-th neural unit with CSO:

$$\begin{aligned}\Delta w_{abc}^{(j)} &= -\mu \frac{\partial \mathrm{E}_{total}(n_0, n_1)}{\partial w_{abc}^{(j)}} \\ &= \mu \sum_{n=n_0+1}^{n_1} \delta_j(n) x_i(n-1)\end{aligned} \tag{3.14}$$

where μ is learning rate and $x_i(n-1)$ is the input applied to the *i*-th synapse of *j*-th neural unit at time $(n-1)$.

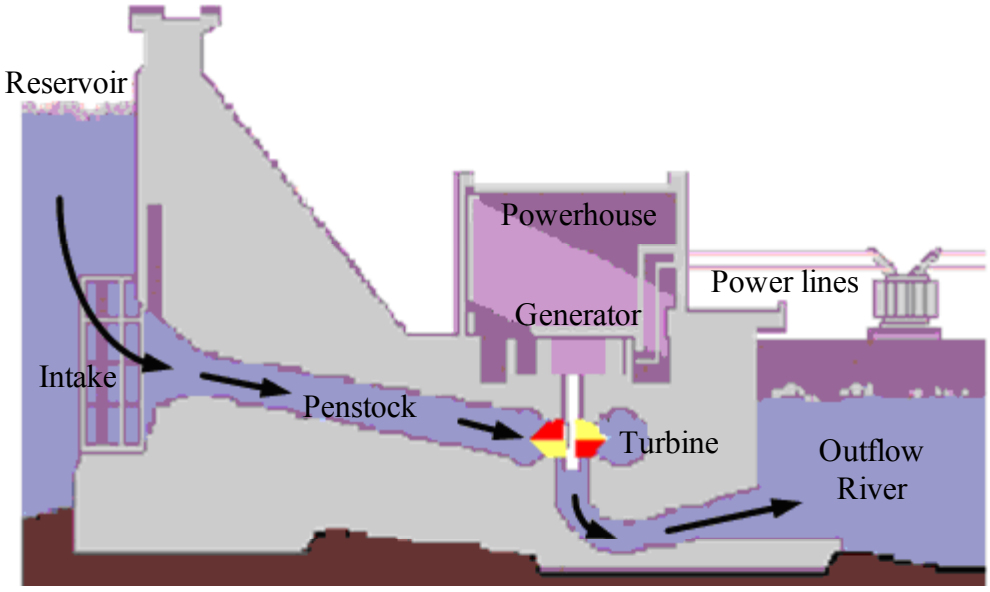
3.5 The Dynamic Neural Unit with CSO for the Identification of Nonlinear Systems

NNs are now being used as a function to identify linear and nonlinear dynamic systems in engineering. NNs have the potential of developing attributes such as parallelism, adaptability, robustness, and the inherent ability to handle nonlinearity. The NNs have established their usefulness in such fields as function mapping, pattern recognition, and image processing. However, dynamic function mapping, including the structural dynamic

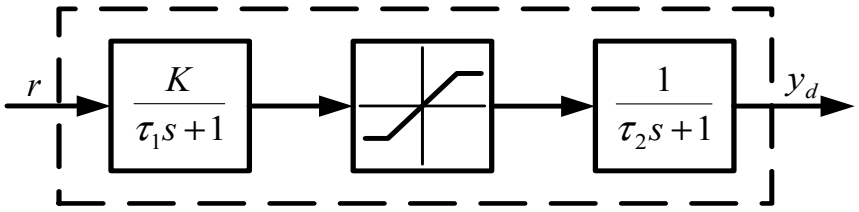
model identification, remains a challenge for neural network applications [43]. In this section, the novel dynamic neural unit with CSO for the application in model identification is presented. Owing to its properties such as higher-order computation and dynamic structure, the dynamic neural unit with CSO has the capability of identifying higher-order systems. For simulation purposes, a nonlinear second-order system is used as the model system. The performance of the static neural unit with CSO for model identification is compared with the performance of the dynamic neural unit with CSO.

3.5.1 Simulated Nonlinear System

Figure 3.7 shows a nonlinear system being used for the simulation. A typical hydroelectric power plant is given as an example [43]. These kinds of systems are very common in industrial processes, especially in motion control systems and in any typical example of nonlinear systems. This contains a cascade of first-order systems with a saturation element between them.



(a) A typical hydroelectric power plant



(b) The block diagram of the hydroelectric power plant

Figure 3.7 A nonlinear system for the simulation studies

3.5.2 Simulation studies

For this simulation, one static neural unit with CSO and one dynamic neural unit with CSO were used as model identifiers instead of using NNs. The higher-order computation of the synaptic operation in these neural structures aids the neural units to execute the impressive performance of these NNs. The input of the systems was set to change the amplitude at every 100 iteration interval as shown in Fig. 3.8.

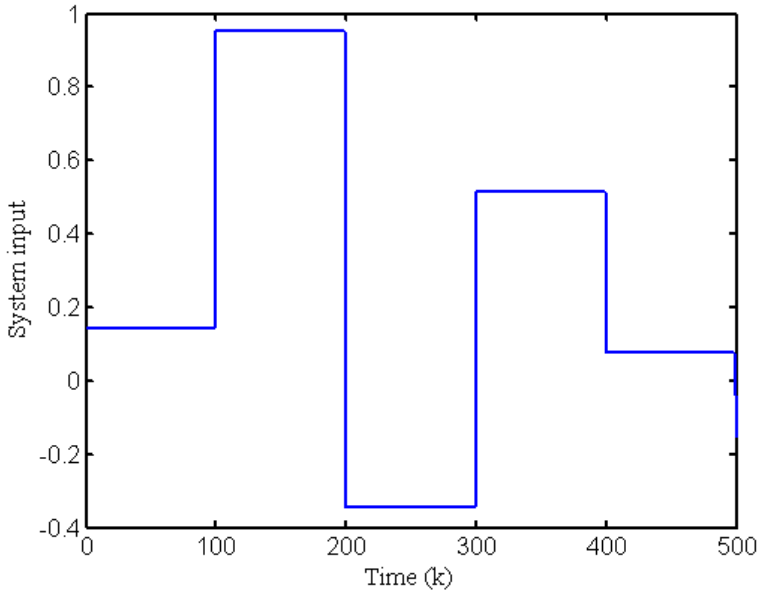
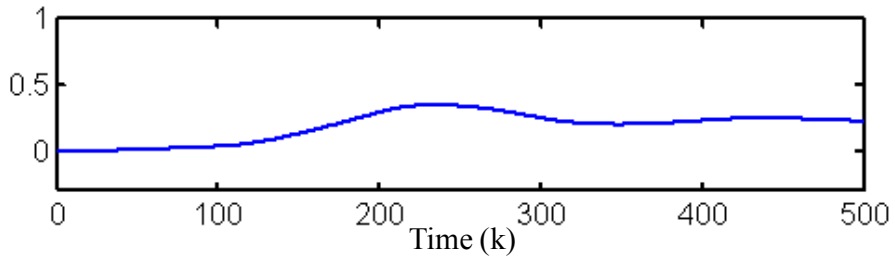
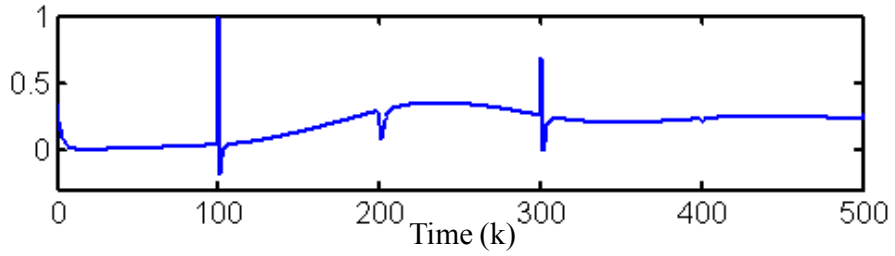


Figure 3.8 System input for the simulation

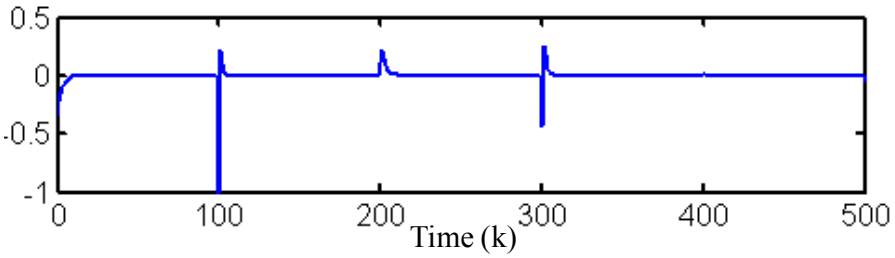
For the power plant, the parameters were $\tau_1 = 1$, $\tau_2 = 1$, $K = 1$, and the saturation limits = 1,-1. Figure 3.9 shows the outputs of plant and the dynamic neural unit with CSO. The learning rate μ of the neural unit was chosen to be 0.5. The block diagram of the plant was converted to discrete system with sampling time 0.01. The identification error between the plant and the neural unit is also described in the figures. As seen in the figures, as the identification error decreases, the neural unit is able to identify the plant with minimal error.



(a) The plant output



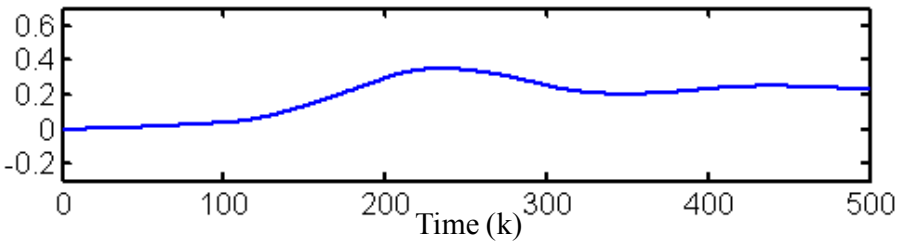
(b) The output of the dynamic neural unit with CSO



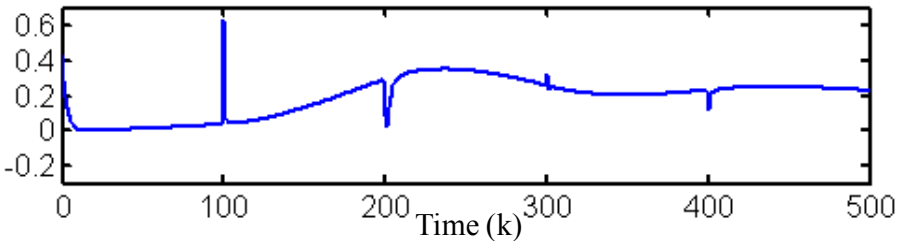
(c) The error between the plant output and neural output

Figure 3.9 The model identification of the plant with a dynamic neural unit with CSO

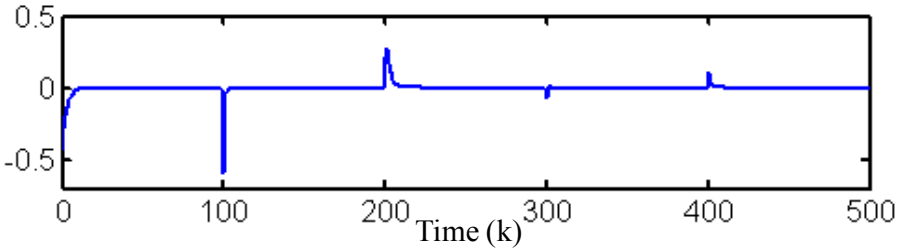
After using the dynamic neural unit identifier, a static neural unit with CSO was used to identify the given system. Figure 3.10 describes the outputs of the plant and the neural unit and identification error. The learning rate of the neural unit was again chosen to be 0.5. The identification error was reduced, and the static neural unit with CSO could identify the plant. This error can be compared with the error from the dynamic neural unit.



(a) The plant output



(b) The output of the static neural unit with CSO



(c) The error between the plant output and neural output

Figure 3.10 The model identification of the plant with a static neural unit with CSO

Both of the static and dynamic neural units with CSO were able to identify the system after a small number of iteration times. This proves the capability of the higher computation of the neural units with HOSO. The dynamic neural unit had higher overshoot at the beginning, but the error became less than the error of the static neural unit afterwards as shown in Figs.3.9 and 3.10. Hence, it is concluded that the dynamic structure, which has a memory attribute in the neural unit, influences the neural unit to perform more effectively.

3.6 Summary

In this chapter, the DNS has been described based on the design of the biological neuron. The basic motivation of the dynamic structure is that the biological neurons have many recurrent connections and also have the function of memory. The recurrent links and memory are emulated as the feedback in the dynamic neural unit with CSO. Additionally, the basic learning scheme, back-propagation, was illustrated for the DNNs.

With the help of computer simulations, the model identification was described with both the static and the dynamic neural units with CSO. From the simulation results, the dynamic and static neural units with CSO showed the excellent results. According to the simulation results, it is proved that the memory causes the neural structure to have some potential and influence the performance. In the literature, NNs composed of many neural units are commonly used for model identification. However, one neural unit with HOSO is capable of performing the task of the NNs with better results. Hence, the neural networks with an even small number of the neural unit with HOSO may present advanced efficiency.

Chapter 4

Edge Detection with HOSO Neural Units

4.1 Introduction

Vision is considered to be the richest of all the sensory processes in a human system because of its diverse informative nature and the intensity of the vision sensors as compared to other physical senses [30]. This robust but sensitive biological organ has been emulated and used in machine vision applications. One of the applications of machine vision emulations of the biological vision is edge detection. The edge is detected by different intensities of objects. Edge detection of images is a very significant aspect of computer vision, remote sensing and image analysis [27]. Edge detection is usually considered as a subjective task. The eye is illustrated in Fig. 4.1. From the figure, it is clear that the eye consists of numerous nerve cells which receive many light stimuli. The light to the eye from the surroundings interferes with each stimulus and may be interpolated by other stimuli to project a clear picture on the retina. These interference and interpolations are considered as some of the properties of the neuron. The neural unit does emulate this biological property, and hence the vision application with the neural structures becomes preferable. As proved in the previous chapter, the neural units with HOSO are capable of solving more complicated problems. In this thesis, a new scheme of edge detection is introduced with the static neural unit with CSO.

In this chapter, Section 4.2 shows the novel neural edge detector with CSO. In this section, differential operators are introduced. The differential operator represents the

biological vision system in mathematical terms [30]. The comparison of the neural edge detectors with different synaptic operations is shown in the succeeding sections.

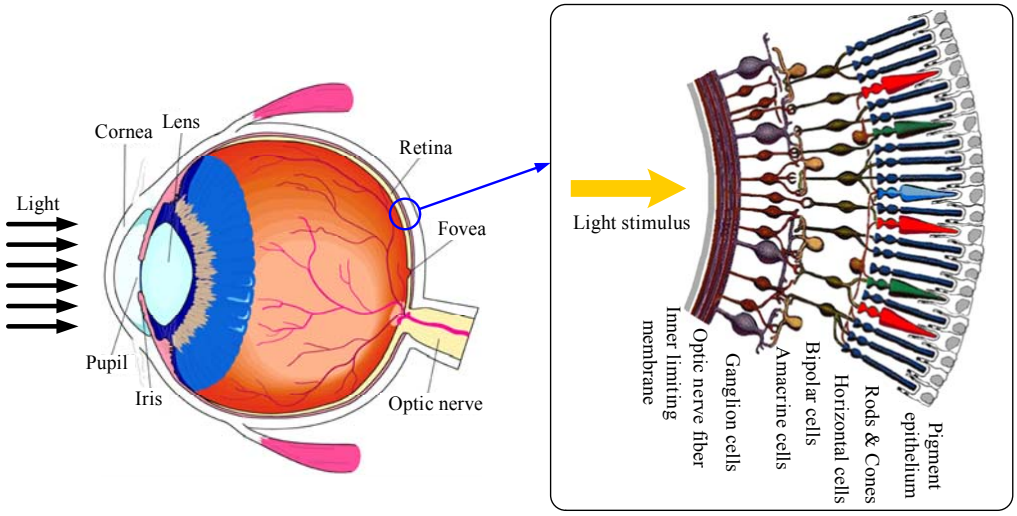


Figure 4.1 The structure of eye and retina

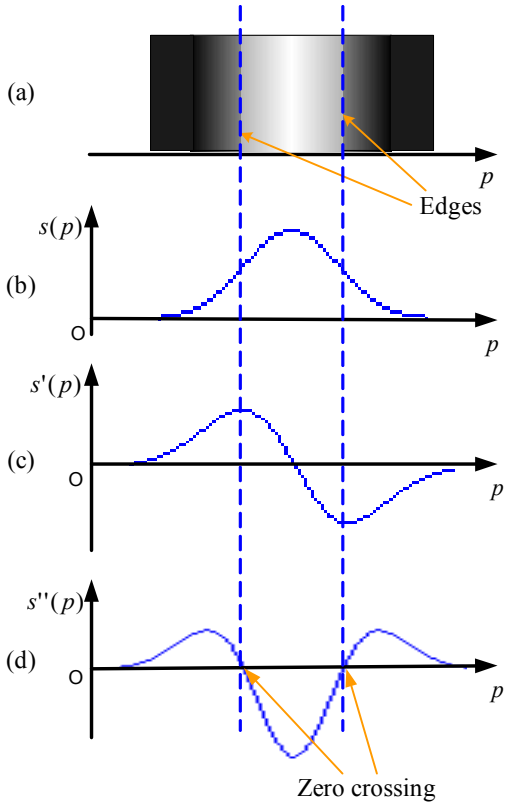
4.2 Neural Edge Detection

Experimental studies have found that the biological neural cells respond to specific visual stimulus patterns projected onto the photoreceptors, or the rods and cones, in the retina. The ganglion cells in the retina respond to the presence of the light stimuli. This procedure recognizes the edge of the object by different contrast and brightness of light [29]. The neural edge detectors are designed to reproduce this biological visual phenomenon. An edge exists if there is a large difference in the contrast or brightness between pixels of the given image. In order to analyze the contrast or brightness, a color spatial function corresponding to the intensity of the color is introduced as shown in Fig. 4.2. The second-order differentiation of the color spatial function detects the zero crossing points, which represent the edge in the image. The color intensity around the

edge is distributed similar to the Gaussian function. Thus, the Gaussian function is applied as the color spatial function and also for defining and detecting the edges. The Gaussian function is defined as

$$Ga(x) = e^{-\alpha x^2} \tag{4.1}$$

where α is the slope rate of the Gaussian function.



