A CEREBELLUM-LIKE LEARNING MACHINE

by

ROBERT DUNCAN KLETT

B.A.Sc., University of British Columbia, 1975

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Department of Electrical Engineering)

We accept this thesis as conforming
to the required standard.

THE UNIVERSITY OF BRITISH COLUMBIA

July, 1979

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the Head of my Department or by his representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering

The University of British Columbia
2075 Wesbrook Place
Vancouver, Canada
V6T 1W5

Date July 19, 1979

ABSTRACT

This thesis derives a new learning system which is presented as both an improved cerebellar model and as a general purpose learning machine. It is based on a summary of recent publications concerning the operating characteristics and structure of the mammalian cerebellum and on standard interpolating and surface fitting techniques for functions of one and several variables. The system approximates functions as weighted sums of continuous basis functions. Learning, which takes place in an iterative manner, is accomplished by presenting the system with arbitrary training points (function input variables) and associated function values. The system is shown to be capable of minimizing the estimation error in the mean-square-error sense. The system is also shown to minimize the expectation of the interference, which results from learning at a single point, on all other points in the input space. In this sense, the system maximizes the rate at which arbitrary functions are learned.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

NOTATION

a) Standard Forms

a = a scalar

A = a vector

$\mathbf{A}$ = a matrix

a_i = the i^{th} element of A

a_{ij} = the element of $\mathbf{A}$ at co-ordinates (i,j)

A^T or $\mathbf{A}^T$ = the transpose of A or $\mathbf{A}$ respectively

b) Definition of Variables

$\triangle e$ = an error correcting function

ϵ = an arbitrary constant, $\epsilon \geq 0$
used as the acceptable estimation error

$\hat{f}$ = Purkinje cell activity or estimated output function

$\overline{f}$ = system corrected output
(Subcortical Nuclear cell output)

$f(H)$ = target function

$\triangle f$ = $f - \hat{f}$ = the estimation error

$\mathbf{G}$ = Golgi feedback matrix

H = region of definition in the hyperspace of the control function

H_1 = a particular point in H

I = identity matrix

k = convergence gain factor

NOTATION (Continued)

L = number of elements in the basis set $\{\Phi\}$ and in the weight set $\{\Pi\}$

m = number of elements in a one-variable interpolation function

n = number of independent input variables or arbitrary power

$\{\Phi\}$ = the function-space basis set (Parallel fiber activity)

$\{\Phi(H_1)\}$ = the vector of values of $\{\Phi\}$ at the point H_1

$\{\Pi\}$ = expanded input set

$s(H)$ = a cost density function

Θ = $(I+G)^{-1}$ = Granule-Golgi transfer matrix also used as an arbitrary non-singular matrix

$u(H)$ = the training point density function

v = arbitrary scalar for optimization techniques

$\{W\}$ = the set of variable weights (effective synaptic weights)

X = augmented matrix whose columns are eigenvectors

x_i = the i^{th} element of the input set

ACKNOWLEDGEMENT

There are many people and institutions to whom I owe much gratitude on the completion of this thesis.

My wife, Marian, has shown great patience and understanding while encouraging me in this project.

It is with much gratitude that I acknowledge the assistance of my supervisor, Dr. P.D. Lawrence. His suggestions and criticism have added much to this work while his encouragement has helped to bring it to a successful conclusion.

I wish to thank my co-reader, Dr. E.V. Bohn, and the other members of my committee for their careful inspection and criticism of this thesis.

To my fellow students in Electrical Engineering at the University of British Columbia, I extend thanks for providing an enjoyable and stimulating environment. In particular, I thank Mr. Rolf Exner for developing the program which typed this thesis.

This research has been supported by the Natural Sciences and Engineering Research Council of Canada under a Postgraduate Scholarship to its author and grant No. A9341, and by the University of British Columbia in the form of a Teaching Assistanceship.

I INTRODUCTION

Artificial intelligence and pattern recognition are currently areas of great interest. This thesis analyses a phylogenetically lower portion of the brain, the cerebellar cortex, in the belief that such an analysis will assist the development of a new type of intelligent system. It is further believed, since the higher regions of the brain are phylogenetically newer than the lower ones, that a study of the cerebellum is a logical starting point in the study of intelligent structures in general.

The cerebellum is a part of the brain which is located in the rear portion of the head, at the base of the skull. It is generally accepted that the cerebellum is involved in maintaining posture and co-ordinating motor activities of the body [7,12,28]. Although other levels of the brain can perform these functions, it has been postulated that the cerebellum is designed both to provide finer co-ordination than other levels and to relieve higher levels of most of the motor tasks [17,18,46]. Studying the cerebellar cortex is particularly appealing due to its extremely regular arrangemant of cells (which will be discussed in Chapter 2) and the fact that its inputs (only two types) and outputs (only one type) are so well differentiated. Although the following analysis is based on the structure and function of the cerebellum, it should be established at this point that much additional research is required to show if the cerebellum in fact operates according to the theory to be developed.

There are thus two main goals of this research:

1. to present a model of the cerebellum which is consistent with current anatomical and physiological data; thus being an improvement over existing models, and
2. to develop a learning machine, incorporating the improved cerebellar model, which may form the basis of a new approach to designing intelligent machines.

With these goals in mind, the thesis begins by describing the structure of the cerebellum. Chapter 3 presents recent cerebellar models, some of their deficiencies, and the requirements of an improved model. Next, Chapter 4 begins to analyse the nature of possible cerebellar computations, leading to Chapter 5 which ends with the development of an optimized learning algorithm which is consistent with the cerebellar system. The performance and operating characteristics of examples of this system are presented in Chapter 6. Finally, Chapter 7 discusses implications of both the new cerebellar model and the learning system, ending with suggestions for further research.

II CEREBELLAR STRUCTURE

There have been numerous publications dealing with the cerebellum. Unless otherwise indicated, the following information is derived from some of those which deal with the cat [7,12,48]. A block diagram of the interconnections of neurons in the cerebellum is shown in Figure 2.1 in

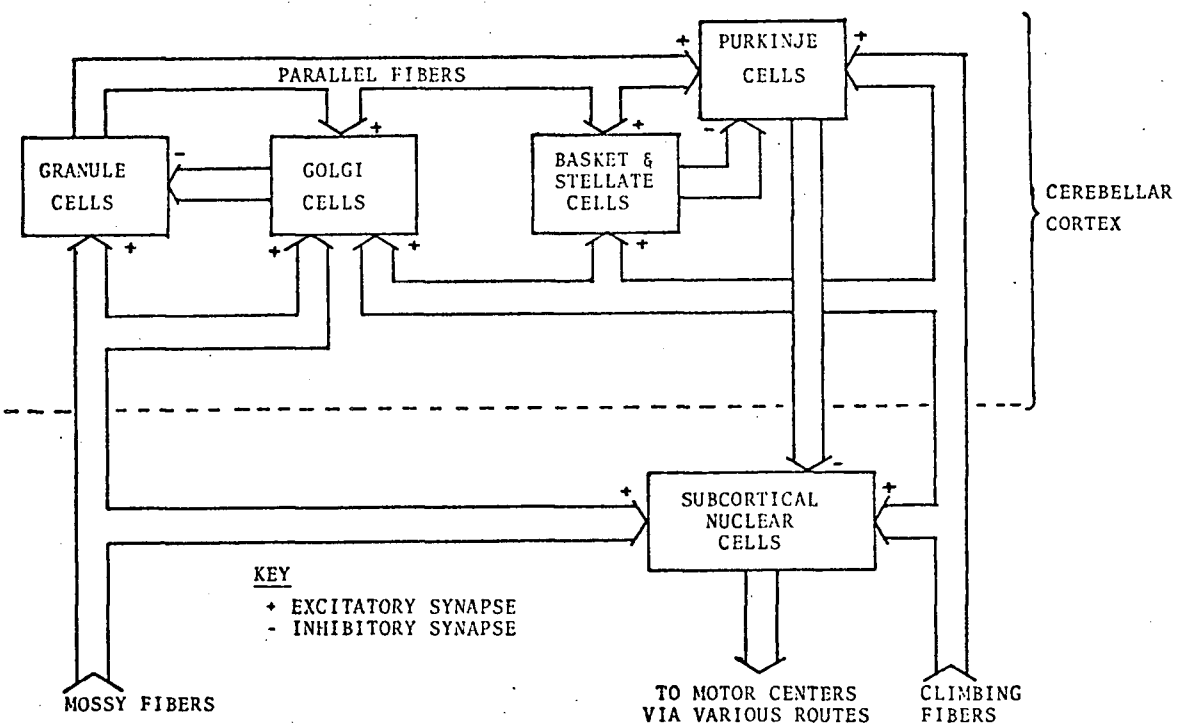


Fig 2.1 Block Diagram of the Cerebellar System

which the interconnections are marked to indicate whether they are excitatory or inhibitory. It can be seen that there are only six types of neurons in the cerebellum. Of these, five are interneurons located

in the cerebellar cortex which have no direct effect outside the cerebellum (Granule, Golgi, Basket, Stellate, and Purkinje cells), while the sixth (Subcortical Nuclear cells) are the only cells which do have a direct external effect. There are also only two types of input fibers (Mossy and Climbing fibers).

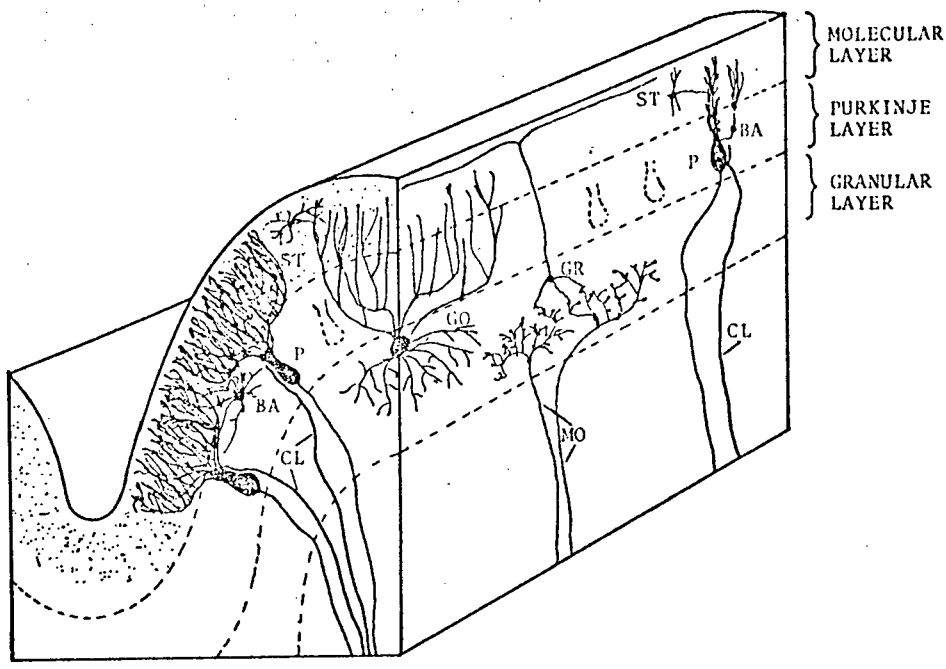
The interconnections of the cerebellar neurons, particularly those in the cerebellar cortex, have been extensively studied, revealing the remarkably regular geometry of Figure 2.2.

2.1 The Mossy Fiber Pathway

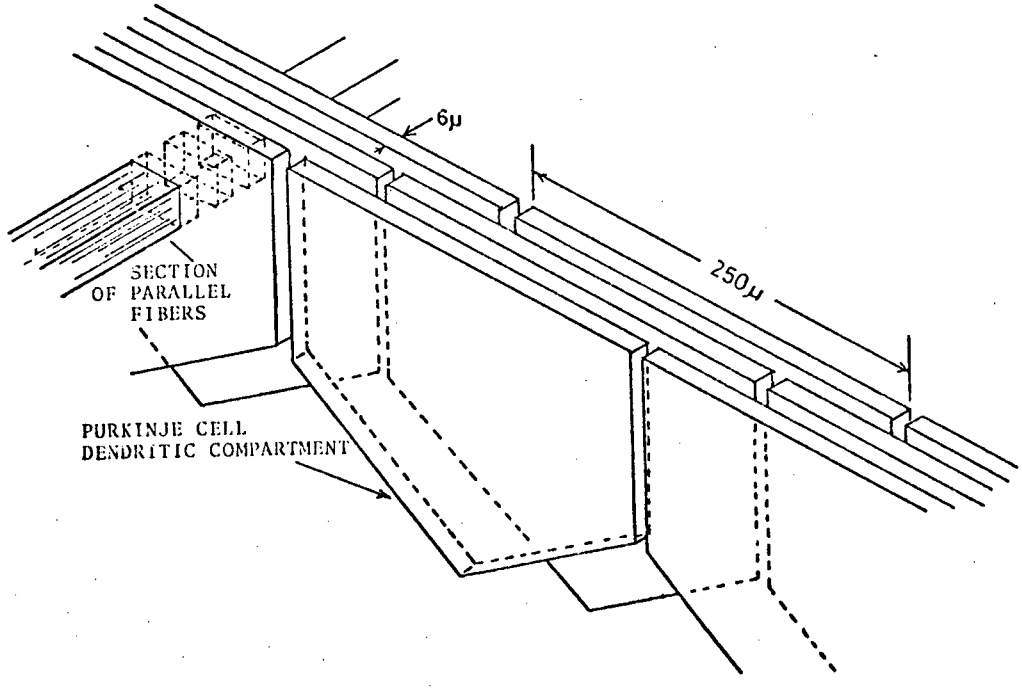
The effects of Mossy fiber activity upon the cerebellum are very widespread. Upon entering the cerebellum, each Mossy fiber sends branches to numerous folia throughout the cerebellum. Some collaterals terminate on Subcortical Nuclear cells, while the branches which enter the cortex branch profusely before terminating as Mossy Rosettes in the Granular layer of the cerebellar cortex. The Rosettes form the nucleus of an excitatory synapse between a Mossy fiber, dendrites from numerous Granule cells, and dendrites from a few Golgi cells.