(a) pixels of a image,

(b) color spatial function,

(c) first differentiation of color spatial function, and

(d) second differentiation of color spatial function

Figure 4.2 A simple illustration of color spatial function to detect the edge of the image

4.2.1 Differential Operators

When the retina is stimulated by a light incentive, the photoreceptors, which are nothing but rods and cones, produce signals, and the ganglion cells accumulate the signals in order to process the stimuli [30]. The response of the ganglion cells can be expressed in mathematical terms as

$$G(c) = \sum_{m=1}^{q} f_m[R(c)] \tag{4.2}$$

where $G(c)$ is the ganglion signal, $R(c)$ is the receptor signal, c is the spatial coordinate, q represents the order of differential operation and $f_m[\bullet] = \frac{dR(c)}{dc}$.

The correlation between photoreceptors and ganglion cells is shown in Fig. 4.3.

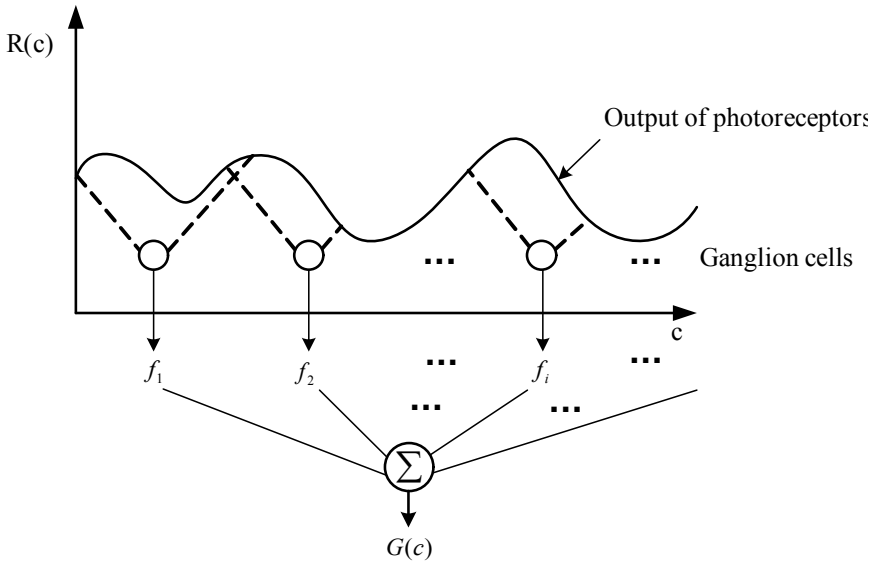


Figure 4.3 Correlation between the receptor signal and the ganglion signal

The differentiation function $f_m[\bullet]$ of the receptor signal can be expressed in terms of sampling spatial coordinate Δc. Equation (4.1) then becomes the first-order differential operation expressed as

$$G(c) = \sum_{m=1}^{1} f_m[R(c)] = \frac{R(c+\Delta c) - R(c)}{\Delta c} = \frac{1}{\Delta c}\big(R(c+\Delta c) - R(c)\big) \tag{4.3}$$

This transference is shown in Fig. 4.4.

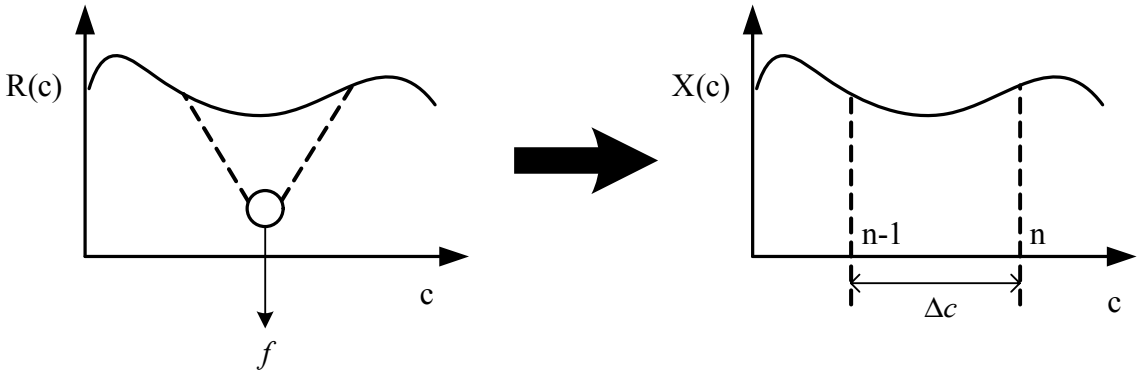


Figure 4.4 Transference from signal domain to sampling spatial coordinate domain

Equation (4.4) shows the differential operator with respect to the spatial coordinate instance Δc

$$D_1 = \frac{1}{\Delta c}\big(X_n - X_{n-1}\big) \tag{4.4}$$

where D_1 is the first-order differential operator and X_n represents the input of the differential operator at the *n*-th spatial coordinate.

The first-order differential operation can be illustrated with the differential operator and the impulse response shown in Fig. 4.5.

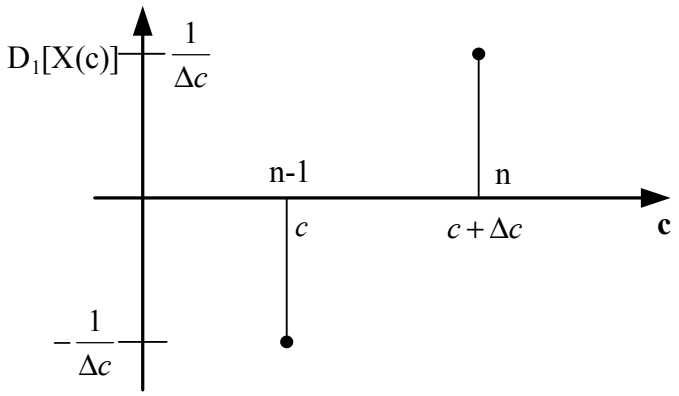


Figure 4.5 The first-order differential operator on impulse response

The second-order differential operation can be derived as

$$G(c) = \sum_{m=1}^{2} f_m[R(c)] = f_1[R(c)] + f_2[R(c)] \tag{4.5}$$

The differentiation functions are defined in terms of differential operators as

$$\begin{aligned} D_2 &= f_1[R(c)] + f_2[R(c)] \\ &= D_1^{(n)} + D_1^{(n+1)} \\ &= \frac{1}{\Delta c}\left(X_n - X_{n-1}\right) + \frac{1}{\Delta c}\left(X_{n+1} - X_n\right) \\ &= \frac{1}{\Delta c}\left(X_{n-1} - 2X_n + X_{n+1}\right) \end{aligned} \tag{4.6}$$

Figure 4.6 shows the second-order differential operation in sampling interval Δc .

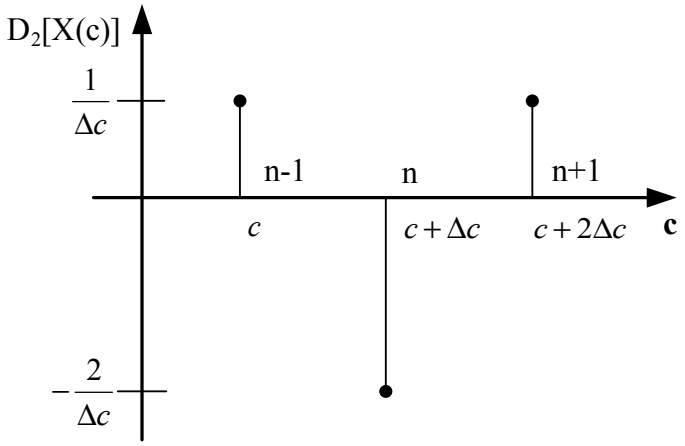


Figure 4.6 The second-order differential operator in impulse response

This impulse response representation for the differential operator provides a clear geometric interpretation of the process. The higher-order differentiations can be represented in the form of an impulse response demonstration.

4.2.2 Receptive Fields

In the biological visual system, the retina has a receptive field to receive light stimuli. Each spot on the fields affects the illumination of other spots by charging or discharging the stimuli [4]. The aggregation function reproduces the performance of the receptive fields. In the literature, several aggregation functions corresponding to the differential operator are shown to emulate the receptive fields of the retina. The structure of the receptive fields is illustrated in Fig. 4.7.

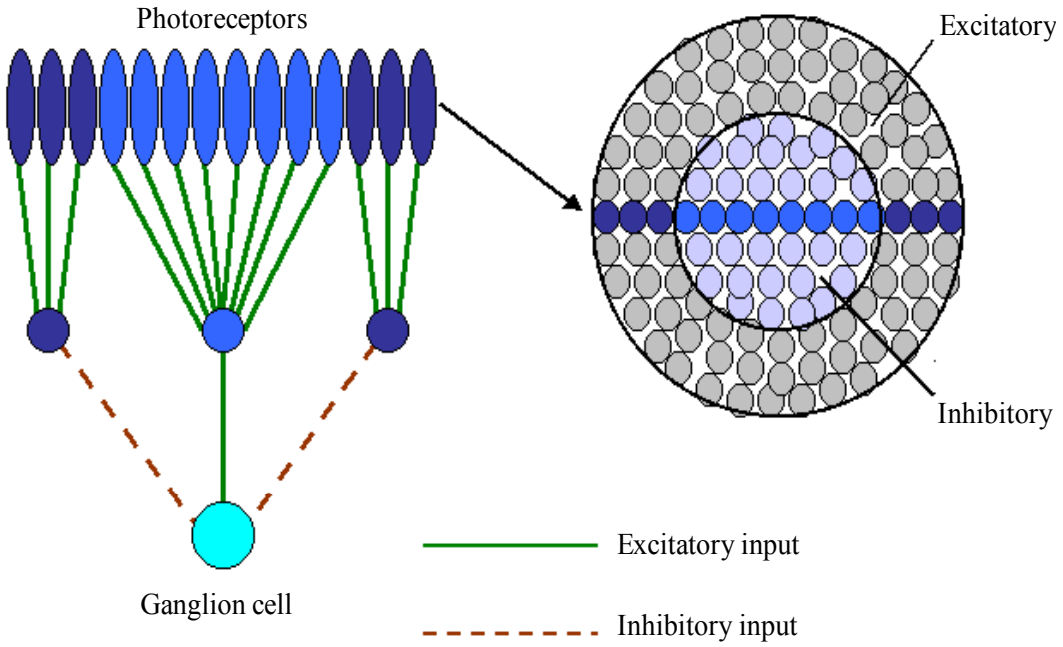
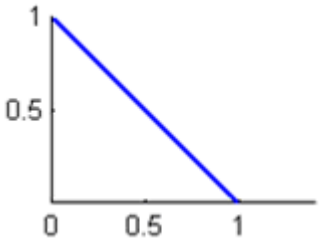
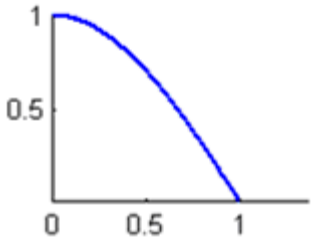


Figure 4.7 A schematic diagram of the retinal ganglion receptive field

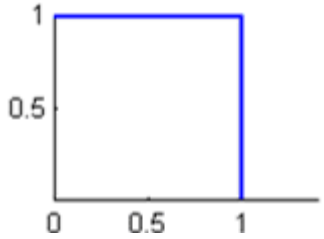
Several examples of aggregation functions are shown in Fig. 4.8. The aggregation function takes the identical role of the receptive field. Each pixel of the image is affected by the neighbouring pixels as in the scheme of the receptive field. This aggregation function is regarded as the discriminant function. The discriminant function extracts the information of the image as the receptive fields do in the visual organ [30].



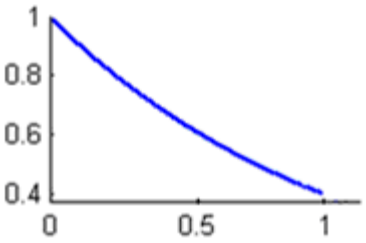
(a) Triangular



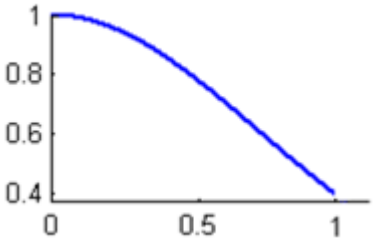
(b) Sinusoidal



(c) Rectangular



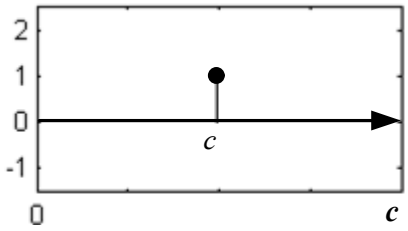
(d) Truncated exponential



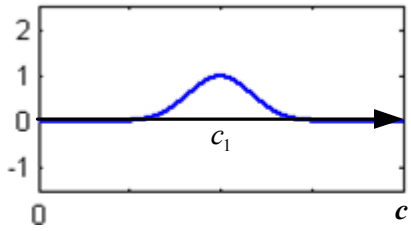
(e) Truncated Gaussian

Figure 4.8 One-dimensional examples of aggregation functions

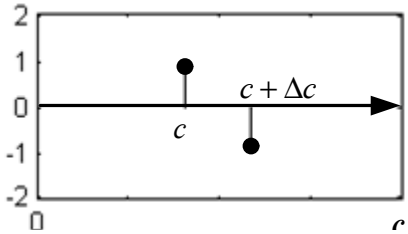
In this section, the Gaussian function is applied to the discriminant function as the Gaussian function calculates the mean of instance values with neighbourhood. Moreover, the Gaussian function is continuous and differentiable, so that this function should be taken as a discriminant function. Figure 4.9 shows the Gaussian function and the impulse response of the differential operators.