Granule cells, each of which are contacted by 2 to 7 Mossy Rosettes (average 4.2), generate a single axon which climbs through the cerebellar folia to the Molecular layer where it forms a "T"-shaped branch. Each branch of the axon, now classed as a Parallel fiber, runs longitudinally along the folia for a distance of up to 1.5 mm (2.5 mm in man [9]).

Each Parallel fiber traverses a large number of characteristically flattened dendritic trees of Basket, Stellate, and Purkinje cells (up to



a) Perspective view of the cerebellar cortex (after [12]). BA=Basket cell, CL=Climbing fiber, GO=Golgi cell, GR=Granule cell, MO=Mossy fiber, P=Purkinje cell, ST=Stellate cell.



b) Diagramatic view of the Molecular layer of the cerebellar cortex showing the packing arrangement of Parallel fibers and Purkinje cell dendrites.

Fig 2.2 Structure of the Cerebellar Cortex

300 of the latter). Although there is some uncertainty concerning the proportion of Parallel fiber-Purkinje cell intersections which in fact result in synapses, it is clear that the geometry maximizes the number of possible synapses. Those intersections which do result in synapses between Parallel fibers and Purkinje cells, Basket cells, or Stellate cells are excitatory. The axons of Basket and Stellate cells form inhibitory synapses with the dendritic trees and pre-axon areas of nearby Purkinje cells. Purkinje axons terminate in inhibitory synapses on Subcortical Nuclear cells.

As well as forming excitatory synapses with Basket, Stellate, and Purkinje cells, Parallel fibers also form excitatory synapses with the dendritic trees of Golgi cells. Unlike the dendritic trees of the other cerebellar interneurons, Golgi cell dendrites spread throughout a cylindrical volume whose base is in the Granular layer and whose top is in the Molecular layer of the cerebellar cortex. Each tree is divided into two regions, the top one being excited by Parallel fibers, the bottom one by Mossy Rosettes. Each Golgi cell generates axons which inhibit a large fraction of the Granule cells which are in the volume enclosed by the Golgi dendritic tree.

The capacity of the cerebellum to process (co-ordinate) information is often discussed in terms of the divergence and convergence of information carrying neurons [13,14,28]. Table 2.1 illustrates these properties. Of particular note is the remarkably large number of Parallel fibers (up to 200 000) which synapse with each Purkinje cell. This convergence factor is a direct consequence of long, thin, Parallel fibers forming a lattice with fan-shaped dendritic trees of Purkinje cells in the Molecular layer of the cerebellar cortex.

Table 2.1 Connectivity of Cerebellar Neurons in Cat

a) Mossy Fiber Pathway

	DIVERGENCE	CONVERGENCE
MOSSY FIBERS		
↓	460[14]-600[13]	4.2[14]
GRANULE CELLS		
PARALLEL FIBERS		
↓	100[14]-300[13]	100 000[14]-200 000[13]
PURKINJE CELLS		
PARALLEL FIBERS		
↓	20[14]-30[13]	20 000[14]
BASKET CELLS		
↓	8[14]-50[13]	20[13]-50[14]
PURKINJE CELLS		

b) Climbing Fiber Pathway

	DIVERGENCE	CONVERGENCE
CLIMBING FIBERS		
↓	10[14]	1[14]
PURKINJE CELLS		

eg. Each Mossy fiber makes synaptic contact with between 460 and 600 Granule cells while each Granule cell is contacted by an average of 4.2 Mossy fibers.

2.2 The Climbing Fiber Pathway

In contrast to Mossy fibers, the effects of Climbing fiber activity are very localized. Although collaterals which form excitatory synapses with Subcortical Nuclear cells have been found, Climbing fibers branch very little after entering the cerebellum. Each Climbing fiber which enters the cerebellar cortex typically forms a synapse with only one, or with at most a few, Purkinje cells. This synapse, which engulfs the dendrites and cell body of the Purkinje cell is very strongly excitatory. Climbing fibers also form excitatory synapses with the Basket and Stellate cells which are in close proximity to the target Purkinje cell. The relation of Climbing fibers to Golgi cells is less well understood.

III MODELING THE CEREBELLUM

3.1 Recent Cerebellar Models

A number of theories have been proposed which explain some aspects of cerebellar function and organization. Unfortunately, most of the theories do not acceptably explain the mathematics of a learning algorithm which must be the basis of a truly valid model. Alternatively, those papers which do describe workable learning algorithms are not fully compatible with known cerebellar structure.

In one of the first theories to utilize current knowledge of cerebellar structure, Marr [28] proposes that the cerebellum directs a sequence of elemental movements to generate a desired action. That is, the cerebellum acts as a pattern recognition device, relating patterns of Mossy fiber activity to learned outputs. The recognition is performed according to a "codon" (subset) of Parallel fibers which are active at a given time. Translating this into mathematical terminology, his proposal is that each Purkinje cell separates a binary hyperspace of Parallel fiber activity into linearly separable regions in which the Purkinje cell is either active or inactive. The orientation of the separating hyperplanes is learned by adjusting Parallel fiber-Purkinje cell synaptic strengths as a function of Climbing fiber activity.

Albus [1,2] extended and modified Marr's theory by describing the cerebellum in terms of Perceptron theory [32,34,44,45]. A perceptron, which typically consists of binary inputs, a combinatorial network of "association cells", adjustable weights, and a summing device is shown

in Figure 3.1.

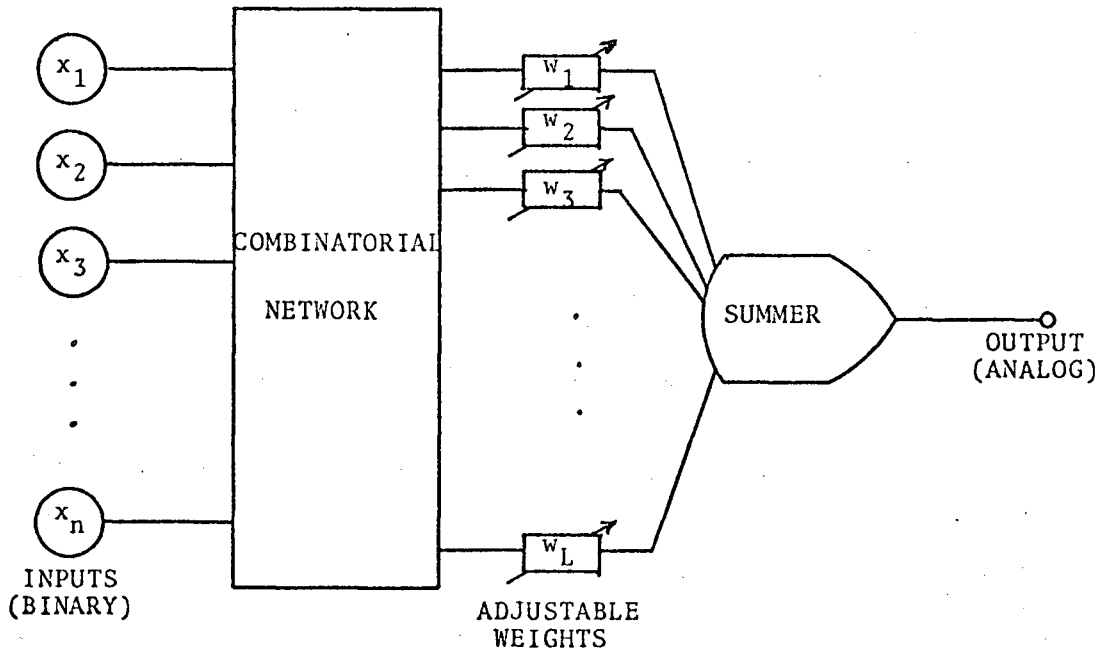


Fig 3.1 A Perceptron

In Albus' model, real-valued data, such as a joint angle, is converted to a set of binary signals in a manner similar to the mapping shown in Figure 3.2. He proposes that these signals are transmitted along Mossy fibers before being recoded by the Granular layer into patterns of Parallel fiber activity which form an expanded set of binary signals. The purpose of expansion recoding is to map all possible patterns of Mossy fiber activity into sets of Parallel fiber activity which are linearly separable. Weights of synapses between Parallel

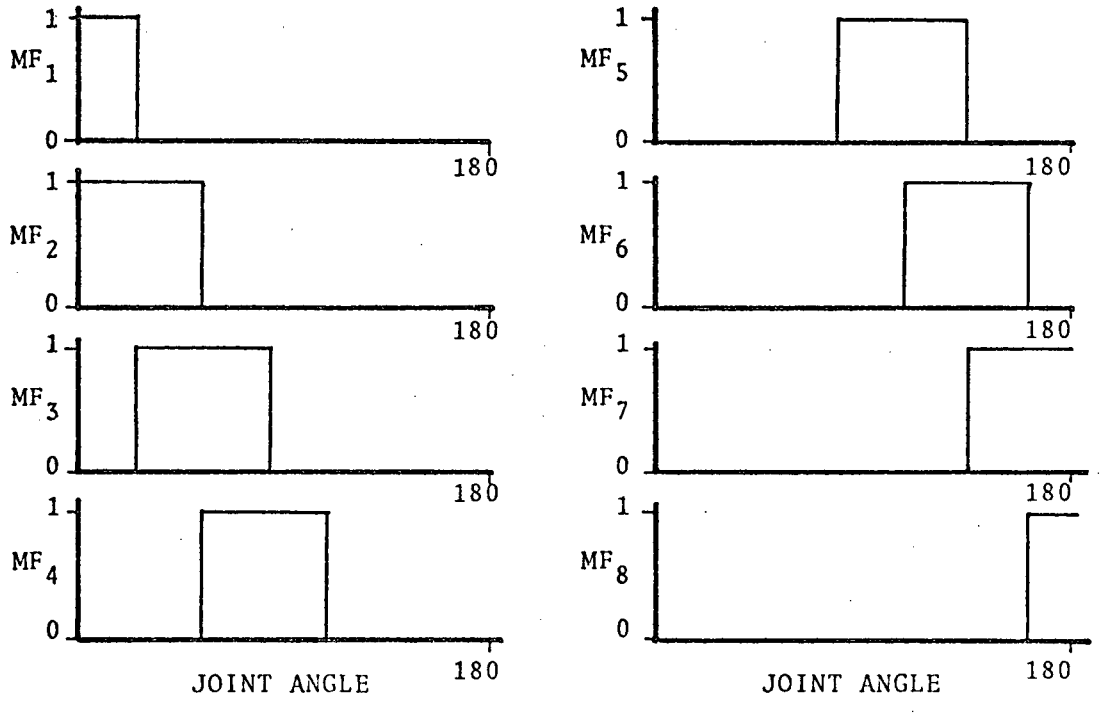


Fig 3.2 Albus' Mossy Fiber Activity Pattern

fibers and Purkinje cells, Basket cells, and Stellate cells are adjusted under the influence of Climbing fiber activity to obtain the desired Purkinje cell output.

The effectiveness of this model in performing cerebellum-like functions has been convincingly demonstrated by its ability to learn to control a number of movements of a mechanical arm [2,3,4].

Unfortunately, Albus' model is somewhat incompatible with some aspects of cerebellar structure and physiology. His expansion recoding scheme employs a hash-coding function which seems to be beyond the computational powers of the Granular layer. Furthermore, there is

little evidence to support his proposed mapping scheme of joint angle to Mossy fiber activity [22,23].

A model which is somewhat similar to Albus' has been proposed by Gilbert [17]. The model computes Purkinje cell activity as a weighted sum of Parallel fiber activity; the Parallel fiber-Purkinje cell and Basket and Stellate cell-Purkinje cell synapses being modifiable under the influence of Climbing fiber activity. The theory is consistent with cerebellar structure but does not describe the relation between the system's inputs, Mossy fiber activity, and Parallel fiber activity.