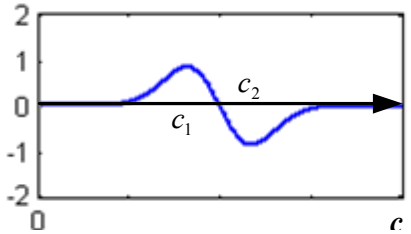
(a) Differential operator



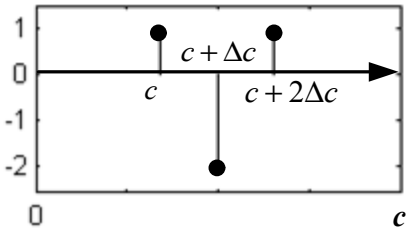
(b) Gaussian function



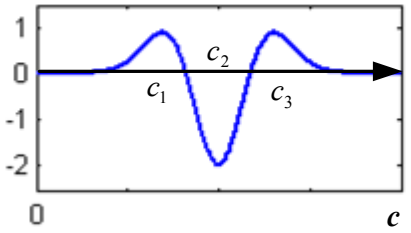
(c) First-order differential operator



(d) First-order differentiation with Gaussian operation



(e) Second-order differential operator



(f) Second-order differentiation with Gaussian operation

Figure 4.9 The differential operators without and with Gaussian function

4.2.3 Edge Detection

The edge is obtained from the difference of the brightness or contrast of an object and represented by the second-order differential operator. The edge detection can be based on temporal correlations between all neurons in the image domain that are hit by the same object. This is possible with the help of pre-existing connections that encode the likelihood of a pair of neurons belonging to the same segment. Signals from neurons in

the same segment become correlated and can be interfered with by other signals in graph matching. Signals from neurons in different segments become anti-correlated and cannot interact [27]. This procedure for the neural edge detection is similar to the biological vision procedure shown in Fig. 4.10 and the following:

- Scan an image
- Apply second-order differentiation of the Gaussian function, *Laplacian of the Gaussian* (LG), to convolute the scanned image to represent the receptive field
- Make the neural unit, neural processor, learn color spatial functions
- Pass the convoluted image through the neural processor to detect the edges of the objects in the scanned image

The Gaussian function is defined as Eqn. (4.1). The first-order differentiation of the Gaussian function is derived as

$$DG(x) = Ga'(x) - 2\alpha x e^{-\alpha x^2} = -2\alpha x Ga(x) \tag{4.7}$$

Thus, the LG is derived as

$$\begin{aligned} LG(x) &= \{-2\alpha x Ga(x)\}' \\ &= (-2\alpha)Ga(x) + (-2\alpha x)\{-2\alpha x Ga(x)\} \\ &= (-2\alpha + 4\alpha^2 x^2)Ga(x) \\ &= 2\alpha(2\alpha x^2 - 1)Ga(x) \end{aligned} \tag{4.8}$$

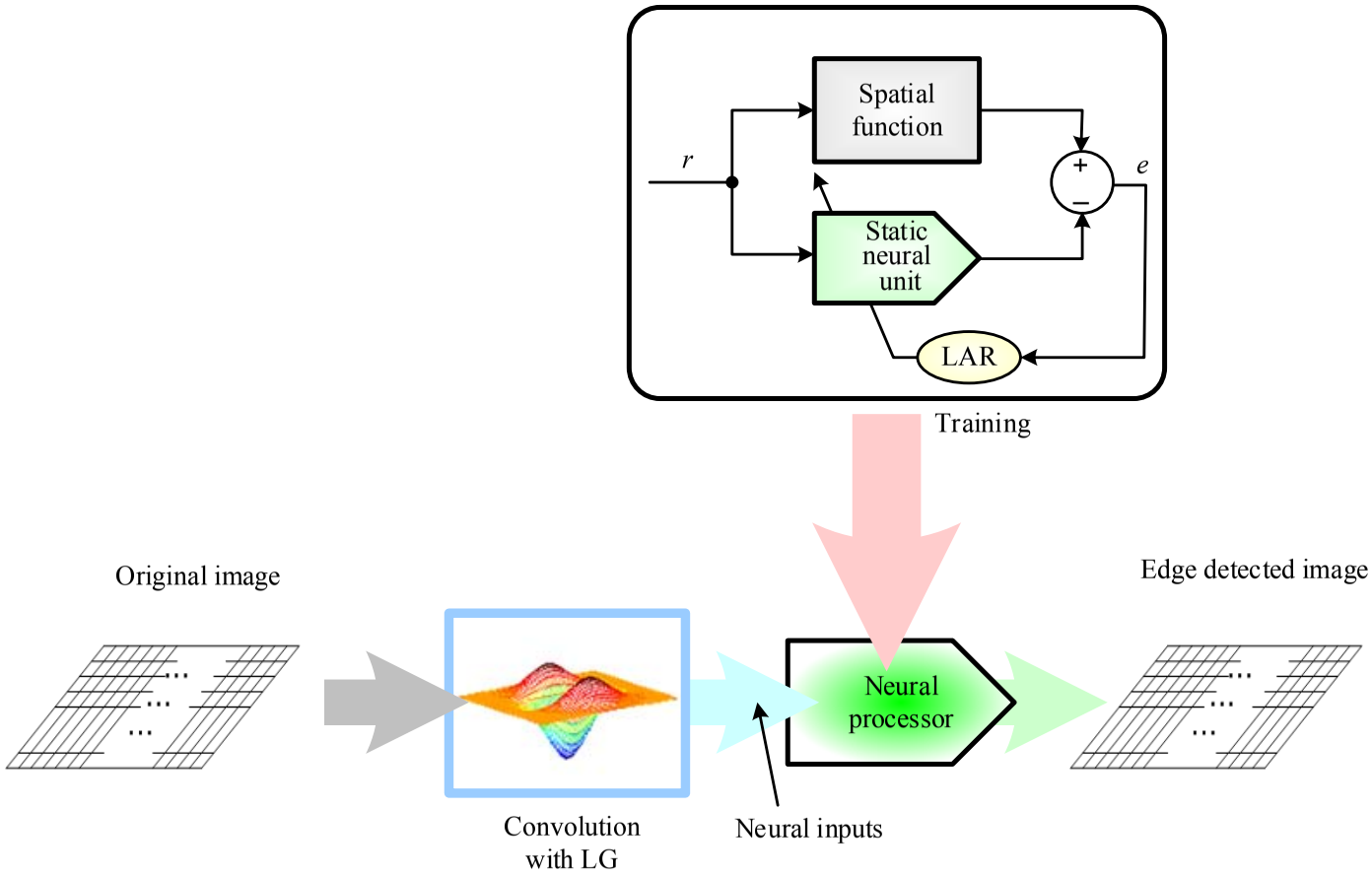


Figure 4.10 Neural procedure for edge detection: The convoluted image is processed through the trained neural processor by the spatial function

For optimal results, (5×5) LG matrix is recommended. The three-dimensional LG is applied as an optimal aggregation function. The given image is convoluted by the LG performing differentiation with averaging as the receptive fields. The convoluted data of the image is fed to the neural processor as the neural input. The neural processor is constructed by training the static neural units with the color spatial function. The spatial function verifies the contrast or brightness of the color in the image. The range of the spatial function is from 0 to 255, which signifies the gray level color. The neural processor generates a (1×1) neural output with a (3×3) neural input matrix from the convoluted image. The (3×3) neural input matrix generates the optimal output from these experiments. Each neural output affects the neighboring segments charging or discharging the illumination. The effect produces a clear edge detected image. Figure 4.11 shows the section of the converted image which supplies the neural inputs for the neural processor.

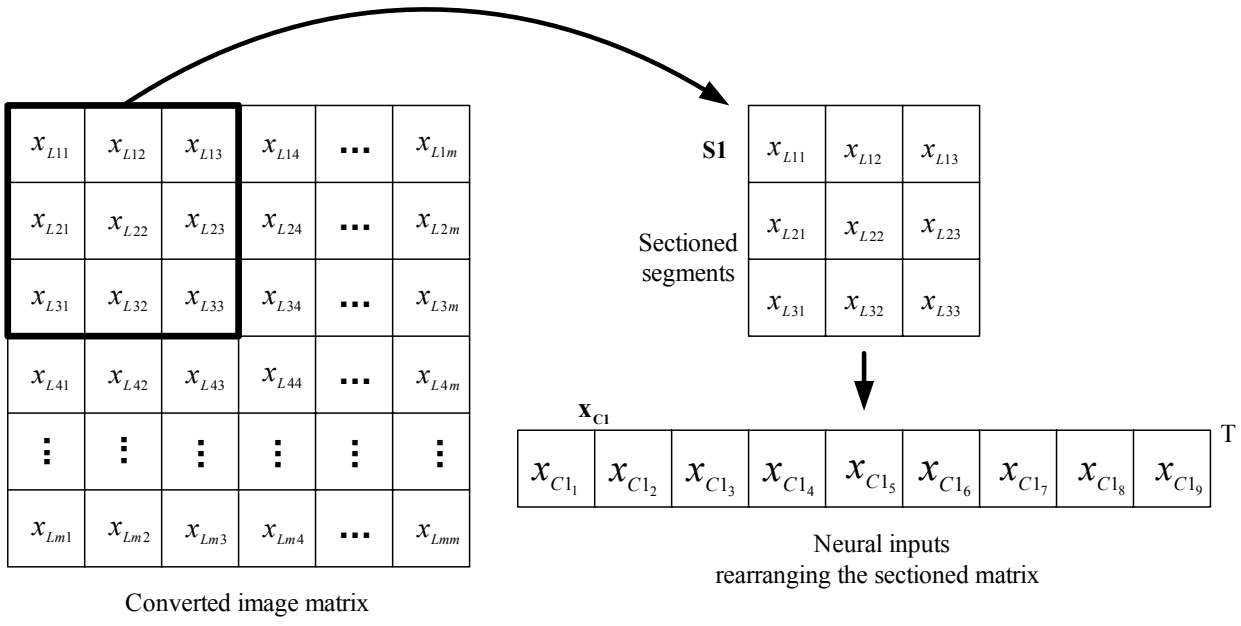
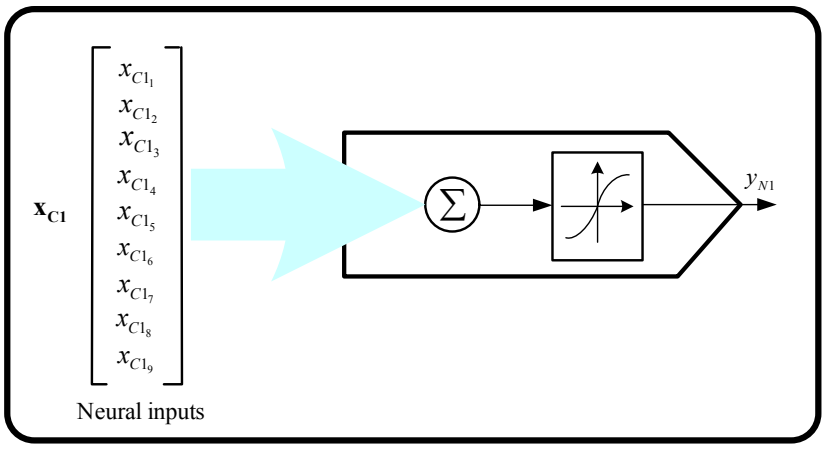


Figure 4.11 Neural input matrix from the convoluted image matrix

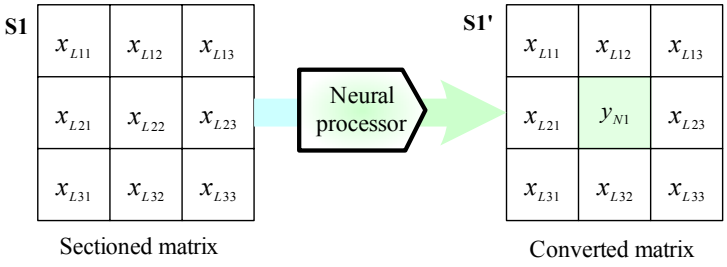
The sectioned segments are rearranged as neural inputs in order of elements in row and column. The selected (3×3) matrix **S1** becomes the (9×1) neural input $\mathbf{x}_{\mathbf{C1}}$. After the synaptic and somatic operations, the neural processor generates the neural output as

$$y_{N1}(x) = \Phi[v_1(x)] \tag{4.9}$$

After the first selected matrix procedure, x_{L22} of **S1** is replaced with y_{N1} because the middle element of **S1** is adjacent to every other element and the middle element may be mostly affected by its neighbours. Thus, **S1** is renamed as **S1'**. Figure 4.12 illustrates the neural procedure of the first converted sectioned segments.



(a) Neural processor



(b) Converting the selected matrix

Figure 4.12 The neural processor with the first selected neural matrix and the conversion of the matrix

The next sectioned matrix **S2** contains a new element from the previous converted matrix and is illustrated in Fig. 4.13. As the process is continued, **S2** generates y_{N2}, **S3** creates y_{N3} and so forth.

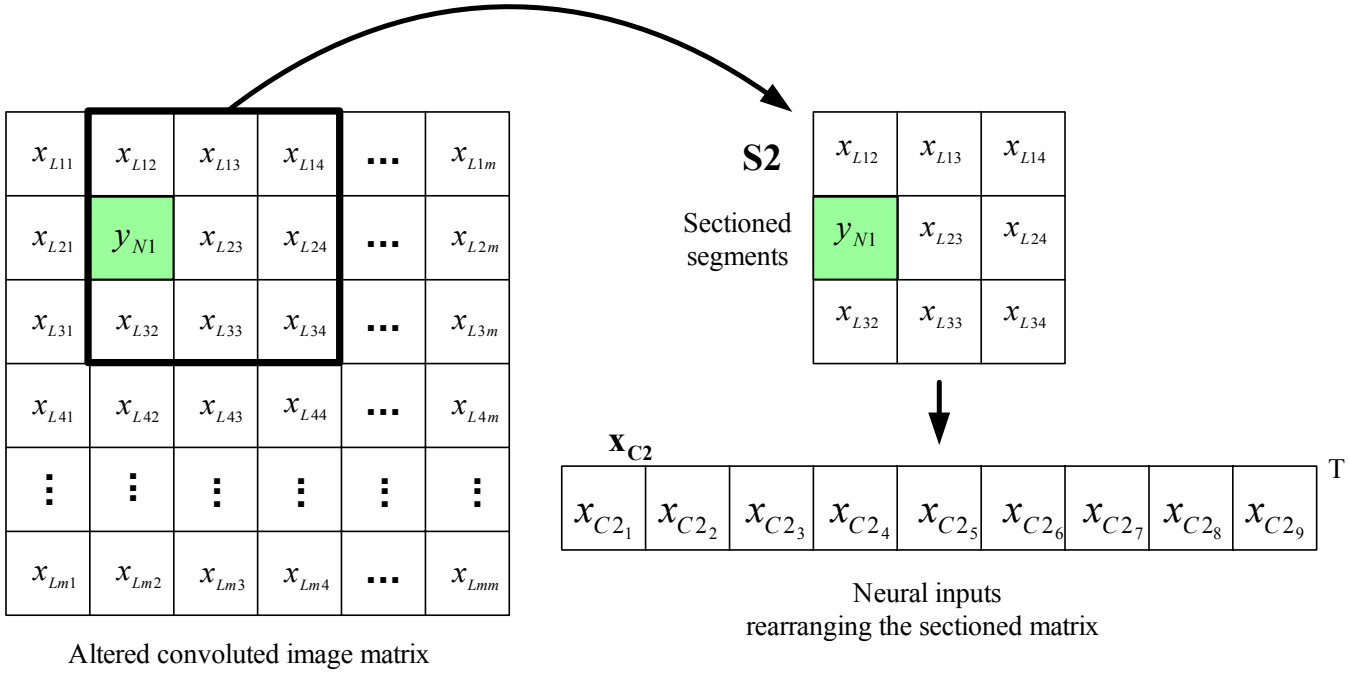


Figure 4.13 The next sectioned segments with y_{N1} for the neural inputs

After the entire neural procedure is completed, the convoluted image matrix is changed as shown in Fig. 4.14. The gray coloured cells signify that the neural processor, which represents the function of the central nervous system, computes the elements in these cells.

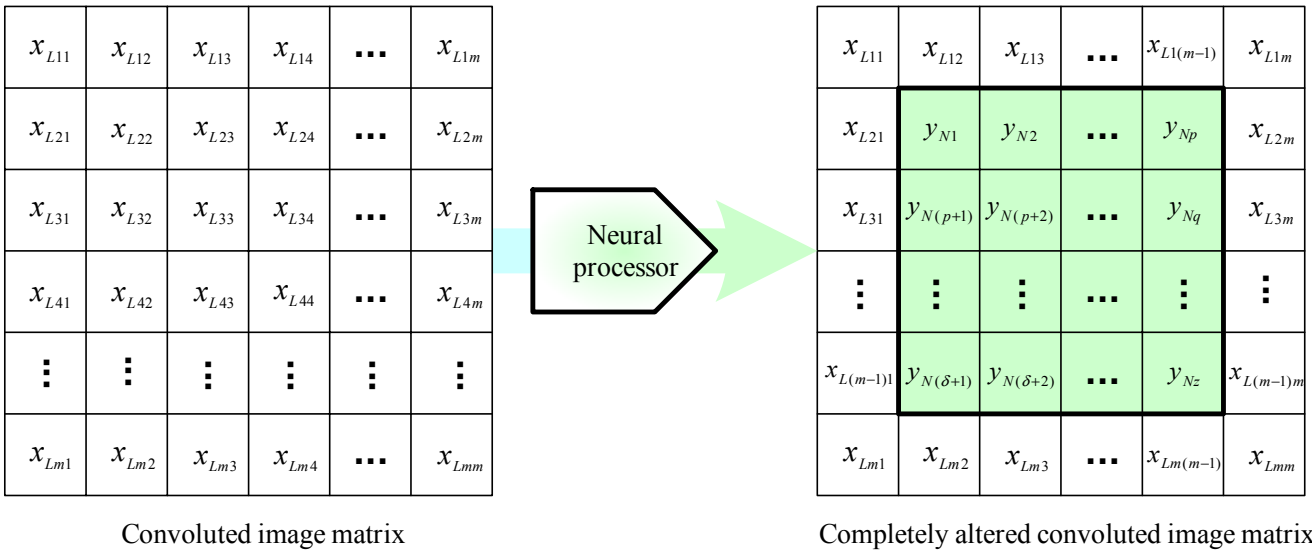


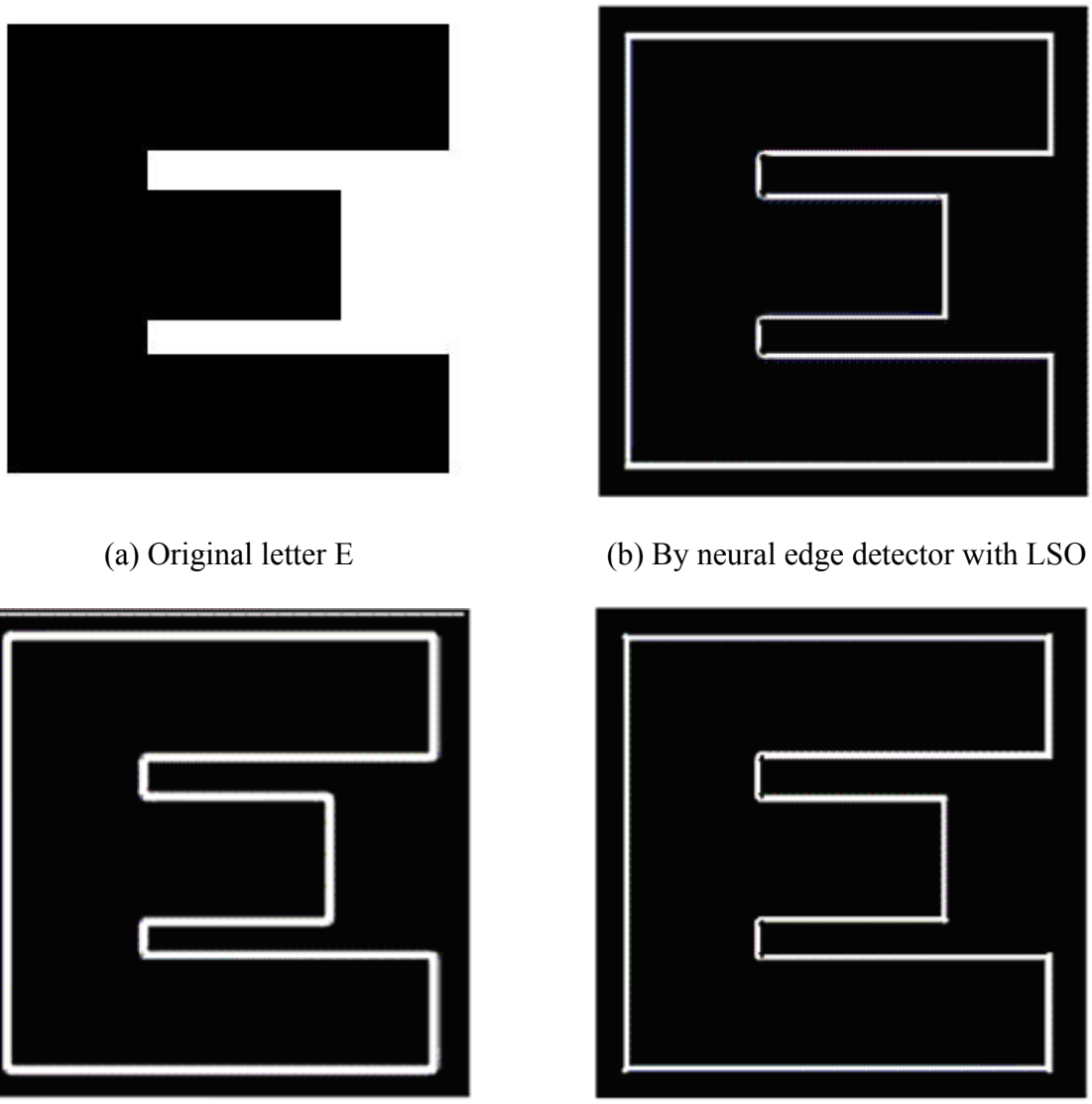
Figure 4.14 Completely altered image matrix after the neural processor

The transformed convoluted image from the neural processor enhances the edges of the objects in a given image by higher-order computation. In order to obtain the optimal edge detected image, the rate of slope of the Gaussian function and the size of the sectioned matrix should be adjusted.

4.3 Simulation Studies of Neural Edge Detectors

In this section, the simulation of the neural edge detection is illustrated following the procedure of the neural method explained in the previous section. The neural unit with CSO is used as the neural processor, and simulation results are displayed and compared with the performances of the neural units with LSO and QSO. In this thesis, two different images are used for the edge detection. One is an image of the letter E, and the other is a gray picture of a lady, Lena. The original letter E and the edge detected images processed by the neural units with LSO, QSO and CSO are displayed in Figs. 4.15. The two-dimensional plots do not give a clear representation of the edge detected images. Thus,

the three-dimensional plots are shown. In three-dimensional plots, the z-axis of the graph expresses the gray level of each pixel. The edge of the image is extracted with the higher value in the z-axis. Figure 4.16 gives the three-dimensional plots of the original images and edge detected images of the letter E. The original Lena image and the edge detected images by the neural edge detectors are shown in Fig. 4.17. Three-dimensional plots of images of Lena are shown in Fig. 4.18. The neural edge detectors, however, did not present thin edge images. The thinning procedure makes the line thin to present the clear border of the objects. As a further step, a thinning procedure was applied following the neural processor. Lena's image was processed by the thinning procedure, and the results are shown in Fig. 4.19. As shown, the edge-detected images are not clear due to the size of the pictures. For a clear view of the simulation results, a zoomed version of several regions of the edge detected images is illustrated in Fig. 4.20. In these figures, each row of the pictures corresponds to the result of neural edge detectors, and each column represents selected regions of the image. Hence, the edge detected images from different neural edge detectors can be compared. The three-dimensional plots of the thin edge detected images of Lena are displayed in Fig. 4.21.



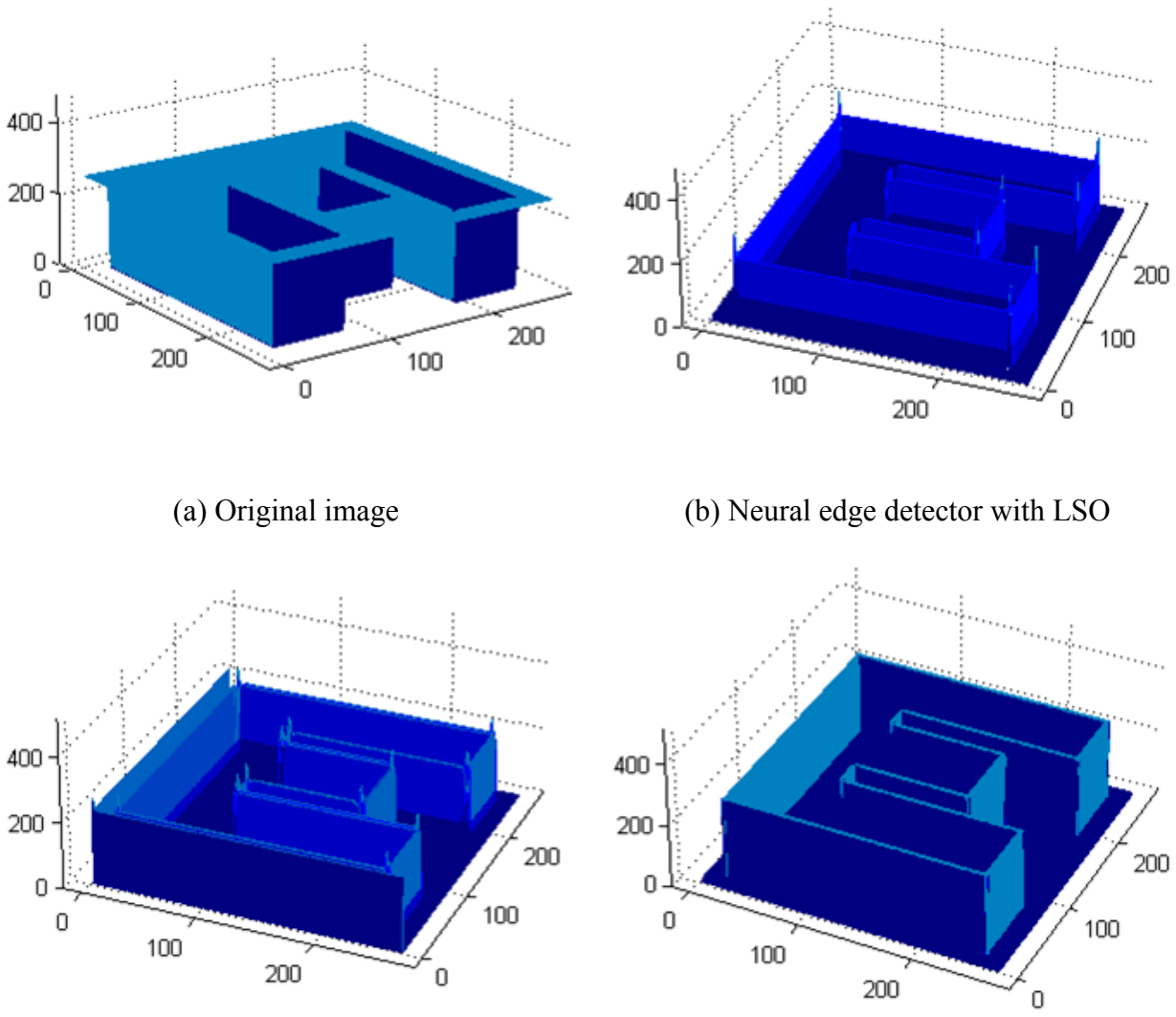
(a) Original letter E

(b) By neural edge detector with LSO

(c) By neural edge detector with QSO

(d) By neural edge detector with CSO

Figure 4.15 Original letter E image and the edge detected images from different neural detectors with the optimal slope rate of the Gaussian function, $\alpha = 0.03$



(a) Original image

(b) Neural edge detector with LSO

(c) Neural edge detector with QSO

(d) Neural edge detector with CSO

Figure 4.16 Three-dimensional plots of the original letter E image and edge detected images by different neural processors

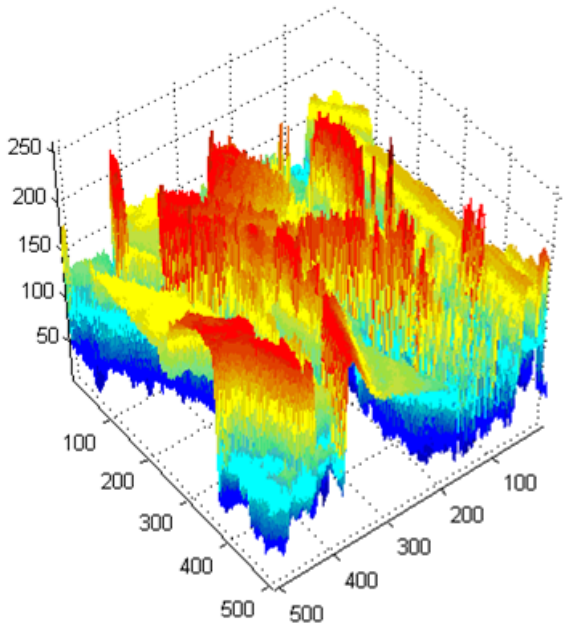
(a) Original Lena image

(b) By neural edge detector with LSO with $\alpha = 0.035$

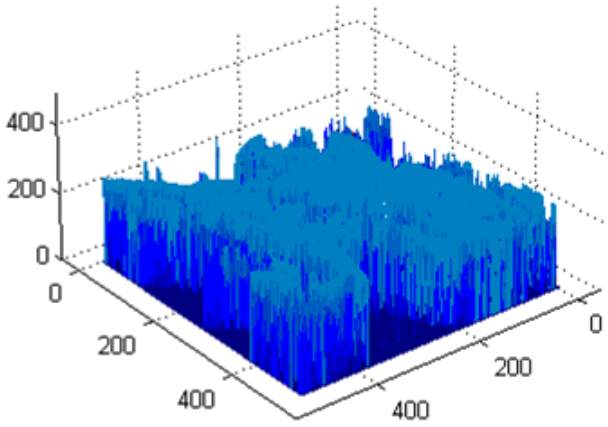
(c) By neural edge detector with QSO with $\alpha = 0.08$

(d) By neural edge detector with CSO with $\alpha = 0.01$

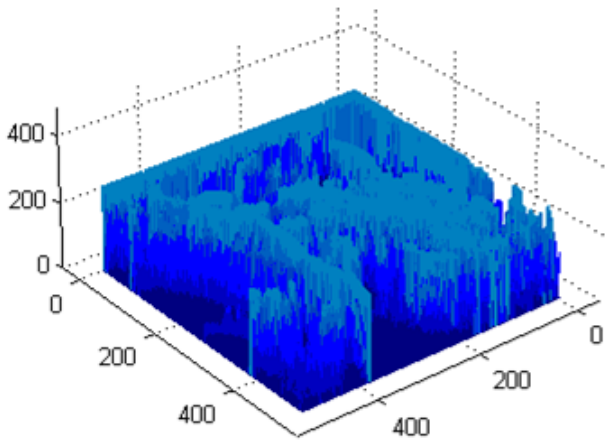
Figure 4.17 The original Lena image and the edge detected images processed by different neural edge detectors with the optimal slope rate of the Gaussian function, α