In another theory, Kornhuber [24] emphasizes the capacity of the cerebellar cortex to act as a timing device. He proposes that the action potential velocity along long, thin, Parallel fibers provides a timing mechanism for the control of ballistic motions such as saccadic eye movement. His model, although not rejecting it, does not propose a mechanism by which learning could take place.

Calvert and Meno [11] use spatiotemporal analysis to show that interconnections found in the cerebellum may cause it to act as an anisotropic spatial filter which enhances both the spatial and temporal details of Mossy fiber activity patterns. Their analysis assumes that all synaptic weights are determined strictly according to the type of pre and post-synaptic neuron. Hence, the model does not account for learning.

Hassul and Daniels [20] present a theory, along with supporting experimental evidence, that at least part of the function of the cerebellum is to act as a higher order lead-lag compensator. They further propose that this compensation provides stability for reflex actions. The model does not propose a scheme whereby the correct compensation might be learned.

A number of computer simulations of activity in a cerebellum-like network have been presented [33,38,39,40]. A common prediction of these models is that lateral inhibition and Golgi cell feedback should cause the surface of the cerebellum to contain long, narrow, bands of active Parallel fibers. That is, Mossy fiber activity is "focussed" into regions of Parallel fiber activity. A similar study includes peripheral feedback and muscle fibers in the model [37].

A common problem with these computer simulations is that they assume all Parallel fiber-Purkinje cell synaptic weights are equal. That is, the models do not consider the effects of unequal synaptic weights which might result from the application of a learning process.

3.2 Requirements of an Improved Cerebellar Model

The above cerebellar theories form a good starting point to begin understanding the cerebellum. Unfortunately, each theory is deficient in at least one aspect. Furthermore, the theories tend to be mutually exclusive so that an all encompassing one is not easily synthesized as a combination of the strong points of each proposal. A common shortcoming of those models which do propose a learning scheme is their treatment of Parallel fiber activity as a binary quantity. In these models, learning operates to form linearly separable regions in a binary hyperspace of Parallel fiber activity. Although each action potential is undoubtedly binary, the information carried by a nerve is generally accepted to be a function of the frequency of action potentials [53], not the presence or absence of one. An improved model must therefore deal with signals which are interpreted as real, not binary, variables.

To be truly valid, the operations of a cerebellar model must be consistent with those operations which are feasible to implement using the known structure of the cerebellum. In particular, the mapping of Mossy fiber activity to Parallel fiber activity must be consistent with the arrangement of cerebellar neurons, and the memory requirements of the learning algorithm must be consistent with the number of modifiable synapses in the cerebellar cortex.

IV MODEL SIMPLIFICATIONS AND ASSUMPTIONS

4.1 Arithmetic Functions of Cerebellar Neurons

The basic assumption to be used in the following discussion is that cerebellar information processing may be interpreted as a number of mathematical operators acting upon real-valued data. That is, data which is translated into a sequence of action potentials whose average frequency is a function of some biologically significant variable, is processed, according to common mathematical operations, by cerebellar neurons. It has been shown that neurons and neural networks, using frequency coded data, can perform basic arithmetic operations (add, subtract, multiply, and divide) upon pre-synaptic action potential frequencies [8,43]. The operation performed by a particular neuron depends upon a number of factors including the location of the synapses relative to the post-synaptic neuron's cell body, the nature of each synapse (excitatory or inhibitory), the level of inherent inhibition or excitation of the post-synaptic neuron (ie. the level of resting activity or inhibitory threshold), and the relative activity of each pre-synaptic neuron. The model to be presented will thus use real numbers to represent data and a selection of the above arithmetic operations to represent the functions of relevant neurons.

The frequency at which action potentials are transmitted is obviously a non-negative number. This restriction upon the values to be represented will be relaxed as it is equally obvious that a fixed positive bias can be superimposed upon any bounded negative variable in

order to guarantee a net positive value. Similarly, although the nature of a particular synapse (excitatory or inhibitory) is fixed, either a bias, or an arrangement of interneurons can be established to permit a synapse to have a net effect which is either positive or negative [2,17].

It is useful to regard the cerebellar cortex system as a form of associative memory [16,25,27]. In such an approach, a set of memories (the firing frequencies of the Purkinje cells) is associated with a set of inputs (the activities of the Mossy and/or Climbing fibers). Due to their remarkable specificity to the output cells, it seems most likely that Climbing fibers are equivalent to "data" lines while Mossy fibers are equivalent to "address" lines. The only other possible arrangement, Mossy fibers providing data and Climbing fibers providing the address, is exceedingly unlikely due to the dispersion of information between Mossy fibers and the system's output, Purkinje cells.

The structure of the Mossy fiber pathway of the cerebellum suggests a number of possible formulations for the value of Purkinje cell activity (system output) as a function of Mossy fiber input. These are: a sum of sums, a sum of products, a product of products, and a product of sums. A moment's reflection will indicate that a sum of products is the most likely form for representing non-linear functions of several variables since both a product of sums and a product of products are always zero whenever any one of the primary terms is zero (either due to coincidence or to a faulty transmission line). Similarly, both a sum of sums and a product of products would not justify the two-stage structure of the system since a single neuron could perform the same computation. Furthermore, Taylor series expansions of functions of several variables show that any function can be adequately expressed as a sum of products,

providing enough terms are used. The choice of formulating system output as a weighted sum of products is not new as it has been suggested by other workers, and it seems highly plausible, that each Purkinje cell effectively computes a weighted sum of the activity of Parallel fibers which synapse with it [2,17,28,32]. That is,

$$\hat{f} = \sum w_i \phi_i \tag{4.1}$$

where $\hat{f}$ =Purkinje cell activity

w_i=i^{th} synaptic weight

ϕ_i=activity of the i^{th} Parallel fiber.

Learning is assumed to take place by adjusting the synaptic weight set {W} as a function of Climbing fiber activity [2,17,21,28].

The function of cerebellar interneurons (Granule, Basket, Stellate, and Golgi cells) in the model must now be considered. This model will take the same approach as a number of previous workers who have suggested that Basket and Stellate cells act to permit the net effect of some w_i to be negative and to permit both increasing and decreasing w_i [2,17,28]. (The nature of a synapse, excitatory or inhibitory, is fixed.) A simple and reasonable model is that Basket and Stellate cells compute an unweighted sum of the activities of the Parallel fibers with which they synapse. Since Basket and Stellate cells are inhibitory, this sum is then subtracted from Purkinje cell activity. That is,

$$\hat{f} = \sum p_i \phi_i - c\sum \phi_i \tag{4.2}$$

where p_i = the synaptic weight between the i^{th} Parallel fiber and the Purkinje cell

c = effective (constant) synaptic connectivity between Parallel fibers and Purkinje cells via the Basket and Stellate cell route.

Re-arranging (4.2) yields

$$\hat{f} = \sum (p_i - c)\Phi_i . \tag{4.3}$$

Letting

$$w_i = p_i - c , \tag{4.4}$$

will result in w_i which may be positive or negative, depending upon the value of p_i. This approach does, however, place a lower limit (-c) on the value of every w_i.

It should be noted that the dendritic tree of each Basket and Stellate cell is less widespread than that of a Purkinje cell. This arrangement reduces the volume of cortical cells by permitting a single Basket or Stellate cell to influence a number of Purkinje cells, despite the fact that each Purkinje cell has synapses with a slightly different subset of Parallel fibers.

As for Granule cells, it has been suggested above that (4.1) must represent a sum of products. This implies that each Granule cell forms a term which is the product of the activity of the Mossy fibers which make contact with its dendrites. To account for Golgi inhibition, Granule cells will be modeled as the two-part cells shown in Figure 4.1. The first part computes the product of the activity of relevant Mossy fibers while the second part subtracts Golgi cell activity (Golgi inhibition). The maximum number of different products is given by the number of all possible combinations of the set of input variables. The set of those products which are actually used in (4.1) (a set to be determined in following sections) will be represented by $\{\Pi\}$. In general, $\{\Pi\}$ contains more elements than the number of Mossy fibers which enter the cerebellar cortex. The set will therefore also be

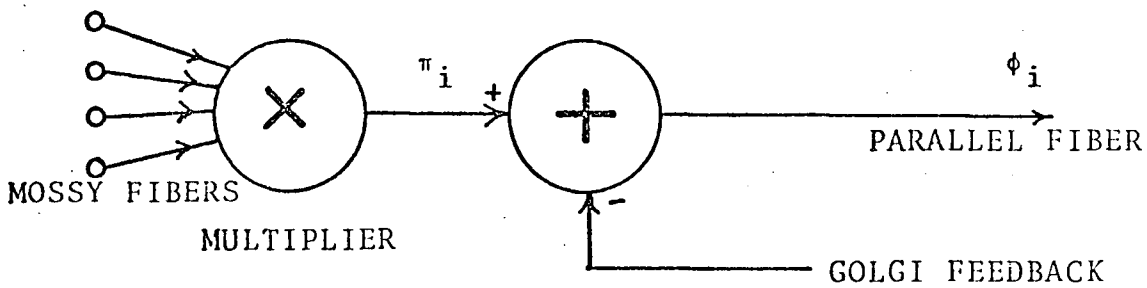


Fig 4.1 Model Granule Cell

referred to as the "Granule expanded input set". Such an expansion is consistent with the large Mossy fiber-Granule cell divergence shown previously in Table 2.1.

Since Golgi cells have two distinct dendritic trees, and since the upper tree is much more dense [12], it would seem that Parallel fiber synapses are more significant to the Golgi system and that Mossy fiber synapses are merely an improvement, possibly to reduce inherent time delays of the system. This assumption does tend to simplify the following analysis but is not an essential ingredient of its basic principles. In any case, the Parallel fiber-Golgi cell system will be

modeled as a negative feedback network of the form

$$\Phi = \mathbb{T} - G\Phi \tag{4.5}$$

where G=Golgi feedback matrix

$\mathbb{T}$=Granule expanded input set.

Once a steady-state condition has been reached, Parallel fiber activity can be expressed as

$$\Phi = (I+G)^{-1}\mathbb{T}. \tag{4.6}$$

This can be re-written as

$$\Phi = \Theta\mathbb{T} \tag{4.7}$$

$$\text{where } \Theta=(I+G)^{-1}. \tag{4.8}$$

Substituting (4.7) into (4.1) and re-writing using vector notation, the system can be expressed as

$$\hat{f} = W^{T}\Phi \tag{4.9}$$

$$= W^{T}\Theta\mathbb{T}. \tag{4.10}$$

4.2 Cerebellar System Considerations

The cerebellum functions as a peripheral controller. That is, using visual, tactile, and other forms of sensory feedback, higher levels of the brain "teach" the cerebellum to control desired motions. This procedure proves advantageous by relieving higher levels of these control tasks once the motions have been learned. To be most efficient,

the cerebellar system should be designed to learn the control parameters as they are being generated by the higher levels. In such a scheme, whenever the estimation error is deemed to be excessive, higher levels simultaneously correct the body position and adjust the weight set so as to reduce the estimation error for a subsequent estimation at that point in the control hyperspace. The existence of synapses between Climbing fibers and Purkinje cells, and between Climbing fibers and Subcortical Nuclear cells adds support to this proposal as it may be further proposed that, during training, Climbing fiber activity overrides that of Purkinje cells in such a way as to first correct the body position and to then reduce the Purkinje cell's estimation error. Also, the cerebellum should be most effective in controlling the most usual motions, thus minimizing the amount of time spent by higher levels in controlling body motions. That is, since (4.1) really estimates a control function, the estimation error should be least in those regions which are used most frequently.

Referring to equations (4.1) and (4.4) and the figures showing cerebellar structure, it is apparent that the only variable elements in this model of the cerebellum are the weights of synapses between Parallel fibers and Purkinje cells. (Basket and Stellate cells may also have modifiable synapses without significantly changing the model.) These variable weights are considered by this model as the set {W}. From a different viewpoint, this means that the only stored values (memory) of the system are the current weights. This fact completely prevents the use of standard matrix inversion techniques in solving (4.1). It is therefore apparent that the system must learn by the use of an error correcting algorithm which effectively performs matrix inversion in an iterative manner.

A well known property of animals, particularly mammals, is their ability to adapt to changes in their size, strength, and environment. Therefore, the mammalian motion controller, the cerebellum, must be able to modify its control parameters at any stage in the animal's development. When considering the cerebellum as a learning machine, this means that the system must be able to learn any number of training points or patterns which may be presented at any time.

A final consideration in any mathematical model of the cerebellum is the number of terms required in the summation given by (4.1). Motion co-ordination would seem to involve non-linear functions of numerous independent variables [42] (sets of current positions, current velocities, desired positions, desired velocities, etc). Each Purkinje cell must therefore generate a function of several variables; the larger the number of variables, the better the co-ordination. Unfortunately, increasing the number of variables rapidly increases the number of terms required. As the model must not require more terms than a Purkinje cell is capable of adding, an upper limit of approximately 200 000 is imposed upon the number of terms in each estimation function as this is the largest estimate of the number of Parallel fibers which synapse with a single Purkinje cell.

V MATHEMATICAL ANALYSIS OF THE CEREBELLAR SYSTEM

5.1 The Cerebellum and Estimation Functions

The nature of the cerebellar system is suggestive of several properties of those mathematical procedures which will be referred to here as "estimation functions". These procedures include the related theories of curve fitting, regression, and interpolation. These theories are discussed in numerous books and papers, including [19], [35,36], and [41] respectively. Estimation procedures typically operate to minimize an expression of the error between a target function and an estimated function. The estimated function is generally expressed as a weighted sum of basis functions

$$\hat{f} = \sum w_i \phi_i(H) \tag{5.1}$$

where H = a vector of one or more independent variables

which is essentially the same equation as (4.1).

There are two significantly different approaches to finding $\hat{f}$ which are reflected in the form of $\{\phi(H)\}$ in (5.1): those in which $\{\phi\}$ is a set of functions which are continuous (and whose derivatives are all continuous) over the hyperspace of definition, and those in which $\{\phi\}$ is a set of functions which are piecewise continuous (or whose derivatives are piecewise continuous). An advantage of the latter approach is that it simplifies the computations, which invariably involve matrix inversion, required to generate $\{W\}$. Piecewise continuous functions,

such as splines, can be arranged so that the matrix is banded, thereby making the inversion process simpler and less prone to numerical (round-off) errors [41].

The only restriction upon $\{\Phi\}$ is that it must be a linearly independent set of functions if the resulting weight set is to be unique. That is, the Granule expanded input set $\{\Pi\}$ forms a basis set which spans a function space. If the elements of $\{\Pi\}$ are not linearly independent, some of them may be deleted, resulting in a reduced set which spans the same function space and is linearly independent. On the other hand, the practical advantages of improved system accuracy and reliability may result from a system in which Parallel fiber activity forms a set of functions which are not linearly independent [15,47]. Although the following derivations will assume linear independence, this condition may be relaxed in some cases.

The choice of the form of $\{\Phi\}$ (ie. polynomials, trigonometric functions, exponentials, etc.) and of the number of elements in the set is a matter of judgement, dependent upon any known properties of the function which is to be estimated.

The above mentioned estimation techniques have usually been developed either for functions of a single variable or for linear functions of several variables. Extension to non-linear functions of several variables is often not as straight-forward as one might expect. (See Chapter 5 of Prenter [41] for a discussion of interpolation functions of several variables.) A particularly severe problem is to ensure the continuity of functions which are interpolated with piecewise continuous interpolation functions. The usual solution to this problem is to form $\{\Phi(H)\}$ as the Cartesian product of interpolation functions of single variables. As an example, for interpolating a function of two

variables, one could use

$$\Phi_k(x,y) = \Phi_i(x) \cdot \Phi_j(y) \tag{5.2}$$

where $k = j + m(i-1)$

m = the number of elements in $\{\Phi(x)\}$.

An important disadvantage of this approach is that the number of elements in $\{\Phi(H)\}$ grows rapidly as the numbers of independent variables and elements in $\{\Phi(x)\}$ increase. That is,

$$L = m^n \tag{5.3}$$

where L = the number of elements in $\{\Phi(H)\}$

n = the number of independent variables.

Fortunately, the extension, to several variables, of continuous interpolating functions is not similarly restricted. The space spanned by a continuous set $\{\Phi(x)\}$ may be extended to $\{\Phi(H)\}$ as a "reduced Cartesian product" in which only those terms with a total order which is less than or equal to some maximum value are retained. For example, the set

$$\{\Phi(x)\} = \{1, \sin(x), \sin(2x)\}$$

may be extended to two dimensions as a full Cartesian product requiring 9 terms,

$$\{\Phi(x,y)\} = \{1, \sin(x), \sin(2x), \sin(y), \sin(2y), \sin(x)\sin(y), \sin(x)\sin(2y), \sin(2x)\sin(y), \sin(2x)\sin(2y)\}$$

or as a reduced Cartesian product requiring only 6 terms.

$$\{\Phi(x,y)\} = \{1, \sin(x), \sin(2x), \sin(y), \sin(2y), \sin(x)\sin(y)\}$$

For a reduced Cartesian product, the number of terms required is given by

$$L = \frac{(m-1+n)!}{(m-1)!n!} \tag{5.4}$$

Where m = the number of elements in the single variable set which is usually system order + 1.

The effectiveness of using an extension based on the reduced Cartesian product is clearly demonstrated by Table 5.1 which compares the number of terms generated by full and reduced Cartesian product expansions for various values of m and n. It is particularly interesting to note the values of n and m which generate approximately 175 000 terms, a number which corresponds to the number of Parallel fibers which synapse with each Purkinje cell.

A somewhat different approach to estimating functions of several variables has been taken by Specht [50]. Based on an analysis of discriminant functions [36], he develops polynomial discriminant functions which can be trained to distinguish between a number of patterns. This theory has been extended to include general regression surfaces [29,51] which can be interpreted, with a suitable change in normalization, as generalized functions of several variables. Unfortunately, the training procedure requires the selection of an arbitrary "smoothness" factor and pre-determining the fixed number of points in the training set. Also, since the technique is based upon approximating a Gaussian function at each point in the hyperspace of definition, a large number of terms must be used throughout the

procedure.

Table 5.1 Number of Terms Generated
by Full and Reduced Cartesian Products

n	m	FULL PRODUCT	REDUCED PRODUCT
3	2	8	4
3	4	64	20
3	5	125	35
6	7	117 649	924
7	5	78 125	330
8	5	390 625	495
15	8	$3.52 \cdot 10^{13}$	170 544
42	5	$2.27 \cdot 10^{29}$	163 185
44	5	$5.68 \cdot 10^{30}$	194 580
100	4	$1.61 \cdot 10^{60}$	176 851

5.2 An Optimized Learning Algorithm

Previous sections have stated a number of constraints under which learning is assumed to take place in a cerebellum-like system. These are:

1. $\hat{f} = W^T \Phi$, (5.5)
2. the system has no memory other than of the current values of synaptic weights $\{W\}$,
3. the system must operate in an error correcting mode,

4. the order and number of training points neither needs to be fixed nor pre-determined, and
5. the basis set, $\{\Phi\}$, can be considered as a set of linearly independent functions of the form

$$\Phi = \Theta\mathbb{T} \tag{5.6}$$

where $\{\mathbb{T}\}$ is a reduced Cartesian product expansion of the input set.

A traditional approach to solving estimation problems is to minimize the weighted mean-square estimation error over the domain of the input variables.

$$J = \int_H s(H)\cdot(W^T\Phi(H)-f(H))^2\cdot dH \tag{5.7}$$

where H is the hyperspace of definition

f is the target function

s(H) is the error cost density function.

Then

$$\frac{\partial J}{\partial W} = 2\int_H s(H)\Phi(H)\cdot(\Phi^T(H)W-f(H))dH = 0 \tag{5.8}$$

yielding

$$W = (\int_H s(H)\Phi(H)\Phi^T(H)dH)^{-1}\cdot\int_H s(H)\Phi(H)f(H)dH \tag{5.9}$$

Since the cerebellar structure shows no apparent mechanism for evaluating (5.9) in a single operation, the problem is to select an iterative scheme which is both compatible with the given structure and converges toward the optimum weight set in a minimal number of iterations. If the system operates to correct errors, can only store

the current values of the weight set, and the order and number of training points is not known, it would seem reasonable that {W} should be changed, thus reducing the estimation error, so as to minimize the effect, of the changed estimation function, on all other points in the space. This scheme will tend to minimize the number of iterations required to obtain a good approximation of the target function.

Let

$$\triangle f(H_1) = f(H_1) - \hat{f}(H_1) \tag{5.10}$$

be the estimation error at some point, H_1. Then, adjusting the weight set so that

$$\triangle W^T \Phi(H_1) = \triangle f(H_1) \tag{5.11}$$

will result in an error correcting function

$$\triangle e(H,H_1) = \triangle W^T \Phi(H) \tag{5.12}$$

which is superimposed upon the function which existed before the change. In order to minimize the effects of this superimposed error correcting function, use

$$J = \int_H s(H) \triangle e^2 dH + v(\triangle W^T \Phi(H_1) - \triangle f(H_1)). \tag{5.13}$$

Standard optimization techniques can be applied to (5.13).

$$\frac{\partial J}{\partial \triangle W} = 2\int s\Phi\Phi^T \triangle W dH + \Phi(H_1)v = 0$$

$$\frac{\partial J}{\partial v} = \Phi^T(H_1)\triangle W - \triangle f(H_1) = 0$$

This can be re-written as a matrix equation.

$$\begin{pmatrix} 2\int s\Phi\Phi^T dH & \Phi(H_1) \\ \Phi^T(H_1) & 0 \end{pmatrix} \begin{pmatrix} \triangle W \\ v \end{pmatrix} = \begin{pmatrix} 0 \\ \triangle f(H_1) \end{pmatrix} \tag{5.14}$$

In order to simplify the notation, let

$$\mathbf{P} = \int_H s\Phi\Phi^T dH. \tag{5.15}$$

Applying row reduction techniques to (5.14) gives an expression for the optimal adjustment, $\triangle W$.

$$\triangle W = \frac{\triangle f(H_1)\mathbf{P}^{-1}\Phi(H_1)}{\Phi^T(H_1)\mathbf{P}^{-1}\Phi(H_1)} \tag{5.16}$$

At this point it is useful to look at some of the properties of **P**.

Lemma 1. **P** is symmetric.

$$\begin{aligned} \mathbf{P}^T = (\int s\Phi\Phi^T dH)^T &= \int s(\Phi\Phi^T)^T dH \\ &= \int s\Phi\Phi^T dH \\ &= \mathbf{P}. \end{aligned}$$

QED.

Lemma 2. **P** is positive definite.

Consider $Y^T\mathbf{P}Y$ where Y is an arbitrary vector.

Then

$$\begin{aligned} Y^T\mathbf{P}Y &= Y^T(\int s\Phi\Phi^T dH)Y \\ &= \int sY^T\Phi\Phi^T Y dH \\ &= \int s(Y^T\Phi(H))^2 dH \end{aligned}$$

Since $\{\Phi\}$ is a set of functions which are linearly independent over H, the only vector for which

$$Y^T\Phi = 0 \quad \text{for all } H$$

is $Y = 0$.

Thus

$$Y^T P Y > 0$$

for all non-zero Y (s(H) is positive by definition) and **P** is positive definite. QED.

Lemma 3. $\mathbf{P}^n$ exists for all real valued n, is unique, and is symmetric.

It is well known that the eigenvectors of a positive definite, real symmetric matrix are real and positive. Thus, let

$$\mathbf{P} = \mathbf{X}\mathbf{L}\mathbf{X}^T$$

where **X** is an augmented matrix whose columns are the eigenvectors of **P**

and **L** is a diagonal matrix whose entries are the eigenvalues of **P**.

Then

$$\mathbf{P}^n = \mathbf{X}\mathbf{L}^n\mathbf{X}^T \tag{5.17}$$

and

$$(\mathbf{P}^n)^T = (\mathbf{X}\mathbf{L}^n\mathbf{X}^T)^T$$

$$= \mathbf{X}\mathbf{L}^n\mathbf{X}^T = \mathbf{P}^n.$$

In particular this proves the existence of $\mathbf{P}^{-1}$. QED.

Lemma 4. The error correcting function, $\triangle e(H,H_1)$ is invariant for non-singular linear combinations of the basis set and is thus a unique function of the region (H), the function space spanned by $\{\Phi\}$, and the estimation error $\triangle f(H_1)$.

The error correcting function for the space spanned by $\{\Phi\}$ is given by

$$\triangle e(H,H_1) = \frac{\triangle f(H_1)\Phi^T(H_1)\mathbf{P}^{-1}(\Phi)\Phi(H)}{\Phi^T(H_1)\mathbf{P}^{-1}(\Phi)\Phi(H_1)} \tag{5.18}$$

Consider

$$\triangle e(H, H_1, \Theta\Phi)$$

where Θ is any non-singular matrix

$$\triangle e(H, H_1, \Theta\Phi) = \frac{\triangle f(H_1) \cdot \Phi^T(H_1) \Theta^T P^{-1}(\Theta\Phi) \cdot \Theta\Phi(H)}{\Phi^T(H_1) \cdot \Theta^T P^{-1}(\Theta\Phi) \Theta\Phi(H_1)}$$

but

$$P(\Theta\Phi) = \int s\Theta\Phi(H)\Phi^T(H)\Theta^T dH$$

$$= \Theta P(\Phi)\Theta^T$$

and

$$P^{-1}(\Theta\Phi) = (\Theta^T)^{-1} P^{-1}(\Phi)\Theta^{-1}.$$

Thus

$$\triangle e(H, H_1, \Theta\Phi) = \frac{\triangle f \Phi^T(H_1) P^{-1}(\Phi)\Phi(H)}{\Phi^T(H_1) P^{-1}(\Phi)\Phi(H_1)}$$

$$= \triangle e(H, H_1, \Phi). \qquad \text{QED.}$$

Lemmas 3 and 4 suggest a useful simplification. Let

$$\Theta = \left(\int_H s \mathbb{T} \mathbb{T}^T dH\right)^{-\frac{1}{2}} \tag{5.19}$$

$$= P^{-\frac{1}{2}}(\mathbb{T})$$

Since $\Phi(H) = \Theta \mathbb{T}(H)$. (5.20)

Combining (5.19) and (5.20), one obtains

$$P(\Phi) = \int_H s\Theta \mathbb{T} \mathbb{T}^T \Theta^T dH$$

$$= \Theta P(\mathbb{T})\Theta^T$$

$$= P^{-\frac{1}{2}} P (P^{-\frac{1}{2}})^T = I.$$

Thus

$$\triangle W = \frac{\triangle f(H_1)\Phi(H_1)}{\Phi^T(H_1)\Phi(H_1)} \tag{5.21}$$

and

$$\triangle e(H,H_1) = \frac{\triangle f(H_1)\Phi^T(H_1)\Phi(H)}{\Phi^T(H_1)\Phi(H_1)} \tag{5.22}$$

In this way, P^{-1} may be deleted from (5.16).