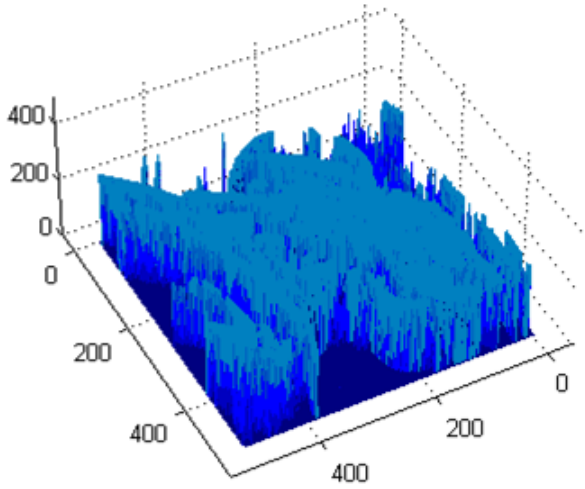
(a) Original image



(b) Neural edge detector with LSO



(c) Neural edge detector with QSO



(d) Neural edge detector with CSO

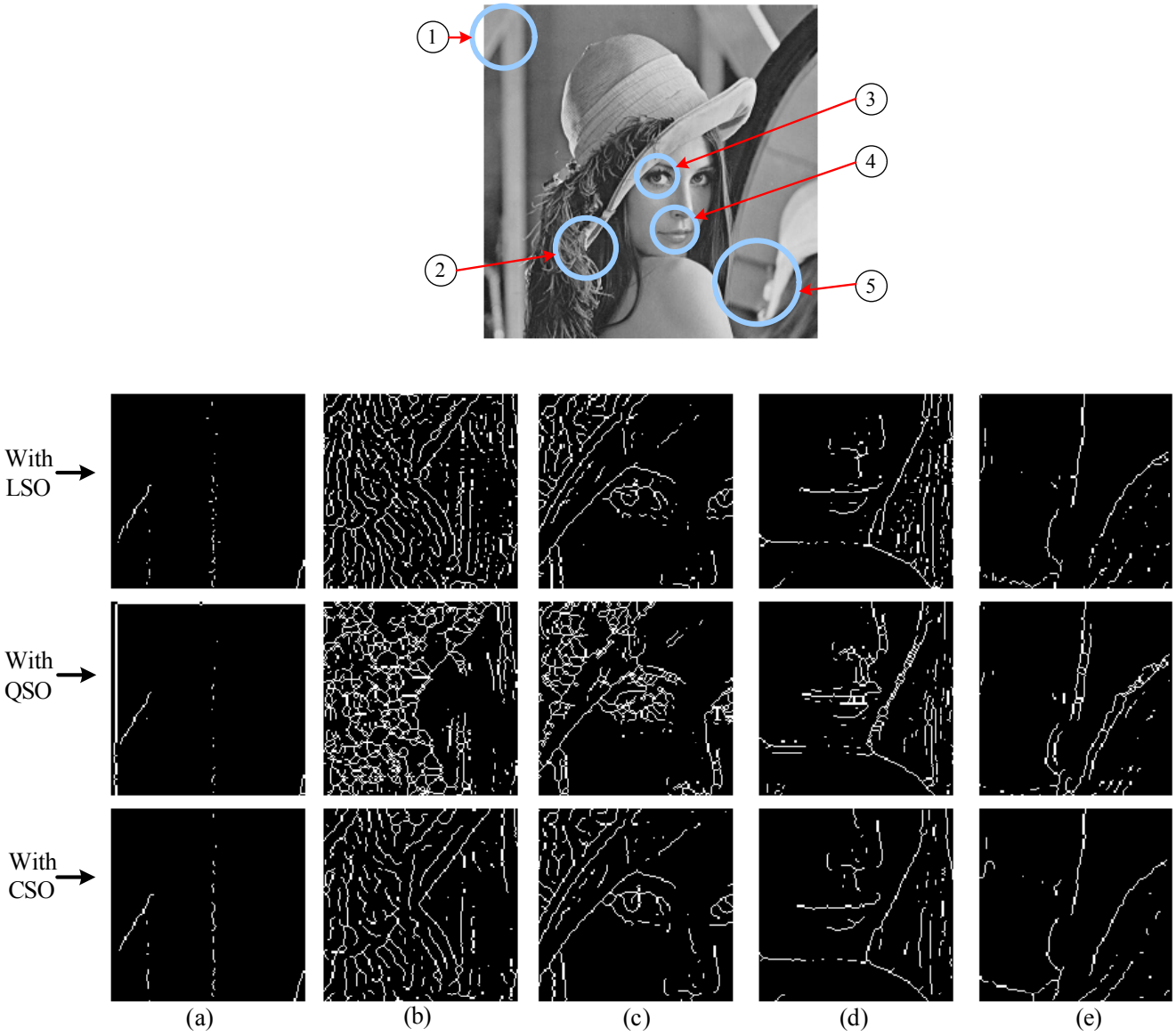
Figure 4.18 Three-dimensional plots of the original Lena image and edge detected images by different neural processors

(a) Neural edge detector with LSO

(b) Neural edge detector with QSO

(c) Neural edge detector with CSO

Figure 4.19 The edge detected Lena images by the different neural processors after the thinning procedure



(a) the close thin figure of region 1 from the original image

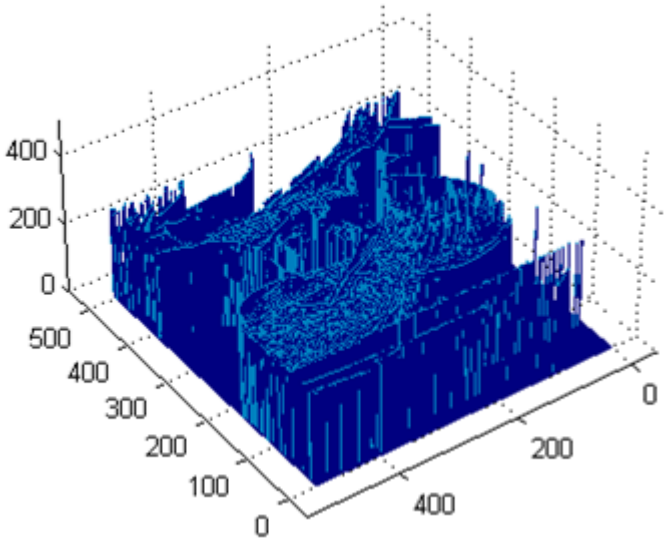
(b) the close thin figure of region 2 from the original image

(c) the close thin figure of region 3 from the original image

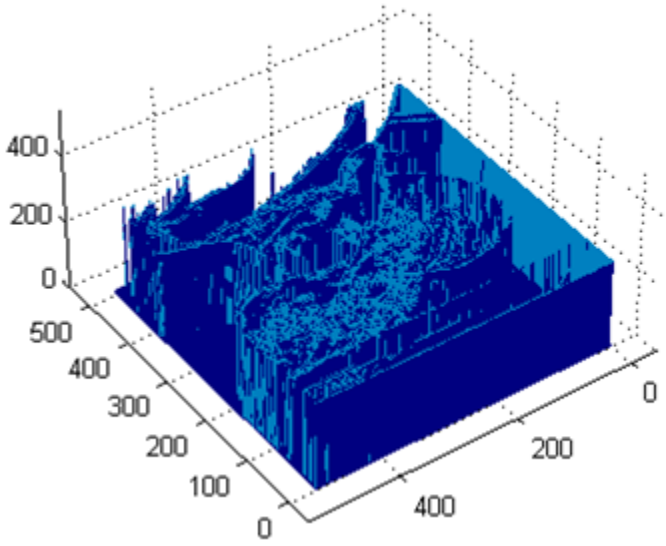
(d) the close thin figure of region 4 from the original image

(e) the close thin figure of region 5 from the original image

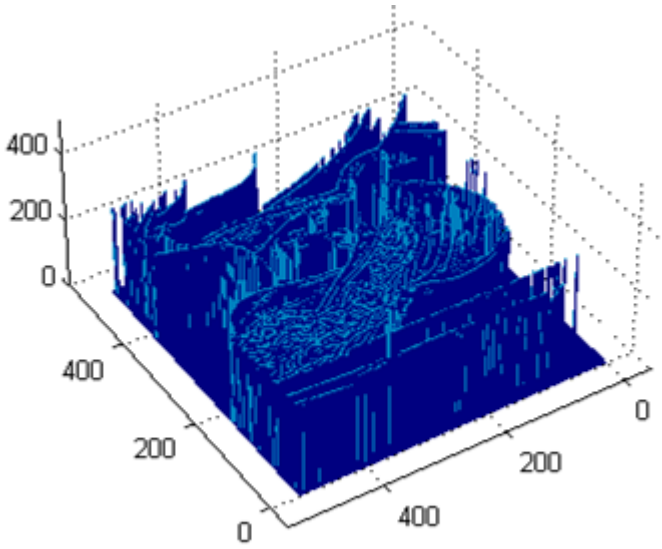
Figure 4.20 The edge detected images from 5 different regions by neural edge detectors with LSO, QSO, and CSO after thinning



(a) Neural edge detector with LSO



(b) Neural edge detector with QSO



(c) Neural edge detector with CSO

Figure 4.21 The three-dimensional plots of the thin edge detected Lena images from different neural edge detectors after thinning

4.4 Summary

A novel neural method of edge detection was illustrated in this chapter. This new neural method was based on the biological vision system. Moreover, mathematical expressions of the biological system were introduced. $A(3\times3)$ neural input matrix and (5×5) LG convolution window generated the optimal edge detection as shown in the simulation results. The new algorithm was applied with the neural units with LSO, QSO and CSO. As shown by the simulation results, the higher-order neural structure gave clearer edged lines of the objects in the given images. Especially, the neural unit with CSO presented more perceptible edge detected images. However, the neural unit with QSO could not adequately extract the edges from Lena's image. As an edge detector, this neural method was not able to generate the thin line which is essential to analyze the edge-detected images. The neural thinning method should be developed for future work.

Chapter 5

Mobile Robots with Neuro-vision and Neuro-control

5.1 Introduction

Human beings have built robots since the twentieth century. According to the Merriam-Webster Online dictionary, a robot is defined as *"a machine that looks like a human being and performs various complex acts (as walking or talking) of a human being"*. Numerous robots are manipulated by various kinds of controllers. Neural networks are starting to be used to control these machines. In the literature, it is thought that the neural networks may result in better performance of the machinery. Most current robotic problems can be categorized into one of three processing levels: task planning (e.g., depth determination and arm-camera coordination), path planning (e.g., robot navigation), and path control (e.g., motor control). Most robotic processing problems can be formulated in terms of optimization or pattern association problems. Neural networks can be adopted to solve these robotic processing tasks [13].

In this chapter, a mobile robot is introduced as a neural control application. The mobile robot is illustrated in Fig. 5.1. A CCD camera for vision is implemented, and several photo-sensors are attached to the machine. It was proved in the previous chapters that neural units with HOSO have superior capability; these higher-order neural vision and controller may give the machine an advanced performance. Vision for the machine is enhanced as neuro-vision. A part of this neuro-vision was processed in the previous chapter as the neural edge detection. In order to apply neuro-vision appropriately, the

Hough transform (HT) method is introduced in this chapter. For the control application of this machine, the neural unit with CSO is used due to its superior performance.

In this chapter, neuro-vision with the HT and the estimation of location for the robot's movement decision is explained in Section 5.2. Neuro-control with the neural unit with CSO to manage the movement of the mobile robot is described in Section 5.3. In addition, a summary of this chapter is given in succeeding section.

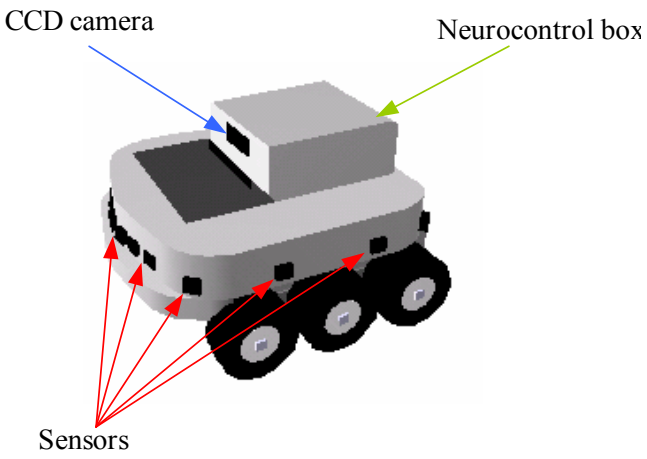


Figure 5.1 Mobile robot

5.2 Neuro-vision System

Vision is one of the most important faculties of perception for human beings. The study of biological vision provides motivation and a general framework for designing and developing fast, robust, and effective machine vision systems. Current machine vision performance is generally no more advanced than the most primitive animal vision system [35]. The term 'neuro-vision' is used to refer to any artificial or machine vision system that embodies the computational principles of biological neural circuits. The process of

designing neuro-vision systems based on biological analogies is more properly termed *reverse bioengineering* or *inverse biomedical engineering* [29, 42].

5.2.1 Hough Transform (HT) Method

In computer vision and image processing, it is imperative to detect basic shapes such as lines and circles. Neural edge detection was developed in the previous chapter in order to find out the shapes of the figures. However, neural edge detection cannot enable the machine to recognize objects completely. In order for the robot to use the edge detected image, the HT is required to compute and transfer the image data. The HT is one of the most powerful methods for detecting the basic shapes from landmarks, even though some landmarks may be distorted or covered up [37]. The original form of the HT involved parametrizing lines and was described by the slope-intercept equation as

$$y = mx + c \tag{5.1}$$

In this equation, every point on a straight edge of the edge detected image is plotted as a line in (x, y) space corresponding to all the (m, c) values consistent with its coordinates, and the lines are detected in this space. One of the detriments of the original HT is that the value of m or c becomes infinity if the line is parallel with x-axis or y-axis. A modified HT was, therefore, introduced in order to remove this disadvantage, which substitutes the normal (θ, ρ) form for the slope-intercept format for the straight line as

$$\rho = x\cos\theta + y\sin\theta \tag{5.2}$$

where ρ is the distance from the origin to the point **P**, and θ is the angle from the x-axis to the point **P**.

The set of lines passing through the point **P** is represented as a set of sine curves in (θ, ρ) space. Multiple hits in (θ, ρ) space signify the presence of lines in the original image [36]. Figure 5.2 shows the scheme of the normal (θ, ρ) parametrization of a straight line, and the (θ, ρ) parametrization of a straight at point **P** in (θ, ρ) space is illustrated in Fig. 5.3.

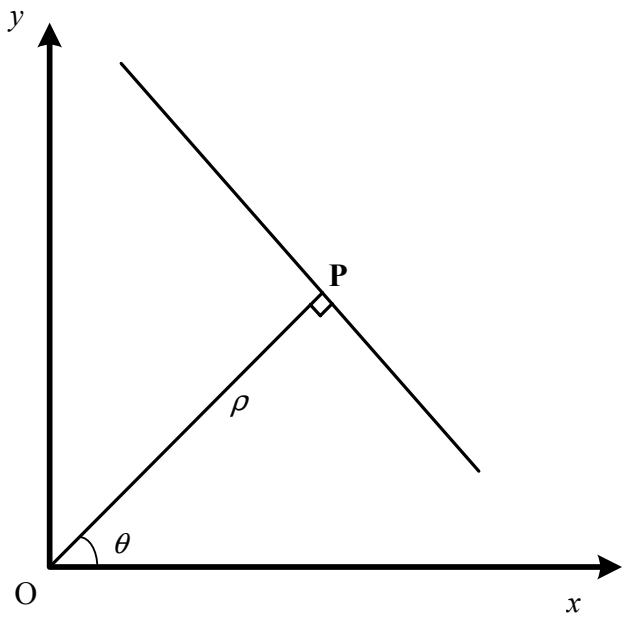


Figure 5.2 Normal (θ, ρ) parametrization of a straight line at point **P** in *x*-*y* space

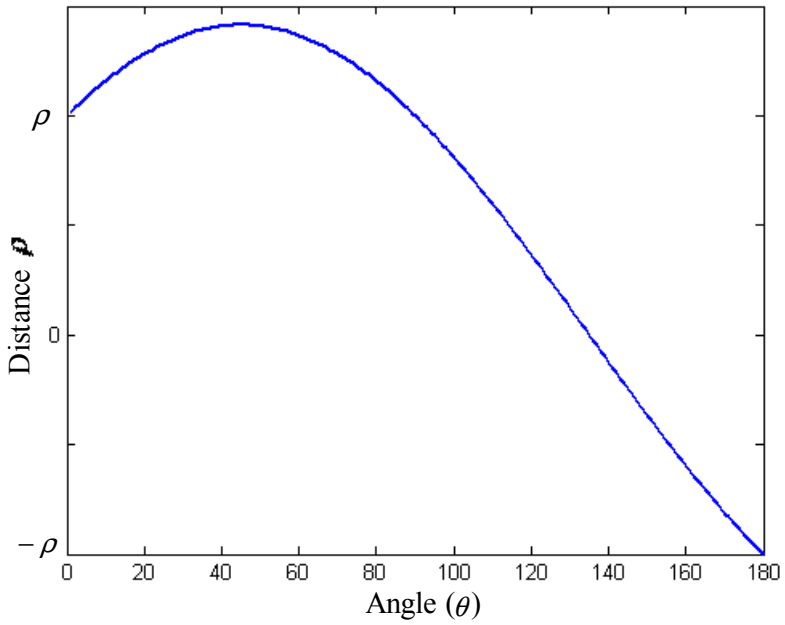
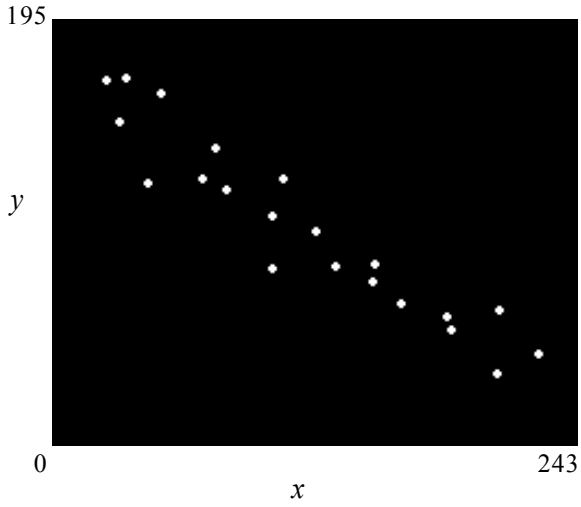
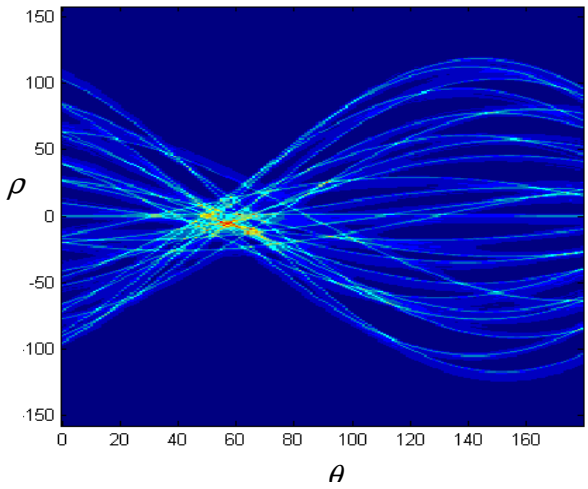


Figure 5.3 Normal (θ, ρ) parametrization of a straight at point **P** in $\theta - \rho$ space

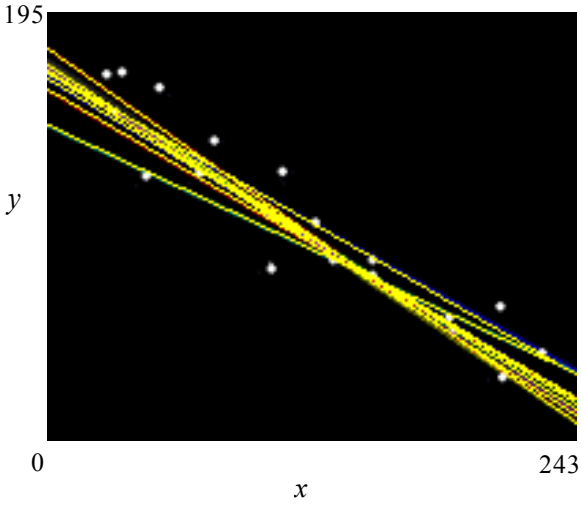
Consider that some lines are detected as the landmarks shown in Fig. 5.4(a). Figure 5.4(b) shows the relationships between θ and ρ for all the landmarks. The parameter space ($\theta - \rho$ space) is divided into many small subspaces, and the relationships vote for the corresponding subspace. The central values of the subspaces which have many votes are selected as the parameters of the lines to be detected. Figure 5.4(c) and 5.4(d) show the detected line and the landmarks. The lines are detected corresponding to the threshold of the HT.