Equation (5.21) results in a scheme which tends to reduce the estimation error for a given function. Further analysis of the set generated by

$$\Phi = P(\mathbb{T})^{-\frac{1}{2}}\mathbb{T} \tag{5.23}$$

indicates an even more efficient approach, employing the properties of orthonormal functions.

Lemma 5. The set of basis functions given by (5.23) is orthonormal over the weighted region.

Consider

$$\begin{aligned} r_{ij} &= (\Phi_i(H),\Phi_j(H)) \\ &= \int_H s\Phi_i(H)\Phi_j(H)dH \end{aligned}$$

Then, R, the matrix whose entries are r_{ij} can be written

$$\begin{aligned} R &= \int_H s\Phi\Phi^T dH \\ &= P. \end{aligned}$$

It has been previously shown that

$$P(\Phi) = I$$

for Φ given by (5.23).

Thus

$$r_{ij} = \begin{cases} 0 \text{ if } i \neq j \\ 1 \text{ if } i = j \end{cases} \tag{5.24}$$

and $\{\Phi(H)\}$ is an orthonormal set over the weighted region. QED.

As a further improvement of (5.21). Consider

$$\triangle W = \frac{\triangle f(H_1)\Phi(H_1)}{k} \tag{5.25}$$

$$\text{where } \text{MAX}[\frac{\Phi^T\Phi}{k}] << 1.$$

Let the weight set be adjusted at a total of k points with a point density u(H).

$$u(H) = \frac{s(H)}{\int s(H)\,dH} \tag{5.26}$$

The sequence of $\triangle W$'s at successive learning points, assuming all weights are initially zero, will be

$$\triangle W_1 = \frac{f(H_1)\Phi(H_1)}{k},$$

$$\triangle W_2 = \frac{f(H_2)\Phi(H_2)}{k} - \frac{f(H_1)\Phi^T(H_2)\Phi(H_1)\Phi(H_2)}{k^2},$$

$$\triangle W_3 = \frac{f(H_3)\Phi(H_3)}{k} - \frac{f(H_2)\Phi^T(H_3)\Phi(H_2)\Phi(H_3)}{k^2}$$

$$+ \frac{f(H_1)\Phi^T(H_3)\Phi(H_2)\Phi^T(H_2)\Phi(H_1)\Phi(H_3)}{k^3},$$

etc.

Yielding

$$W = \sum \Delta W_i$$

$$= \frac{1}{k} \sum_{i=1}^{k} f(H_i)\Phi(H_i)$$

$$- \frac{1}{k^2} \sum_{i=2}^{k} f(H_{i-1})\Phi^T(H_i)\Phi(H_{i-1})\Phi(H_i) + \ldots \qquad (5.27)$$

Since the points have a density given by u(H), the first term of (5.27) is an approximation of the weighted integral

$$\int_H u(H) f(H)\Phi(H) dH = \frac{\int s(H) f(H)\Phi(H) dH}{\int s(H) dH} \qquad (5.28)$$

$$\approx \frac{1}{k} \sum f(H_i)\Phi(H_i) . \qquad (5.29)$$

Comparing (5.29) with (5.9) and applying Lemma 5, it can be seen that, after k points, the weight set approximates the optimal set. Continuing the procedure with additional learning points will improve the estimate as long as

$$\Delta f_i \approx \Delta f_{i-1} .$$

Similarly, a sequence of learning steps at k points will adjust the weight set so that any resultant estimation error is approximately orthogonal to all elements of $\{\Phi\}$. That is, the error is minimized and $\{W\}$ approximates the optimum set. Unless the estimated function can be made to be identical to the target function and the training data is noiseless, the values of the elements of $\{W\}$ will change at each point as learning is applied. That is, $\{W\}$ will not in general converge in the usual sense, but will tend to fluctuate about the optimum weight set. The sequence of weight sets will, however, have a point of

accumulation at the location of the optimum set.

It is important to note at this point that replacing the denominator of equation (5.21) with a constant, k, results in error corrections which are un-normalized. That is, despite applying (5.21) to correct an estimation error, an error may still exist. This variation of the learning algorithm, which will be discussed further in the following chapters, requires that an additional element be added to the system equations. Specifically, the system equations become

$$\begin{aligned} \hat{f} &= W^T\Phi \\ \overline{f} &= \hat{f} + \triangle f \cdot g(t) \\ \triangle f &= f - \overline{f} \\ \triangle W &= \frac{\triangle f \Phi}{k} \end{aligned} \tag{5.30}$$

where $\hat{f}$ = the estimated function (Purkinje cell output)

$\triangle f$=system error term

$\overline{f}$ = system corrected output (Subcortical Nuclear cell output).

$g(t) = 1$ and $g(t+\triangle t) = 0$

That is, the system requires an element which evaluates $(\hat{f} + \triangle f)$ prior to any weight adjustments being made, holds this value, then permits it to decay to the new value of $\hat{f}$.

Various procedures are available to permit the learning procedure to be halted. The most direct approach is to require learning only if the obtained response (either the estimated function or some physical response of the system) deviates sufficiently from the desired response. This in effect replaces the single valued target function with a band of acceptable estimations.

That is

$$\Delta W = \begin{cases} 0 & \text{if } |\Delta f| \leq \epsilon \\ \dfrac{\Delta f \Phi}{k} & \text{if } |\Delta f| > \epsilon \end{cases} \tag{5.31}$$

The learning algorithm is also effective even if the distribution of points is not known a priori. If possible, the expected point density can be estimated, or if not, it may be replaced by a constant (over a closed reqion) when deriving the basis set. Each time the weights are adjusted, an error correcting function is superimposed upon the previous function. The learning algorithm ensures that interference effects at each point are minimized at all other points in the learning hyperspace. This approach will result in a sequence of weights which tends to converge toward the optimal set. Other researchers have presented arguments showing that similar devices such as Adalines [10] and Perceptrons [32] exhibit strong convergence tendencies. Although not optimized, these devices are similar to this one in that they also approximate functions as weighted sums.

Regardless of the known and unknown parameters of the system, it is necessary to select k in (5.25). There are two opposing considerations: stability and learning rate. Larger values of k reduce the magnitude of perturbations, due to points with low probability or high noise, of the estimation function, while smaller values of k reduce the number of points required before the estimation function begins to approximate the target function. Unless very rapid learning is required, k should be determined to ensure that the fluctuations in $\hat{f}$ are acceptably small.

That is, with re-arrangements to (5.25)

$$\text{MAX}[\triangle e(H,H_1)] = \text{MAX}\left[\frac{\triangle f(H_1)\Phi^T(H_1)\Phi(H)}{k}\right] . \tag{5.32}$$

Thus, k must be sufficiently large so as to ensure that the effect, of an adjustment at a single point, is less than ϵ over all points in the space.

$$k \geq \text{MAX}\left[\frac{\triangle f(H_1)\Phi^T(H_1)\Phi(H_1)}{\epsilon}\right] \tag{5.33}$$

To guarantee that continuous iterations at any single point will be stable and converge to generate $f(H_1)$ exactly (pointwise convergence),

$$0 < \frac{\Phi^T(H_1)\Phi(H_1)}{k} < 2. \tag{5.34}$$

The next chapter will present examples which demonstrate the effects, upon the learning system, of varying the values of k and ϵ.

5.3 System Properties

The preceding section has developed a system which can learn to approximate, with minimum mean-square estimation error, arbitrary functions as linear combinations of an orthonormal set of basis functions. The system uses an iterative error correcting strategy in

which the weights are adjusted at each stage according to the expression (repeated here for convenience)

$$\triangle W = \frac{\triangle f(H_1)\Phi(H_1)}{k} .$$

The factor k depends upon:

1. the basis set $\{\Phi\}$ chosen,
2. the required precision of the final estimation function,

and is a constant which may be computed in advance and/or altered at any time.

Concerning the system's operating characteristics, a useful property is that the resultant estimation function is capable of estimating a target function with an error which is less than the error correcting mechanism can detect. In other words, despite the training mechanism being rather inexact in its ability to detect or correct an error, the final estimation function will, in general, be a much more precise approximation of the target function. To support this statement, consider a learning system, the weights of which are corrected whenever the estimation error exceeds an acceptable (or detectable) tolerance, ϵ, as given by (5.31). If the function space spanned by the basis functions is such that the tolerance of ϵ can in fact be met for the whole hyperspace, then the resulting estimation will lie within a region bounded by

$$f(H) - \epsilon \leq \hat{f}(H) \leq f(H) + \epsilon \tag{5.35}$$

as shown in Figure 5.1. For such a bounded, continuous, estimation function

$$|f(H_1) - \hat{f}(H_1)| << \epsilon \tag{5.36}$$

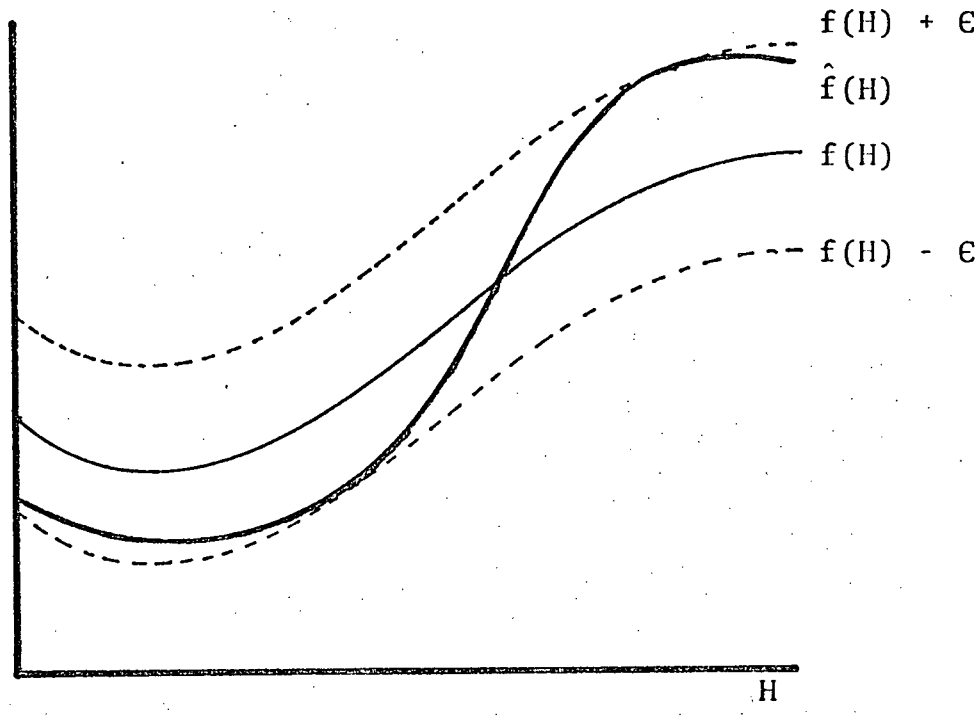


Fig 5.1 A Bounded Estimation Function

over much of the region, H. Hence, $\hat{f}$ produces a better estimate, in terms of the desired result, than could be obtained using the error detecting/correcting mechanism alone. A somewhat similar property, called "learning with a critic", has been described for an Automatically Adapted Threshold Logic Element (Adaline) [54]. In that experiment, the Adaline, is taught the optimal strategy for playing the game of blackjack strictly on the basis of whether it wins or loses a game. That is, despite the fact that the error detecting mechanism determines that the estimation function is in error if, and only if, a game is lost, the Adaline learns to generate the function which optimizes the

strategy for winning games.

VI SYSTEM PERFORMANCE

6.1 General Considerations

The previous chapter has derived an algorithm which minimizes the interference of learning arbitrary functions, when using an iterative, point-by-point algorithm, in a cerebellum-like system. This chapter will demonstrate the effectiveness of the algorithm.

The derivation of the learning algorithm is independent of the function-space which is defined by the basis set. Thus, the algorithm guarantees optimal performance (for that space) for every linearly independent set of functions $\{\Phi\}$, which satisfy (5.19). The choice of which basis set to use is a matter of selecting, subject to implementation constraints, that function-space and basis set which are most likely to match the function or functions to be estimated. Multinomials are a reasonable choice for a large number of applications, and are the functions which will be used in the examples presented in this chapter. That is, the terms to be used are those generated by

$$(1 + \sum_{i=1}^{n} x_i)^m. \tag{6.1}$$

Application of Lemma 4 permits the binomial coefficients to be dropped from each term, yielding

$$\mathbb{T}^T = (1, x_1, x_2, \ldots, x_n, x_1^2, \ldots, x_n^m) \tag{6.2}$$

$$\text{and} \quad \Phi = \left(\int_H s\, \mathbb{T}\mathbb{T}^T dH\right)^{-\frac{1}{2}} \mathbb{T} \tag{6.3}$$

Due to the absence of other information, s(H) will be set to a constant in the following examples.

Some typical, normalized, error correcting functions, $\triangle e(H,H_1,\Phi)$, are shown in Figure 6.1. It can be seen that these functions bear a loose resemblance to normal probability density functions with mean of H_1. The figures also show that limitations imposed by using a system based on low-order polynomials result in error correcting functions whose maxima are often "skewed" or shifted from H_1.

In regard to {T}, it should be noted that any number of the terms listed in (6.2) may be deleted, without jeopardizing system performance, if the coefficients of those terms, in the target functions, are known to be zero.

Another important parameter which was discussed in the previous chapter is k which must be selected in relation to

$$b(H_1,H_2) = MAX[\Phi^T(H_1)\Phi(H_2)] \tag{6.4}$$

as given by (5.31), (5.32), (5.33). For any basis set {Φ}, b will be maximum at some point, H_b, where

$$|\Phi(H_b)| \geq |\Phi(H_j)| \tag{6.5}$$

$$\text{for all } H_j \neq H_b,$$

since

$$\Phi^T(H_i)\Phi(H_j) = |\Phi(H_i)||\Phi(H_j)|\cos\theta \tag{6.6}$$

where θ is the "angle" between the vectors $\Phi(H_i)$ and $\Phi(H_j)$.

Thus

$$b = \Phi^T(H_b)\Phi(H_b) = |\Phi(H_b)|^2 \cos 0$$

$$= |\Phi(H_b)|^2. \tag{6.7}$$

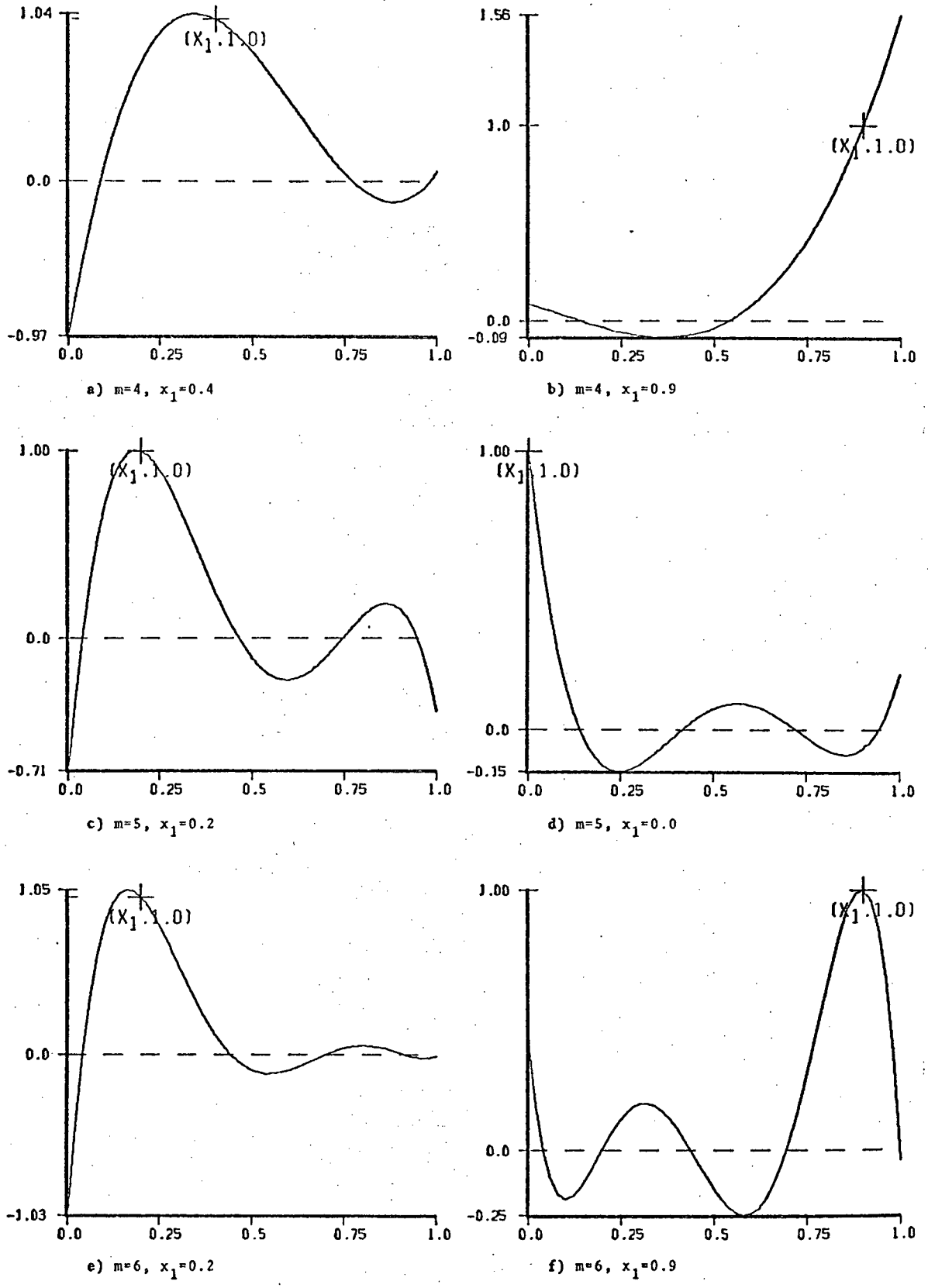


Fig 6.1 Error Correcting Functions for a Single Variable

Table 6.1 Values of $b=MAX[\Phi^T\Phi]$ for Various Values of m and n

n	m	b	Average*	n	m	b	Average*
1	2	4	2.0	2	3	26	6.9
1	3	9	3.1	2	4	70	12.3
1	4	16	4.2	2	5	155	20.0
1	5	25	5.3	3	4	190	32.3
1	6	36	6.4	3	5	553	67.0

*Average: approximate average value of $\Phi^T\Phi$

The location of H_b and the value of b are properties of $\{\Phi\}$. Table 6.1 shows that, for multinomial-based functions, b grows rapidly as m and n increase. This in turn means that k, and hence the number of points required to obtain acceptable estimations, must be similarly increased.

6.2 Examples of Learning a Function of a Single Variable

In order to demonstrate the operation of the training algorithm, and its ability to estimate functions, the function

$$f = \sin(2\pi x) \qquad (6.8)$$

$$\text{over} \quad (0 \leq x \leq 1)$$

was "taught" to a number of computer-based models of the system. The target function is shown in Figure 6.2. The optimal estimation functions, in a least-square-error sense, to the target function are shown in Figure 6.3 for polynomials of various orders. These functions

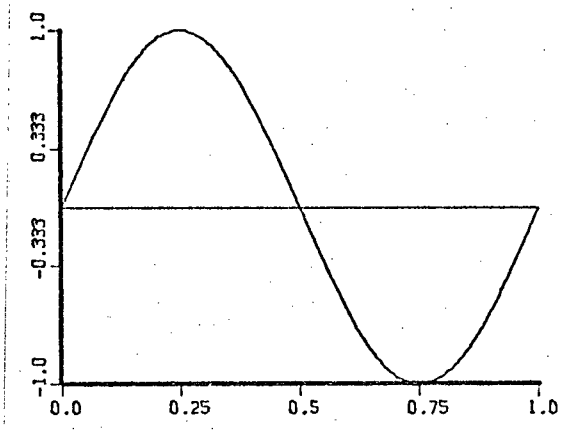
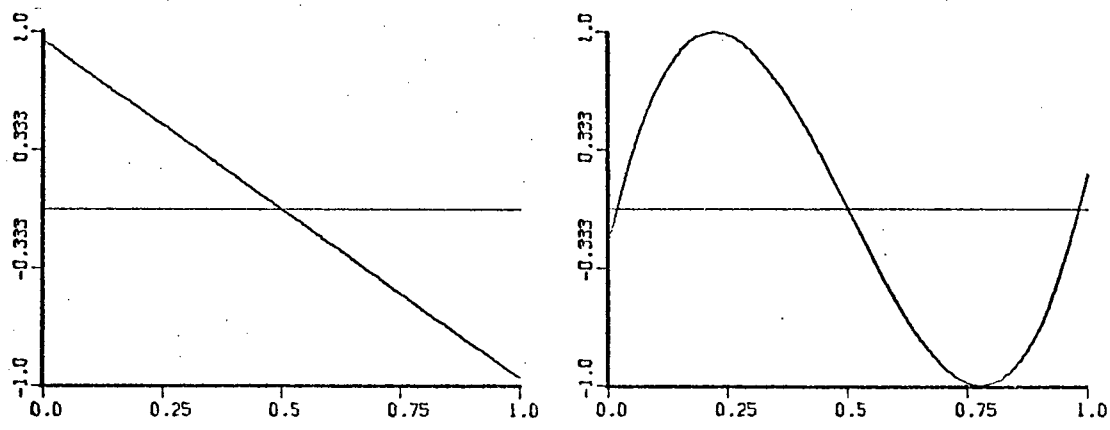


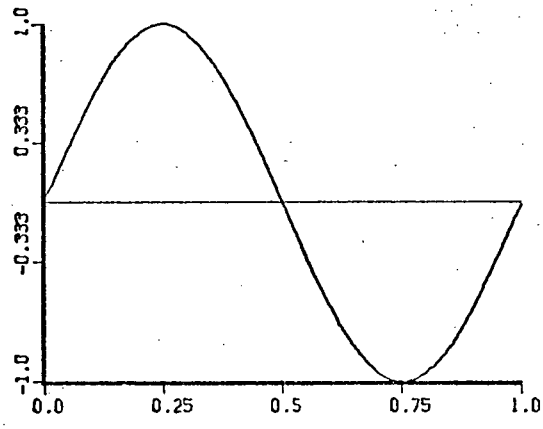
Fig 6.2 Target Function Sin (2πx) over ($0 \leq x \leq 1$)

were calculated using the optimizing formula derived as equation (5.9). Chapter 5 suggests that, as learning progresses, the estimation function grows, then settles, toward the optimal estimation of the target function. This sequence of functions is shown in Figure 6.4. For large values of k, ϵ=0, and many training points, Figure 6.5 shows that learned estimation functions can closely approximate the optimal functions shown previously.

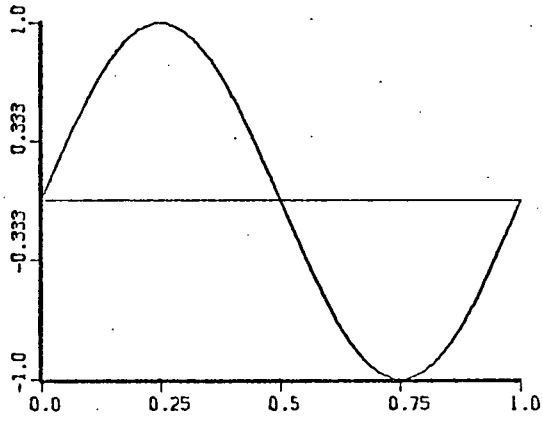


a) m=2,3 RMS ERROR=0.46 EMAX=0.95

b) m=4,5 RMS ERROR=0.073 EMAX=0.20



c) m=6,7 RMS ERROR=0.0050 EMAX=0.016



d) m=8,9 RMS ERROR=0.00019 EMAX=0.00066

Fig 6.3 Optimal Polynomial Approximations of Sin (2πx)