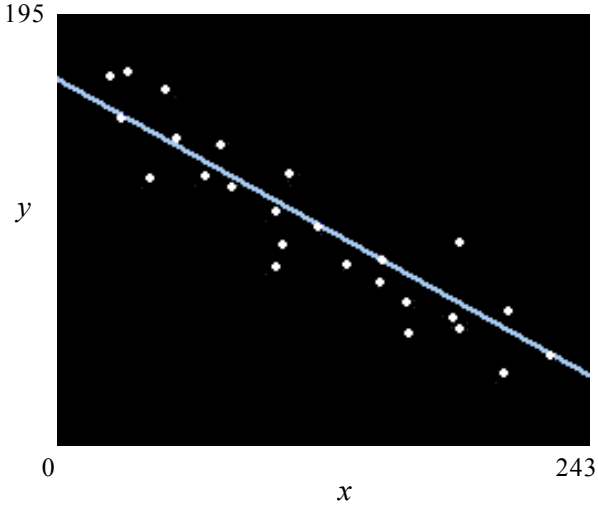
(a) Landmarks



(b) The parameter space for the landmarks



(c) The result of the line detection with the threshold 0.8



(d) The result of the line detection with the threshold 0.9

Figure 5.4 An example of detection of lines by HT

5.2.2 Navigation for the Mobile Robot

It is very essential to implement a vision system for the mobile robot. The basic requirements for the autonomous navigation of a mobile robot are environmental recognition, path planning, driving control, and location estimation/correction capabilities

[45, 46, 47]. A CCD camera is put on the mobile robot to get the surrounding image. The image from the CCD camera is processed into a digital image through the neural edge detector and the HT, so that the mobile robot can receive data for navigation and travel corresponding to the CCD camera information. The detected lines of the path of travel can be the long term memory of the machine. A picture from the mobile robot is taken in Fig. 5.5.

Figure 5.5 The hallway from the CCD camera on the mobile robot

The robot's neuro-vision detects the edge of the picture and identifies the line of the corridor. Figure 5.6 shows the edge detected image of the hallway and Fig. 5.7 illustrates the parameter space of the edge detected image. Figure 5.8 and 5.9 depict the straight lines with different thresholds. The straight lines guide the mobile robot to move along.

Figure 5.6 The edge detected hallway image

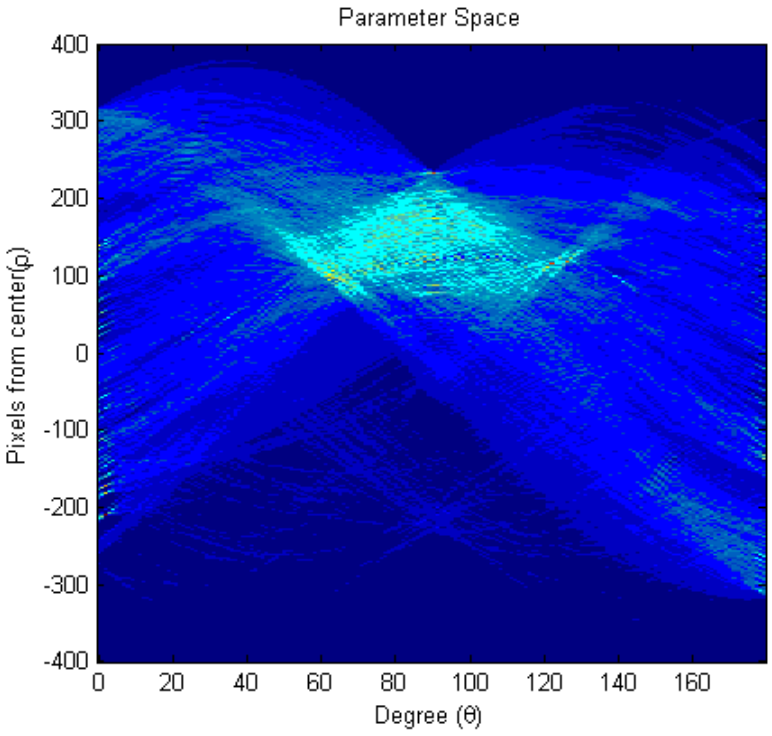


Figure 5.7 The parameter space of the edge detected image after the HT is used to find out the straight lines

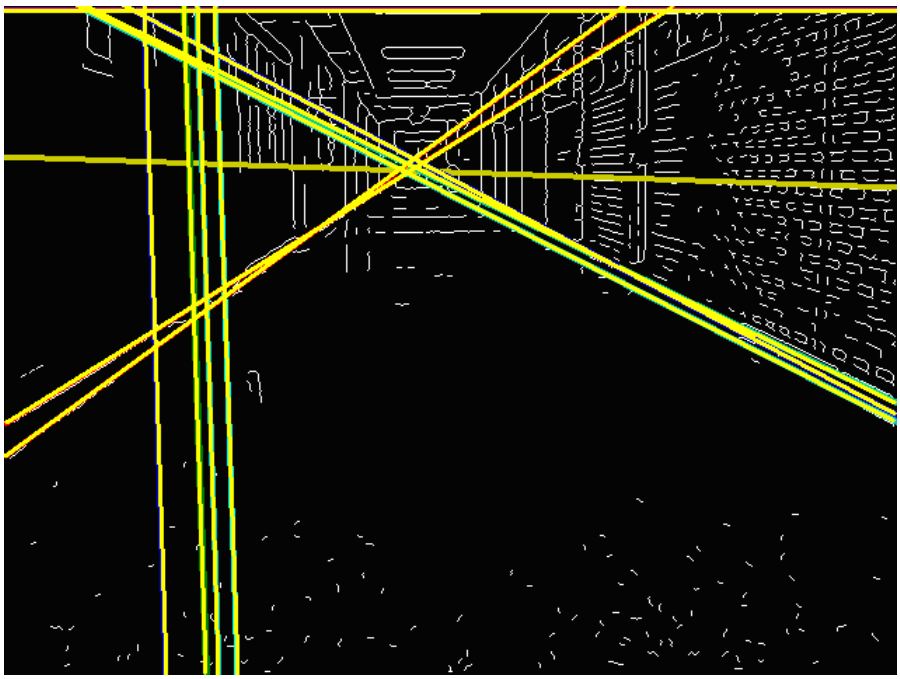


Figure 5.8 The straight lines after HT with threshold 0.7

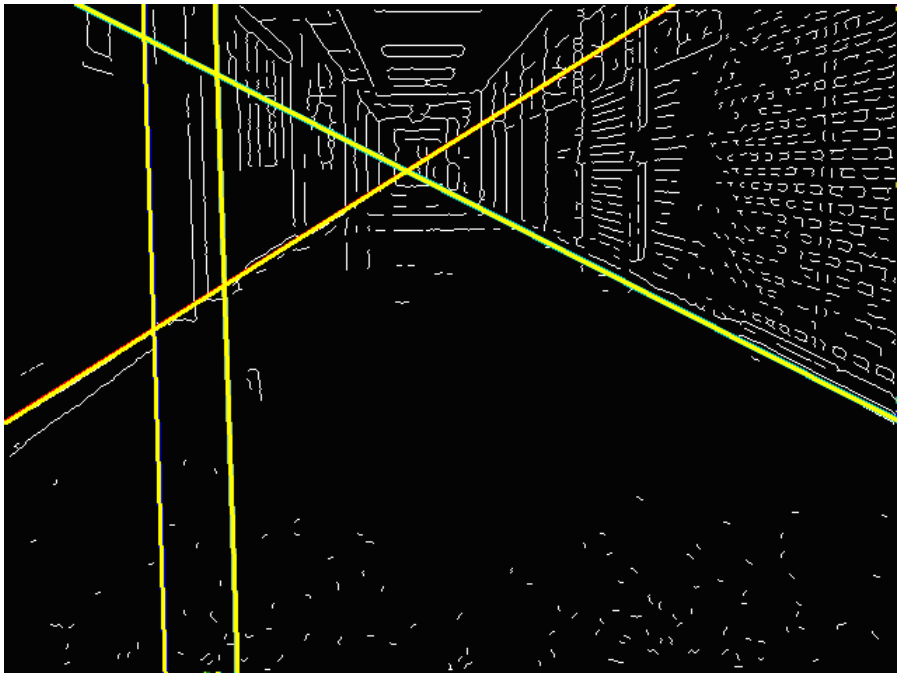


Figure 5.9 The straight lines after HT with threshold 0.8

5.2.3 Estimation of Location and Algorithms for the Robot's Movement Decision

To measure location and distance are very important capabilities for a mobile robot to autonomously execute given tasks. Vision-based methods have some advantages because of their flexibility and simplicity [44]. In the literature, a camera, ultrasonic, laser, radar and/or infrared sensors are used for the estimation of location to recognize and locate beacons. Those devices have become very popular recently, as they can provide precise location data instantaneously [48]. The scheme of the measurement of location for the mobile robot with sensors is shown in Fig. 5.10.

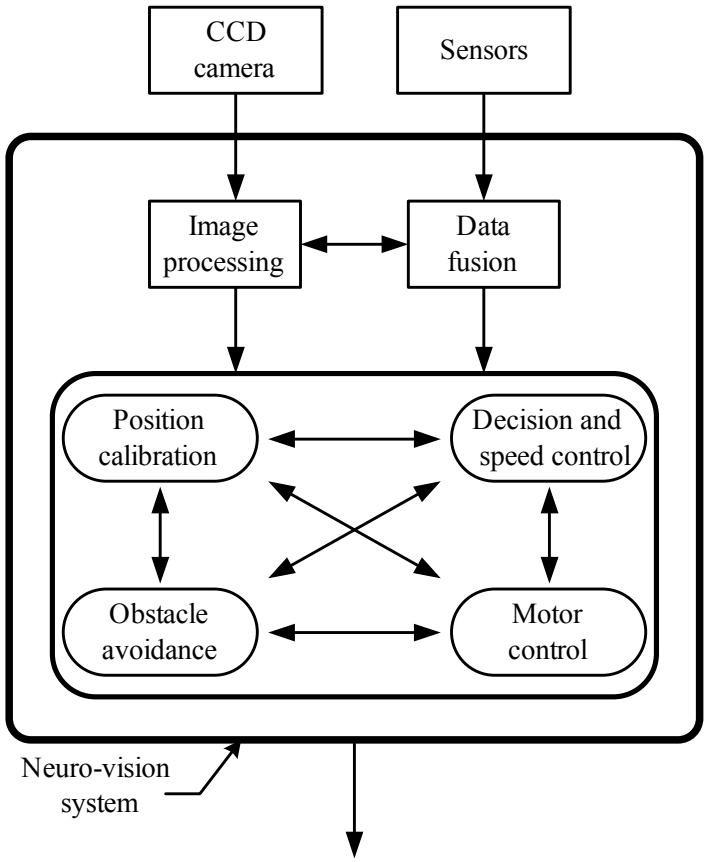


Figure 5.10 The scheme of the estimation of location with CCD camera and sensors

It is very important for the mobile robot to calculate its position and locate itself. In order to estimate the location and decide the direction, the CCD camera performs image processing, and the sensors generate data from the environment. Algorithms for the robot's movement procedure are designed using pseudo codes to meet possible movement problems. Table 5.1 represents the position calibration algorithm of the machine to activate corresponding to the digital image of the hallway. The robot measures the width of the hallway with respect to the width of the robot (Rw), keeping its position in the middle of the hallway with respect to the x coordinates of the mobile robot in the image (Px).

Table 5.1 Pseudo codes for position calibration

```
for Hallway width (H)
    if (H>Rw)
       Go forward
    end if
    if(H<=Rw)
       Turning or go backward
    end if
   for left feature
     COMPUTE intersection point (Pl)
   end for
   for right feature
     COMPUTE intersection point (Pr)
   end for
   COMPUTE mid point (Pm) between Pl and Pr
   for position
     if (Pm>Px)
       Turn Right
     else if (Pm<Px)
       Turn Left
```

(continued)

```
else
       Keep the position
    end if
  end for

end for
```

In order to achieve its tasks, the mobile robot should also have the capability to avoid obstacles by calibrating the distances between the obstacle and the wall of the hallway. The distance from the left side wall to the left side of the obstacle (Dl), and the distance from the right side wall to the right side of the obstacle (Dr) are measured by computing the processed image. Table 5.2 shows the obstacle avoidance algorithm of the mobile robot.

Table 5.2 Pseudo codes for obstacle avoidance

```
if no obstacle
   Keep the position
end if
if obstacle
   COMPUTE the distance (Dl)
   COMPUTE the distance (Dr)
  if (Dl>Dr)
    Turn Left
  else if
    Turn Right
  end if
end if
```

The motor speeds for changing directions and moving forward are altered corresponding to the obstacles. The fused data from the sensors help the motor speed control. Table 5.3 describes the algorithm of the motor speeds for the safe journey of the mobile robot.

Table 5.3 Pseudo codes for motor speeds

```
for data from sensor
  if no obstacle
    High motor speed for forward
    Decrease motor speed for turning
  end if
  if obstacle
    Low motor speed for forward
    Increase motor speed for turning
  end if
end for
```

The hypotheses for the movement decision with motor control are embedded in the neural networks composed of the neural units with CSO. The control system with neuro-vision, sensors and motor control is considered as the neuro-control system.

5.3 The Neuro-control System

Natural neurons play many important functions in the sensory, locomotion, and cognitive aspects of the central nervous system. The neural units in the higher cortical level provide some sort of cognition, or intelligence which represents the power to reason, think, learn and adapt. The term ‘neuro-control’ is given to refer to artificial control systems which have a similar intelligence process [13]. In the literature, difficulty in the control area is generally caused from the computational complexity, presence of nonlinearities and

parameter uncertainties. Currently, with the help of research on the superior abilities of the biological neural systems, robust artificial neural networks have been developed for information processing and control in order to respond to the complex environments of the types given above [7, 38, 39, 40, 41].