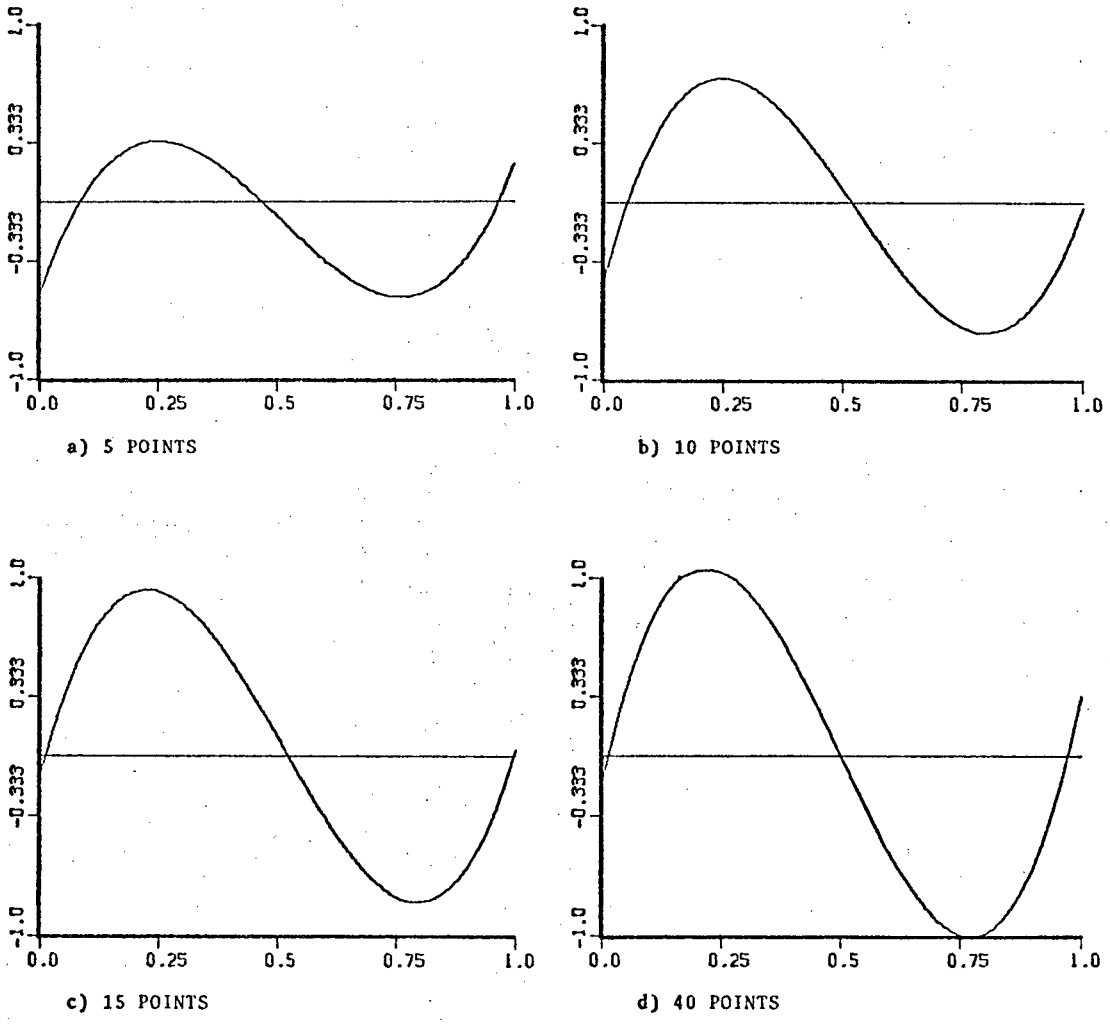


Fig 6.4 Sequences of Estimations of Sin (2Πx)

For m=4, k=8

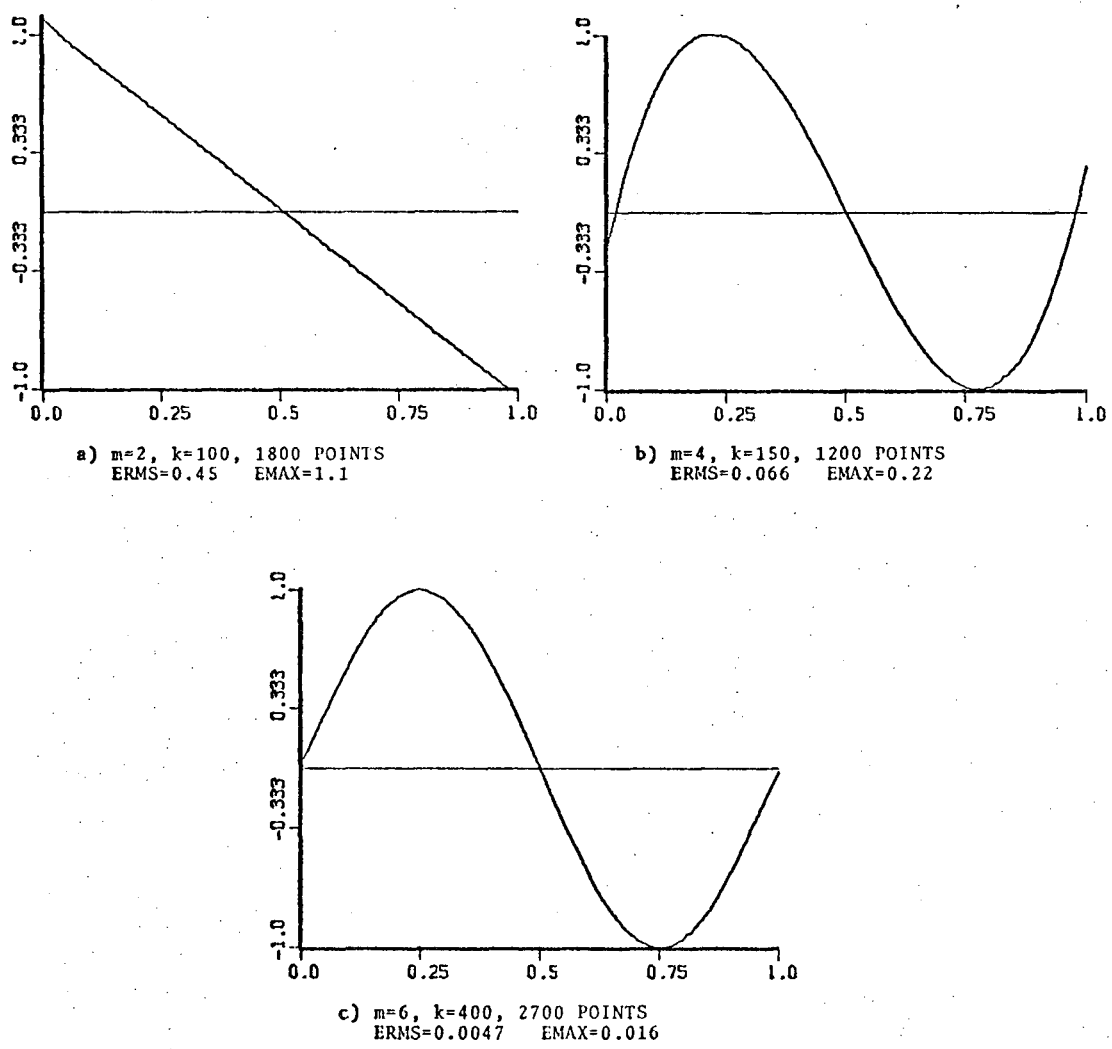


Fig 6.5 Near-optimal Estimations of Sin (2πx)

Once the order of the estimation system has been selected, values of k and ϵ must be determined. The effects, on the learning system, of varying k and ϵ are shown by various calculated measures:

1. EMAX, the maximum estimation error, $\triangle f(H_i)$, which was observed in the course of processing the preceeding J points,
2. CHANGES, the total number of times the weight set has required adjustment,
3. ERMS, an estimate of the RMS estimation error over the preceeding J points

$$ERMS = \left(\frac{1}{J}\sum_{i=1}^{J}\triangle f^2(H_i)\right)^{\frac{1}{2}}, \tag{6.9}$$

4. SF, a measure of the perturbations of the estimated function, which is calculated as the change over the preceeding J points, in the values of the elements of the weight set (SF = stability factor)

$$SF = \left(\frac{1}{L}(W_i - W_{i-J})^T(W_i - W_{i-J})\right)^{\frac{1}{2}} \tag{6.10}$$

where W_i is the current weight set

W_{i-J} is the weight set prior to any adjustments resulting from applying the learning algorithm at the previous J points,

5. POINTS, the total number of points which have been observed.

Table 6.2 The Learning Algorithm as Effected by k and ϵ

Estimating $\sin(2\pi x)$ for $0 \leq x \leq 1$

m=4, J=100

a) Effect of k ϵ=0.0

u(H)=s(H)=uniform

POINTS	k	EMAX	ERMS	SF
100	10	0.76	0.19	0.36
200		0.29	0.077	0.059
400		0.25	0.087	0.036
600		0.34	0.093	0.035
100	50	0.86	0.36	0.30
200		0.19	0.080	0.042
400		0.21	0.070	0.0086
600		0.27	0.077	0.012
100	100	0.91	0.46	0.21
200		0.37	0.19	0.084
400		0.17	0.074	0.0012
600		0.23	0.074	0.0063
800		0.22	0.069	0.0045

b) Effect of ϵ k=100

u(H)=s(H)=uniform

POINTS	ϵ	CHANGES	EMAX	ERMS	SF
200	0.10	153	0.37	0.19	0.080
400		224	0.17	0.079	0.015
600		253	0.15	0.077	0.0090
800		262	0.11	0.074	0.0012
200	0.15	133	0.37	0.20	0.074
400		178	0.20	0.095	0.0093
600		184	0.16	0.098	0.0043
800		184	0.15	0.088	0.0

These measures are used since they show important properties of the learning system, are relatively easy to calculate in conjunction with applying the learning algorithm, and automatically take into account the effect of non-uniformly distributed training points. It will be noted, though, that due to the training points being randomly generated, the measures tend to oscillate rather than decrease monotonically. Table 6.2 demonstrates the following trade-offs in relation to k and ϵ:

1. increasing k increases the number of training points required before the system begins to settle toward the optimal estimate,
2. once the estimation approaches the optimum, larger values of k reduce the magnitude of perturbations as indicated by smaller values of SF,
3. smaller values of ϵ produce estimation functions which tend to have less RMS error, and
4. larger values of ϵ tend to reduce the maximum estimation error at the cost of larger RMS errors.

Chapter 5 also predicts that, although its performance will be sub-optimal, the learning algorithm will continue to function in cases where the point density is not equivalent to the error cost density which was used to generate the basis set, as required by (5.26). Although Table 6.3 does support this prediction, it also shows a disadvantage of this approach since many more training points (1500 versus 1000 training points) are required before the estimated function approaches the optimum.

Table 6.3 Learning Sequences for Cases Where $u(H) \neq s(H)$

Estimating $\sin(2\pi x)$ for $0 \leq x \leq 1$

$m=4$, $\epsilon=0.0$, $k=100$, $J=100$, $u(H)$=uniform

a) $s(H)$ is gaussian (mean=0.5, variance=1.0)

POINTS	EMAX	ERMS	SF
200	0.66	0.39	0.12
400	0.34	0.19	0.056
600	0.18	0.11	0.026
800	0.19	0.074	0.0095
1000	0.17	0.069	0.0058

b) $s(H)$ is gaussian (mean=0.5, variance=0.5)

POINTS	EMAX	ERMS	SF
200	0.81	0.51	0.12
400	0.58	0.34	0.077
600	0.40	0.25	0.053
800	0.29	0.16	0.033
1300	0.15	0.081	0.012
1500	0.13	0.067	0.0062

6.3 Soft Failure

An interesting, and potentially useful, factor to be considered when assessing the value of this learning algorithm is its property of soft-failure. That is, should one or more elements of $\{\phi\}$ or $\{W\}$ become inoperative, the system will still be capable of representing arbitrary

functions to any required accuracy, possibly after additional training, as long as the region of representation is permitted to be sufficiently small. To prove this conjecture, it is only necessary to show that at least one functional element of $\{\Phi\}$ is non-zero at the point where estimation is required. Multinomial-based systems can be arranged so that few (likely no more than n) elements of $\{\Phi\}$ have values of zero at the same point and hence systems based on these sets will exhibit soft failure. In terms of a physical device which implements the algorithm, this means that the device may be useful, possibly over a reduced range, despite the failure of some of its components. When considering systems which employ splines or harmonic series, it is seen that at a number of points in the region of definition, all but a limited number of the elements of the basis function set are zero. Should faulty components cause all these non-zero elements to become inoperative, the system's output will be a fixed, erroneous, value. Thus the present system of multinomial-based functions has a significant, practical advantage.

6.4 Learning Functions of Several Variables

In order to demonstrate the ability of the learning algorithm to learn and generate several functions of several variables, it was applied to a more complicated model. The model used is based upon the geometry of a human arm as shown in Figure 6.6 [26].