5.3.1 Controlling Mobile Robot with Neuro-control System

The mobile robot in this thesis is controlled by three categories: determination, navigation and motor control. The mobile robot can receive the information on the obstacles around the machine. The sensors on the body give this blockage data to a neural processor. The processor, then, analyzes the given information. The neural processor represents the *short term memory* (STM) of the machine. This mobile robot can also attain the navigation data as *long term memory* (LTM) from the neuro-vision system. The motor is controlled corresponding to the STM and LTM. Conversely, the STM and LTM are affected by the movement of the machine. Since the change of direction or movement may influence the machine's relationship to new environment, the robot's motor should be managed by the altered STM and LTM recurrently. Figure 5.11 shows the scheme of the neuro-control system and the mutuality of the three neural requisites of the mobile robot.

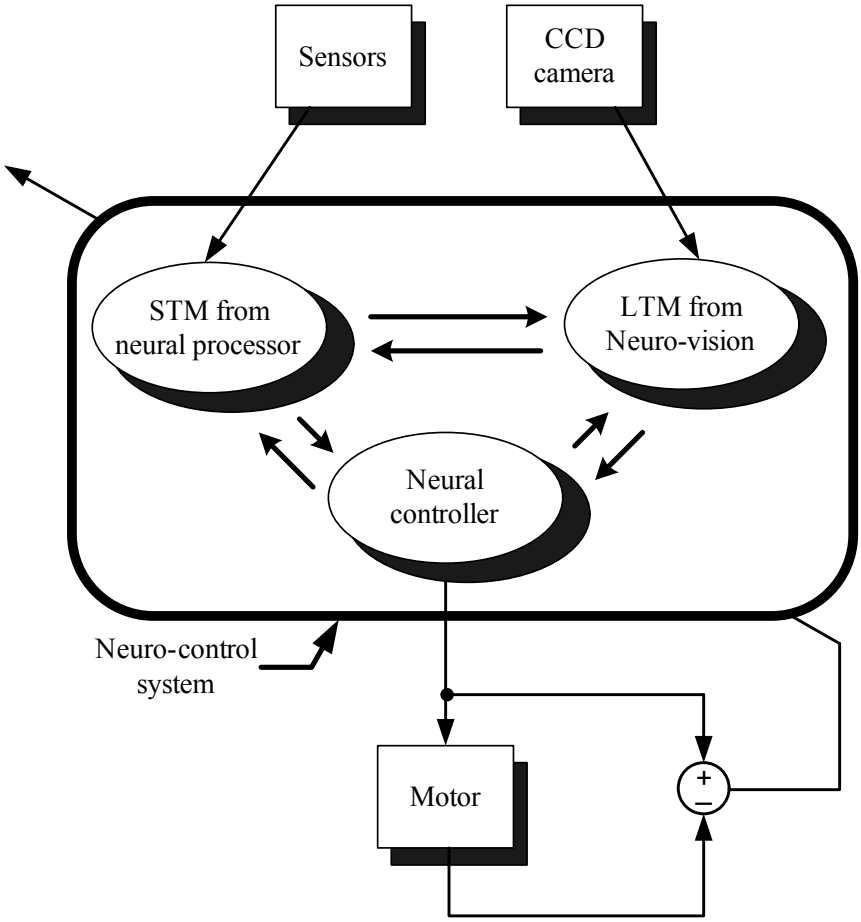


Figure 5.11 The scheme for the neuro-control system of the mobile robot

In this thesis, the movement of the mobile robot is described with the neural units with CSO. The neural unit with CSO is used to control the motor for the higher performance of the neural structure. A servomotor is employed for the mobile robot. A servomotor is a dc motor designed specifically to be used in a closed-loop control system [22]. The circuit diagram of a servomotor is shown in Fig. 5.12. In the Fig. 5.12, $e_a(t)$ is the armature voltage and considered to be the input of the system. R_m and L_m are the resistance and inductance of the armature circuit respectively. The voltage $e_m(t)$ is called back-EMF which represents the voltage generated in the armature coil due to the motion of the coil in the magneto of the motor. Hence, the back-EMF is defined as

$$e_m(t) = K\phi \frac{d\theta}{dt} \tag{5.3}$$

where K is a motor parameter, ϕ is the field flux, and θ is the angle of the motor shaft.

The flux ϕ is assumed to remain constant to make the equation simple and to use the Laplace transform [22]; hence

$$e_m(t) = K_m \frac{d\theta}{dt} \tag{5.4}$$

The Laplace transform of Eqn. (5.4) yields

$$E_m(s) = K_m s\Theta(s) \tag{5.5}$$

The armature voltage of the circuit is derived in Laplace transform as

$$E_a(s) = (L_m s + R_m s) I_a(s) + E_m(s) \tag{5.6}$$

Equation (5.6) can be solved for $I_a(s)$ as

$$I_a(s) = \frac{E_a(s) - E_m(s)}{L_m s + R_m s} \tag{5.7}$$

The equation for the developed torque is

$$\tau(t) = K_1 \phi \, i_a(t) = K_\tau i_a(t) \tag{5.8}$$

since flux is assumed constant. The Laplace transform of this equation yields

$$T(s) = K_\tau I_a(s) \tag{5.9}$$

The final equation is derived from summing the torques on the motor armature. In Fig. 5.12, the moment of inertia J includes all inertia connected to the motor shaft, and B includes the air friction and the bearings friction. Therefore, the torque equation is

$$J\frac{d^2\theta}{dt^2} = \tau(t) - B\frac{d\theta}{dt} \tag{5.10}$$

and thus, the Laplace transform of this equation is derived for the torque as

$$T(s) = (Js^2 + Bs)\Theta(s) \tag{5.11}$$

The motor shaft angle yields

$$\Theta(s) = \frac{T(s)}{Js^2 + Bs} \tag{5.12}$$

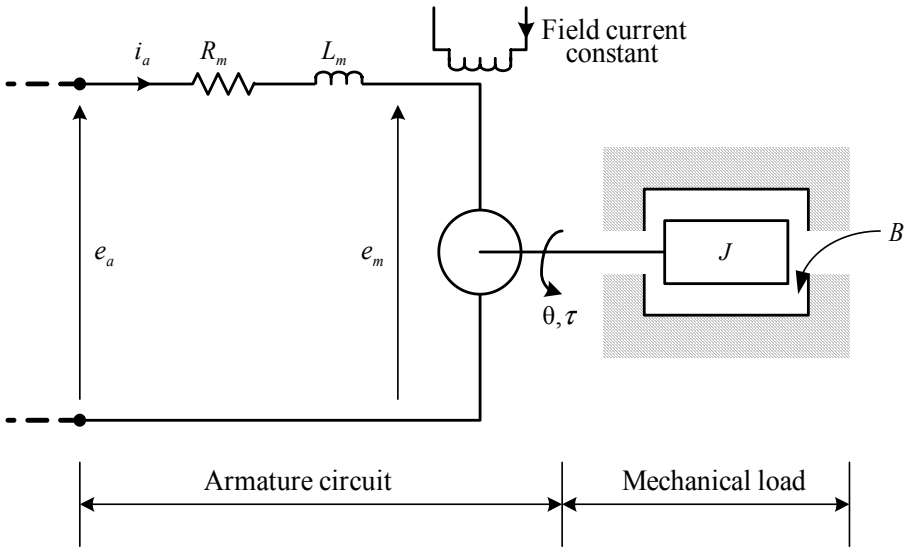


Figure 5.12 Servomotor

A block diagram can be constructed from the Eqns. (5.5), (5.7), (5.9) and (5.12), and is illustrated in Fig. 5.13.

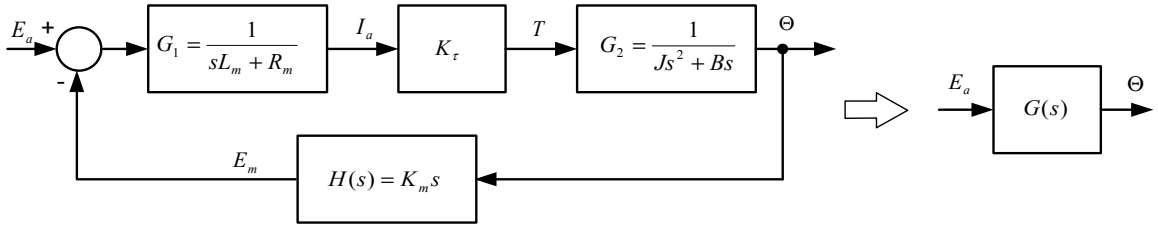


Figure 5.13 Block diagram of the servomotor

From Mason's gain formula the motor transfer function is derived as

$$G(s) = \frac{\Theta(s)}{E_a(s)} = \frac{G_1(s)K_\tau(s)G_2(s)}{1 + K_\tau G_1(s)G_2(s)H(s)} \tag{5.13}$$

Evaluating this expression yields

$$G(s) = \frac{K_\tau}{JL_m s^3 + (BL_m + JR_m)s^2 + (BR_m + K_\tau K_m)s} \tag{5.14}$$

The armature inductance L_m is often ignored in the literature where L_m is small enough. Thus, the transfer function is rewritten as

$$G(s) = \frac{K_\tau}{JR_m s^2 + (BR_m + K_\tau K_m)s} \tag{5.15}$$

Note that this transfer function depends upon the inertia and friction of the load being driven by the motor, as well as the motor parameters.

5.3.1.1 Computer Simulation Studies

The objective of this simulation is to make the servomotor follow the reference model. The reference model generated by the three neural components shown as Fig. 5.11 decides the speed and direction of the mobile robot. In this simulation, the static and dynamic neural units with CSO were used to control the motor to follow the reference

model. The block diagram of motor control is given as Fig. 5.14. In the figure, Ref is the reference input, C is the output of the motor, y_{ref} is the output of the reference model and E is the error between the input and output.

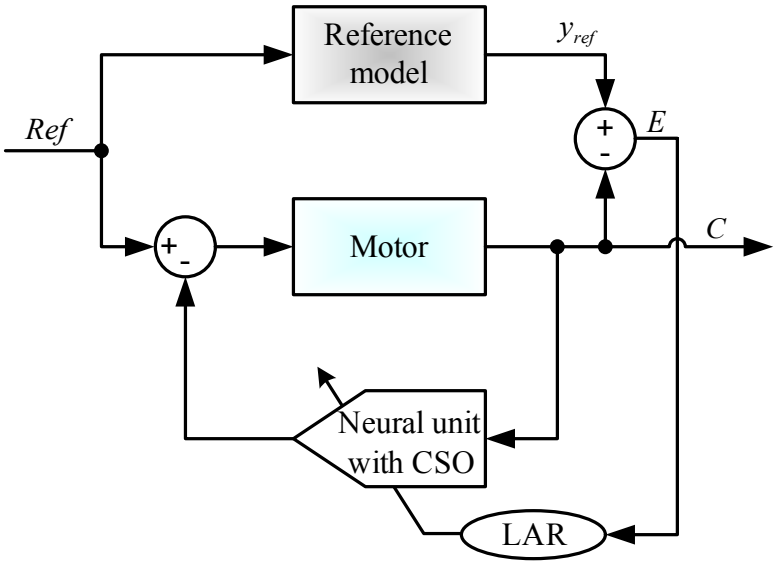


Figure 5.14 Block diagram of the neural motor control with CSO

The physical parameters of the motor are given in Table 5.4.

Table 5.4 The physical parameters of the servomotor

Physical parameters	Value
Resistance (R_m)	4 Ohm
Motor parameters (K_τ, K_m)	0.0274 Nm/Amp
Moment of inertia (J)	3.2284E-6 $kgm^2/\sec^2$
Friction (B)	3.5077E-6 Nms

With the physical parameters of the motor, the Laplace transform of the servomotor is derived from Eqn. (5.15) as

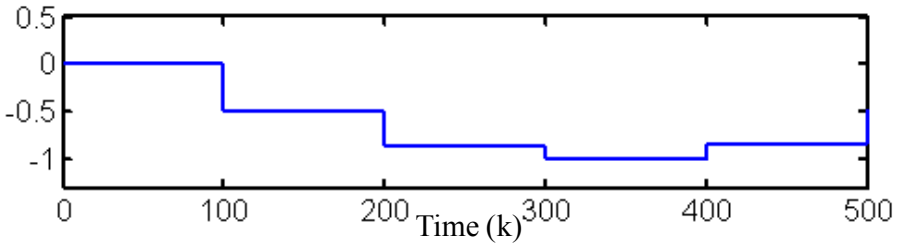
$$G(s) = \frac{2122}{s^2 + 59.22s} \tag{5.16}$$

or in discrete-time system with the sampling time 0.01 as

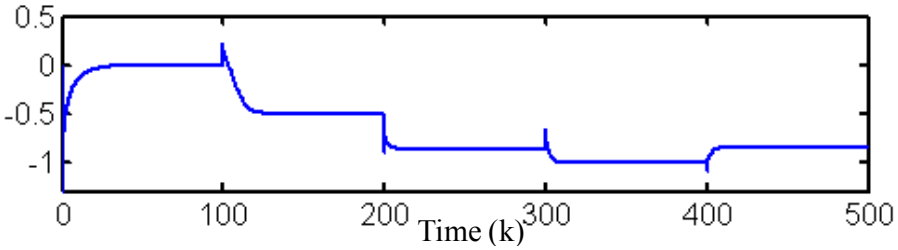
$$G(z) = \frac{0.08972z + 0.07221}{z^2 - 1.553z + 0.5531} \tag{5.17}$$

Case 1: Motor control with static neural unit with CSO

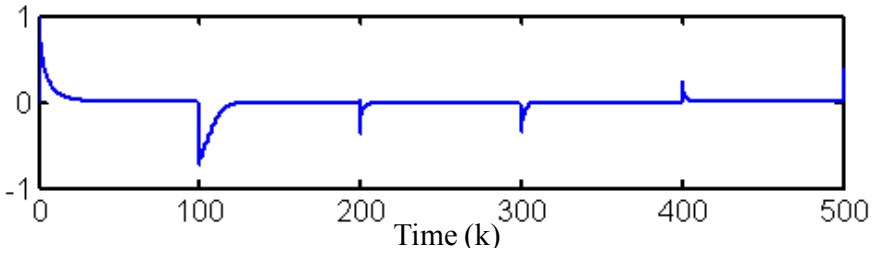
In this case, the static neural unit with CSO was used to control the servomotor. The learning rate μ was 0.01. The reference output was changed at every 100 steps. Figure 5.15 shows the result of the computer simulation in Case 1. It is observed that the static neural unit was initially learning the reference model, and after the learning procedure the motor was able to be controlled corresponding to the reference model. The trajectory of the motor describes the learning procedure. Whenever the signal was altered, the static neural unit with CSO took some time to learn the new change, and then the motor could follow the model in no time as the error shows.



(a) Output of the reference model



(b) Output of the servomotor controlled by the static neural unit with CSO

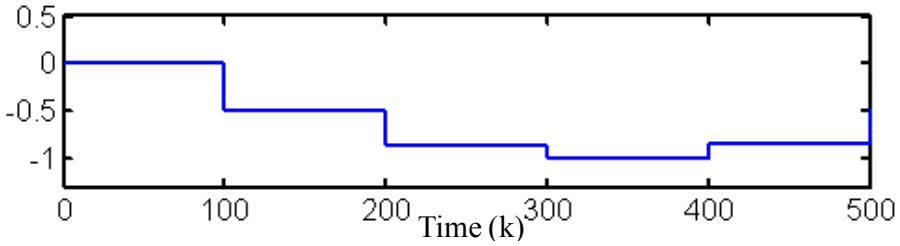


(c) Error between the reference model and the servomotor

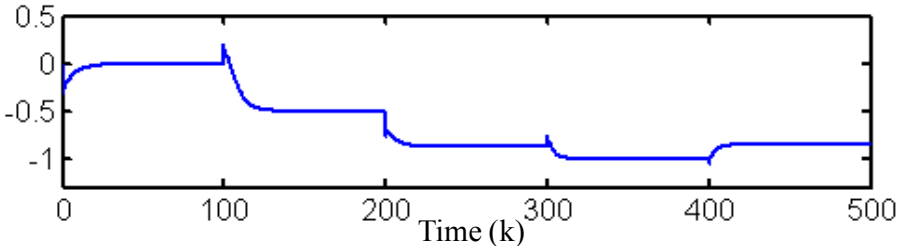
Figure 5.15 Case 1: Motor control with a static neural unit with CSO

Case 2: Motor control with dynamic neural unit with CSO

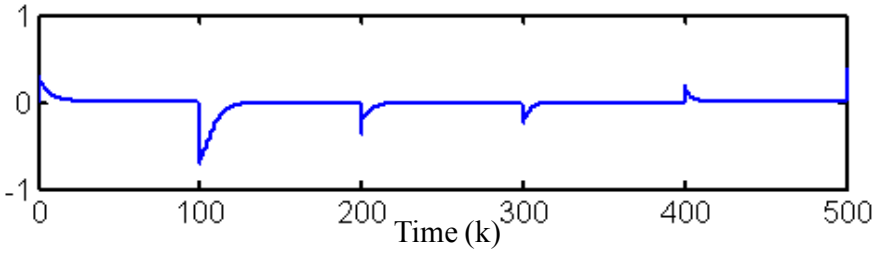
In this case of study, the dynamic neural unit with CSO was used to control the servomotor. The learning rate μ was 0.01 as same as that of the static neural unit with CSO. The change of the reference output was the same as the change in Case 1. The computer simulation with the dynamic neural unit with CSO is shown in Fig. 5.16. It is observed that the motor output could follow the reference output after several seconds. Unlike from the static neural controller, the initial trajectory of the motor shows the high overshoot due to the dynamic structure. However, the overshoot becomes less later on showing that the dynamic neural controller was adapted to the reference model.



(a) Output of the reference model



(b) Output of the servomotor controlled by the dynamic neural unit with CSO



(c) Error between the reference model and the servomotor

Figure 5.16 Case 2: Motor control with a dynamic neural unit with CSO

These simulation studies showed that initially the dynamic neural controller took a bit more time and had higher overshoot than the static neural controller; however, the dynamic neural controller took less time than the static neural controller to adapt to the new environments after the learning and adaptation. On the other hand, the dynamic neural controller showed a bit higher overshoot than the static neural controller. The high computation in the synaptic operation may cause the static and dynamic neural units with CSO to be more sensitive to the systems. In particular, the dynamic neural unit with CSO

contains the dynamic structure which makes the neural structure more complex and sensitive. The sensitivity of the dynamic neural structure may cause the controller to manage the motor with some overshoot and a little sluggish response.

5.4 Summary

In this chapter, a mobile robot was controlled by neuro-vision and neuro-control systems. The HT and the estimation of location were used for an advanced neuro-vision system, and the static and dynamic neural units with CSO were applied for the neural controller. In order to control the mobile robot, many complex problems need to be solved due to the nonlinear environments affecting the STM and the LTM of the machine. The simulation studies demonstrate the adaptive capability of the neural structures of the robot (with respect to the neuro-control system) when the robot is subjected to disturbance by various objects in its paths. The higher performance of the neural unit with CSO was able to control the robot effectively.

In addition, the studies show that the dynamic nature of the neural units causes both advantages and disadvantages in their performance. The superiority of the dynamic structure was demonstrated in the previous chapter. However, usage of the dynamic structure should be selective according to the purpose of the systems because the dynamic structure may damage some performance of the systems.

Chapter 6

Conclusions

6.1 Concluding Remarks

The study of the architecture of biological neurons has influenced the development of the neural structures and their computational process. Based on the neural units, various neural network applications such as function approximation, pattern recognition, and system identification and control have been studied in the literature [17]. Currently, the neural unit with *linear synaptic operation* (LSO) is used in neural networks. However, the linear model of the neural units ignores some of the significant features of the biological neurons, such as the higher-order synaptic computation. The motivation to emulate the superior performance of the biological neurons expands the neural units with LSO to the neural units with *higher-order synaptic operation* (HOSO).

The novel structure of the neural units with HOSO is given in the forms of the neural units with *quadratic synaptic operation* (QSO) and *cubic synaptic operation* (CSO) in Chapter 2. The topology of the neural units with HOSO is based on the highly amalgamated inputs in the synapse of the biological neurons. It was proved that the neural unit with LSO is a subset of the neural units with HOSO in this chapter. The neural units with HOSO emulate some of the important properties of the biological neurons, and perform more efficiently than the neural units with LSO. The **XOR** logic problem gave a clue to the advanced performance of the neural units with HOSO. A

neural unit with HOSO was able to classify the nonlinearity of the **XOR** logic problem. The generalization of the neural units with HOSO is explained at the end of this chapter.

In Chapter 3, the neural unit with CSO was incorporated with the dynamical structure followed by the nonlinear activation function. The neural structure with feedback was named the dynamic neural unit with CSO. A delay with feedback was implemented as the dynamic structure to emulate the memory activities of the brain. There is one adaptive parameter b in the dynamic structure. Furthermore, the novel neural networks were created consisting of the neural unit with CSO. In order to adapt the neural networks, the current back-propagation algorithm used for the neural units with LSO was modified for the neural networks composed of the neural units with CSO. In this chapter, industrial motion control systems were introduced for the model identification simulation. The simulation indicates the advantage of the neural units with HOSO, as well as the dynamic structure in the neural units. In the literature, the neural networks composed of the neural units with LSO accomplish the model identification; however, one neural unit with HOSO was able to perform the same execution as the neural networks composed of the neural units with LSO. Additionally, the dynamic structure effectively influenced the neural performance. The dynamic neural units with HOSO identified the motion systems with less error than the static neural units with HOSO.

A novel edge detector was introduced with a neural approach in Chapter 4. The technique of the neural edge detector was based on the concept of the biological vision procedure which contains the processing of light stimuli to the retina. The neural units with LSO, QSO and CSO were applied as the neural edge detectors, and two different figures were used for the detectors to sense the edges. This simulation implies the capability of the

neural structures for image processing. As a result, the neural edge detector with the CSO generated more clearly defined edge lines than other neural detectors. The three-dimensional edge detected plots demonstrates the enhanced performance of the neural edge detectors. Overall, this simulation proved the superior capability of the neural units with HOSO for image processing.

A mobile robot was introduced for a control application in Chapter 5. In order to control the mobile robot, three neural categories were applied: neural function approximation as classification, neuro-vision system as signal processing and neuro-control system as control. These three components were correlated and affected other aspects of the robot's performance. A neural processor for the classification of sensory data was considered as the *short term memory* (STM), and the neuro-vision system was regarded as the *long term memory* (LTM) of the machine. The *Hough Transform* (HT) and the algorithm of movement decision were used for the advanced presentation of the neuro-vision system. With sensors, the neuro-vision system takes several roles such as position calibration, obstacle avoidance, and motor speed control. From the simulation results, the controller with the neural unit with CSO was able to handle the mobile robot with the STM and the LTM.

6.2 Conclusions

Inspired by the structure of biological neurons, the novel static and dynamic neural units with HOSO have been proposed in this thesis. The neural units with HOSO are more powerful and efficient due to the nonlinear computation of neural inputs and neural synaptic weights. Thus, the neural units with HOSO are well suited to control the

complex (nonlinear) systems. The higher computation of synaptic operation in the neural units with HOSO causes the neural networks to have less number of neurons, which enables the neural networks to save a lot of time in the learning and adaptation. Image processing and motion control with the neural units with HOSO have proved the potential applications of the novel neural units.

At the same time, the exponential complexity due to the increasing number of synaptic weights is a major concern. Hence, the order of synaptic operation should be decided in relation to the problems to be solved.

6.3 Directions for Future Research

In this thesis, only theoretical study and some computer simulations have been carried out. In order to extend the knowledge and area of the novel neural structures, it will be useful to implement the neural units with HOSO in the real-time applications. In addition, a detailed theoretical analysis of the dynamic structure in the dynamic neural units with HOSO is required in the areas of stability, convergence and flexibility. It is interesting to note that fuzzy logic control is another powerful tool for modeling uncertainties associated with human thinking and perception [5]. It is believed that the fuzzy neural networks have considerable potential in the area of expert systems, medical diagnosis, computer vision, pattern recognition, and system modeling and control [13]. It would be very interesting and challenging to develop the fuzzy neural network. Also, it would be very useful to conduct further studies in the field of medical diagnosis and prognosis with the help of these innovative neural structures with HOSO neural units.

References

[1] M.M. Gupta, *"Neural Computing System",* Class notes, Neural Networks, Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, 2000.

[2] J.M. Zurada, *"Introduction to Artificial Neural Systems",* West Publishing Company, St. Paul, MN, 1992.

[3] M.M. Gupta, L. Jin, and N. Homma, *"Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory",* John Wiley & Sons, Hoboken, New Jersey, 2003.

[4] S.W. Kuffler, J.G. Nicholls and A.R. Martin, *"From Neuron to Brain: A Cellular Approach to the Function of the Nervous System",* Sinauer Associates, Sunderland, MA, 1984.

[5] Y. Song, *"Development of Dynamic Neural Units with Control Applications",* MSc. Thesis, University of Saskatchewan, Saskatoon, 2001.

[6] K.J. Hunt, D. Sbarbaro, R. Zbikowski and P.J. Gawthrop, *"Neural Networks for Control Systems – A Survey"*, Automatica, Vol.28, No.6, pp.1083-1112, 1992.

[7] N.A. Deshpande, *"Development of Dynamic Neural Networks with Control Applications"*, MSc. Thesis, University of Saskatchewan, Saskatoon, 1997.

[8] D.H. Rao, *"Development of Dynamic Neural Structures with Control Applications",* Ph.D. Thesis, University of Saskatchewan, Saskatoon, 1994.

[9] J.C. Principe, N.R. Euliano, and W.C. Lefebvre, *"Neural and Adaptive Systems: Fundamentals through Simulations"*, John Wiley & Sons, New York, NY, 2000.

[10] K. Fukushima, S. Miyake and T. Ito, *"Neocogniron: A Neural Network Model for a Mechanism of Visual Pattern Recognition",* IEEE trans. Systems, Man and Cybernetics, Vol.13, No.5, pp.826-834, September/October 1983.

[11] J.J. Hopfield, *"Neurons with Graded Response Have Collective Computational Properties Like of Those Two-State Neurons",* Proc. of the National Academy of Sciences, Vol.81, pp.3088-3092, 1984.

[12] K.S. Narendra and K. Parthasarthy, *"Identification and Control of Dynamical Systems Using Neural Networks",* IEEE Trans. Neural Networks, Vol.1, No.1, pp.4-27, March 1990.

[13] M.M. Gupta and D.H. Rao, *"Neuro-Control Systems: Theory and Applications",* IEEE press, pp.1-43, 1994.

[14] F.C. Chen, *"Backpropagation Neural Networks for Nonlinear Self-Tuning Adaptive Control"*, IEEE Control System Magazine, pp.44-48, April 1990.

[15] L. Jin, P.N. Nikiforuk and M.M. Gupta, *"Adaptive Tracking of SISO Nonlinear Systems Using Multilayered Neural Networks",* Proc. 1992 American Control Conference (ACC), pp.56-60, June 1992.

[16] S.K. Redlapalli, M.M. Gupta and K.-Y. Song, *"Development of Quadratic Neural Unit with Applications to Pattern Classification",* 4th International Symposium on Uncertainty Modeling and Analysis, pp.141-146, September, 2003.

[17] J.A. Anderson, *"Cognitive and Psychological Computation with Neural Models",* IEEE Trans. Systems, Man and Cybernetics, Vol.13, pp.799-815, 1983.

[18] M.M. Gupta and D.H. Rao, *"Dynamic Neural Units in the Control of Linear and Nonlinear Systems"*, Int. Joint Conference on Neural Networks (IJCNN), pp.100-105, Baltimore, June 9-12, 1992.

[19] M.M. Gupta and D.H. Rao*, "Dynamic Neural Units with Applications to the Control of Unknown Nonlinear Systems",* The Journal of Intelligent and Fuzzy systems, Vol.1, No.1, pp.73-92, January 1993.

[20] K.S. Narendra and R.M. Wheeler, Jr., *"Recent Advances in Learning Automata",* Adaptive and Learning Systems, K.S. Narendre, Ed., Plenum Press, New York, 1985.

[21] K. Ogata, *"Modern Control Engineering",* 4th edition, Prentice Hall, Inc., New Jersey, 2002.

[22] C.L. Phillips and R.D. Harbor, *"Feedback Control Systems",* 4th edition, Prentice Hall, Inc., New Jersey, 2000.

[23] B.C. Kuo, *"Discrete-Data Control Systems",* Prentice Hall, Inc., New Jersey, 1970.

[24] A. Cichocki and R. Unbehauen, *"Neural Networks for Optimization and Signal Processing"*, John Wiley & Sons, New York, NY, 1993.

[25] S. Haykin, *"Neural Networks: A Comprehensive Foundation",* Prentice Hall, Inc., New Jersey, 1999.

[26] R.J. Williams and J. Peng, *"An Effective Gradient-based Algorithm for On-line Training of Recurrent Network Trajectories",* Neural Computation, Vol.2, pp.490-501, 1990.

[27] B. Kosko, *"Neural Networks for Signal Processing",* Prentice Hall, Inc., New Jersey, 1992.

[28] Y.T. Zhou and R. Chellappa, "*Artificial Neural Networks for Computer Vision",* Springer-Verlag, New York, 1992.

[29] M.M. Gupta and G.K. Knopf, *"Neuro-Vision Systems: Principles and Applications",* IEEE press, New York, 1993.

[30] M.M. Gupta, *"Neuronal Morphology of Biological Vision and Machine Vision Systems"*, Short course notes for OPTCON '92, Hynes Convention Center Boston, Massachusetts, USA, 1992.

[31] C. Blackemore, *"Mechanics of the Mind",* Cambridge: Cambridge University Press, 1977.

[32] L. Uhr, *"Psychological Motivation and Underlying Concepts",* in Structured Computer Vision, S. Tanimoto and A. Klinger, Ed., New York: Academic Press, pp.1-30, 1980.

[33] J.-K. Paik, J.C. Brailean and A.K. Katsaggelog, *"An Edge Detection Alogorithm Using Multi-State Adalines"*, Pattern Recognition, Vol.25, No.12, pp.1495-1504, 1992.

[34] K.-Y. Song, S.K. Redlapalli, and M.M. Gupta, *"Cubic Neural Unit for Control Applications",* 4th International Symposium on Uncertainty Modeling and Analysis, pp.324-329, September, 2003.

[35] J.D. McCafferty, *"Human and Machine Vision: Computing perceptual organisation"*, Ellis Horwood, West Sussex, England, 1990.

[36] E.R. Davies, *"Machine Vision: Theory, Algorithms, Practicalities"*, Academic Press, San Diego, CA, 1990.

[37] N. Allinson, H. Yin, L. Allinson and J. Slack, *"Advances in Self-Organising Maps"*, Springer, New York, 2001.

[38] K. Hornik, M. Stinchcombe and H. White, *"Multi-layer Feed Forward Networks are Universal Approximators"*, Neural Networks, Vol.2, pp.359-366, 1989.

[39] K. Funahashi, *"On the Approximate Realization of Continuous Mappings by Neural Networks"*, Neural Networks, Vol.2, pp.183-192, 1989.

[40] T. Poggio and F. Girosi, *"Networks for Approximation and Learning"*, Proc. IEEE, Vol.78, No.9, pp.1481-1497, September 1990.

[41] N.E. Cotter, *"The Stone-Weierstrass Theorem and Its Applications to Neural Networks"*, IEEE Trans. on Neural Networks, Vol.1, No.4, pp.290-295, 1990.

[42] M.M. Gupta, *"Biological Basis for Computer Vision: Some Perspectives"*, SPIE's Advances in Intelligent Robotics, Vol. 1192, pp. 811-823, 1989.

[43] R. Griñó, G. Cembrano, and C. Torras, *"Nonlinear System Identification Using Additive Dynamic Neural Networks: Two On-Line Approaches"*, IEEE Trans. on Circuits and Systems I-fundamental Theory and Application, Vol. 47, pp. 150-165, 2000.

[44] W.-H. Lee, K.-S. Roh, and I.-S. Kweon, *"Self-localization of a Mobile Robot without Camera Calibration using Projective Invariants"*, Pattern Recognition Letters, Vol. 21, pp. 45-60, 2000.

[45] R.A. Brooks, *"A Robust Layered Control System for a Mobile Robot"*, IEEE J.Robot. Automat., Vol. RA-2, pp. 14-23, April, 1986.

[46] Y. Nakamura, *"Advanced Robotics: Redundancy and Optimization"*, Reading, MA: Addison-Wesley, 1991.

[47] M.H. Robert and G.S. Linda, *"Computer and Robot Vision"*, Reading, MA: Addison-Wesley, 1993.

[48] R. Sim and G. Dudek, *"Learning visual landmarks for pose estimation"*, 1999 IEEE Int. Conf. Robotics and Automation, pp. 1972-1978, May, 1999.

[49] Z.-G. Hou, K.-Y. Song, and M.M. Gupta, *"Higher-Order Neural Units for Image Processing and Their Applications to Robot Routing Problems"*, 6th International Fuzzy Logic and Intelligent technologies in Nuclear Science (FLINS) 2004.