The system, whose model was programmed on a digital computer, is best described by the block diagram of Figure 6.7. In this figure, it can be seen that a desired wrist position, H_i, is used as the input to

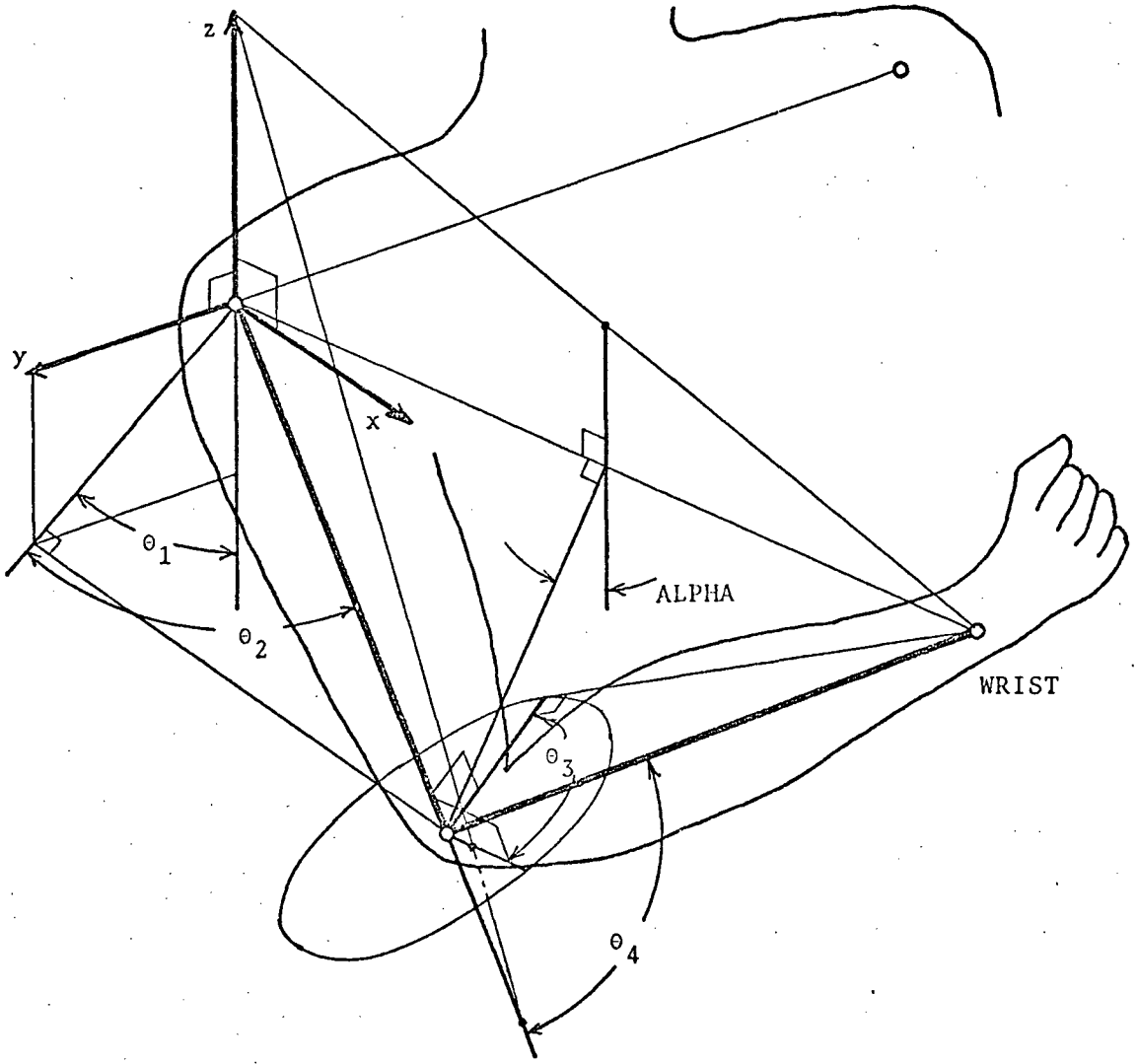


Fig 6.6 The Geometry of the Model Arm
(after [26])

the estimation (learning) system. Values of $\{\Phi(H_l)\}$ are then generated and used to compute each joint angle according to

$$\hat{f}_1(H_i) = \Phi^T(H_i)W_1$$
$$\hat{f}_2(H_i) = \Phi^T(H_i)W_2$$
$$\hat{f}_3(H_i) = \Phi^T(H_i)W_3$$
$$\hat{f}_4(H_i) = \Phi^T(H_i)W_4 \qquad (6.11)$$

where W_i is the weight set associated with $\hat{f}_i$.

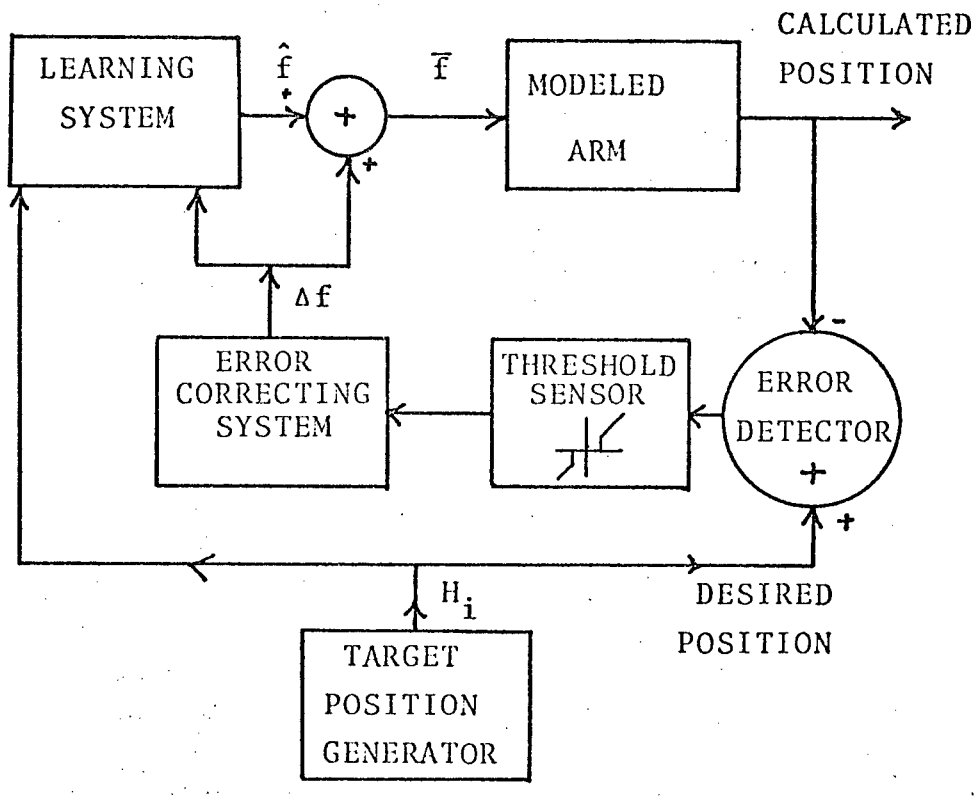


Fig 6.7 Block Diagram of the Arm Model Learning System

These joint angles are then used to compute the resultant wrist position. This position is compared with the desired one, and if the error is greater than ϵ, the angular corrections are computed (modeling a sensory feedback loop), the system output, $\bar{f}$, is corrected with $\triangle f$ to obtain the desired wrist position, and the learning algorithm is applied. It is particularly interesting to note the similarities between Figure 6.7 and the cerebellar system shown in Figure 2.1.

The model was presented with randomly generated points which have a

uniform distribution inside a box which is located in the region

$$5 \leq x \leq 13$$

$$-5 \leq y \leq 12$$

$$0 \leq z \leq -9$$

where all dimensions are in inches.

Table 6.4 demonstrates the rapid learning rate of this model and shows that the model learns to position the wrist with good accuracy after learning at approximately 3000 points. The effect of varying ϵ is also demonstrated. That is, larger values of ϵ tend to reduce the maximum error while increasing the RMS error.

Table 6.4 Learning Sequences for the Arm Model

m=5 k=500 J=500 Arm flap (alpha)=40 degrees

u(H) = s(H) = uniform

ERMS and EMAX are in inches

a) ϵ=0.0

POINTS	CHANGES	ERMS	EMAX	SF
1000	1000	6.70	18.3	0.045
2000	2000	0.99	2.52	0.0066
3000	3000	0.19	0.73	0.0010
4000	4000	0.13	0.48	0.0004
5000	5000	0.14	0.84	0.0005

b) ϵ=0.4

POINTS	CHANGES	ERMS	EMAX	SF
1000	1000	6.70	18.3	0.045
2000	1934	0.99	2.52	0.0065
3000	2150	0.33	0.68	0.0003
4000	2171	0.32	0.46	0.0001
5000	2188	0.32	0.64	0.0003

VII DISCUSSION AND CONCLUSION

7.1 Physiological Implications

The algorithm which has been developed in this thesis is based upon the known anatomy and physiology of the mammalian cerebellum. It is therefore reasonable to predict that cerebellar operations may be very similar to those of the learning algorithm herein described. That is, the mathematics of modifiable synapses, and the functions of cerebellar cells may well be those given by equations (4.1), (4.3), (5.30), and (5.31) which specify the learning system derived by this thesis.

An important property of this system is that all inputs are treated identically. That is, there is no need to differentiate between those Mossy fiber inputs which are related to sensory or peripheral information and those which are related to "context" or commands. This permits both sensory and command parameters to be treated as continuous variables so that commands inherently contain rate factors. Thus "walk", "run", and "sprint" may be the same command, "move", at various intensities. The lack of specificity also means that there is no need for an exact mapping of Mossy fibers to specific points (or to a specific Granule cell) in the cerebellar cortex. Rather, a general target area is all that is required, thus reducing the amount of information which must be stored by cerebellum-related genes.

Both of the above properties represent significant improvements over existing cerebellar models. In particular, this model resolves the deficiencies of Albus' theory [2] by presenting a feasible mapping of

Mossy fiber activity to Purkinje cell activity and by treating all inputs, both commands and peripheral data, identically as continuous variables.

The model has predicted that corrections to Purkinje cell activity are not normalized. That is, Climbing fiber activity ($\triangle f$) causes the synaptic weights of the target Purkinje cell to change, resulting in a change in the frequency of that Purkinje cell's action potentials ($\triangle \hat{f}$) which is not exactly equivalent to $\triangle f$. To permit the proposed cerebellar system to function in this manner it is necessary that Subcortical Nuclear cells perform a "sample-and-hold" function, thus generating an output from the cerebellum ($\bar{f}$) (see equation 5.30) which is corrected exactly. The weight adjustments, which correspond to learning, are not normalized for several reasons:

1. to permit weight adjustments to be strictly local computations; a function of the pre and post-synaptic activities at the synapse whose weight is being adjusted,
2. to permit the optimal weight set to be computed as the approximation of an integral as given by equation (5.29),
3. to speed convergence in the manner of convergence gain factors in numerical optimization and root finding techniques, and
4. to aid system stabilty by reducing the effects of correcting errors at infrequent points.

If further physiological experiments should disprove this un-normalized operation, the algorithm may be modified so that iterations at any single learning point act to compute weight adjustments which result in changing $\hat{f}$ by an amount equal to the exact

error at that point. That is, continuous re-applications of the weight adjustment algorithm will reduce the estimation error to zero, thus computing the weight adjustment as

$$\triangle W = \frac{\triangle f \Phi(H_1)}{k} \cdot \frac{k}{\Phi^T(H_1)\Phi(H_1)} \tag{7.1}$$

which is really the equation given previously as (5.21).

Future experiments may also show that Parallel fiber activity is not constant as implied in Chapter 5. That is, it may be as suggested by Eccles [14], that Parallel fiber activity is normally insignificant, rising during cerebellar computations, then returning to a near-zero level. In this case, Purkinje cell activity would also be transient, rising briefly, then returning to its normal, spontaneous, rate. To account for this modification of the nature of cerebellar operation, the theory again requires only slight variation. Namely phasic (or a combination of phasic and tonic) activity, rather than strictly tonic activity, would be used to form the orthonormal set to which the learning algorithm is applied.

7.2 The Learning Algorithm and Machine Intelligence

The learning alogrithm described in this thesis may be applied directly to learning machines. The cerebellum, after which the system is modeled, is an extremely effective motion controller. This suggests that the current system may prove effective in a number of applications as an adaptive controller. Potential applications include situations where complicated, possibly unknown, non-linear control equations of

several variables are found such as in power system control [52], in industrial process control [30,49], and in robotics [5,6].

Another application of the system is in pattern recognition. The system has been shown to be capable of simultaneously learning to generate a number of functions of several variables. If these variables are parameters derived from a family of patterns, and if the output functions are the probabilities that a given input pattern corresponds to each of a number of classes, then the device is really a pattern recognizer.

In these and other applications, a hierarchical network of learning machines [4,31] such as that shown in Figure 7.1 may prove effective. In this arrangement, successively higher levels control correspondingly higher operations. Each device performs its own control functions, directs lower levels, and processes information to be used by higher levels. The Figure shows a conceptual arrangement. In practice, the large numbers of inputs and outputs which can be handled by a cerebellum-like learning machine permit a single physical device to act as several levels of the hierarchy simply by using computed (output) parameters as input variables. This feedback provides such a system with the capability to compute and/or control complicated functions. It also poses interesting problems regarding programming or teaching strategies for the system.

The learning algorithm may also be considered as a refinement of the Polynomial Discriminant Functions [29,50,51] discussed in Chapter 5. The advantage of the new approach is to remove the requirement of using a fixed number of training points. With the new algorithm, there is no upper limit on the number of points which may be used for training purposes.

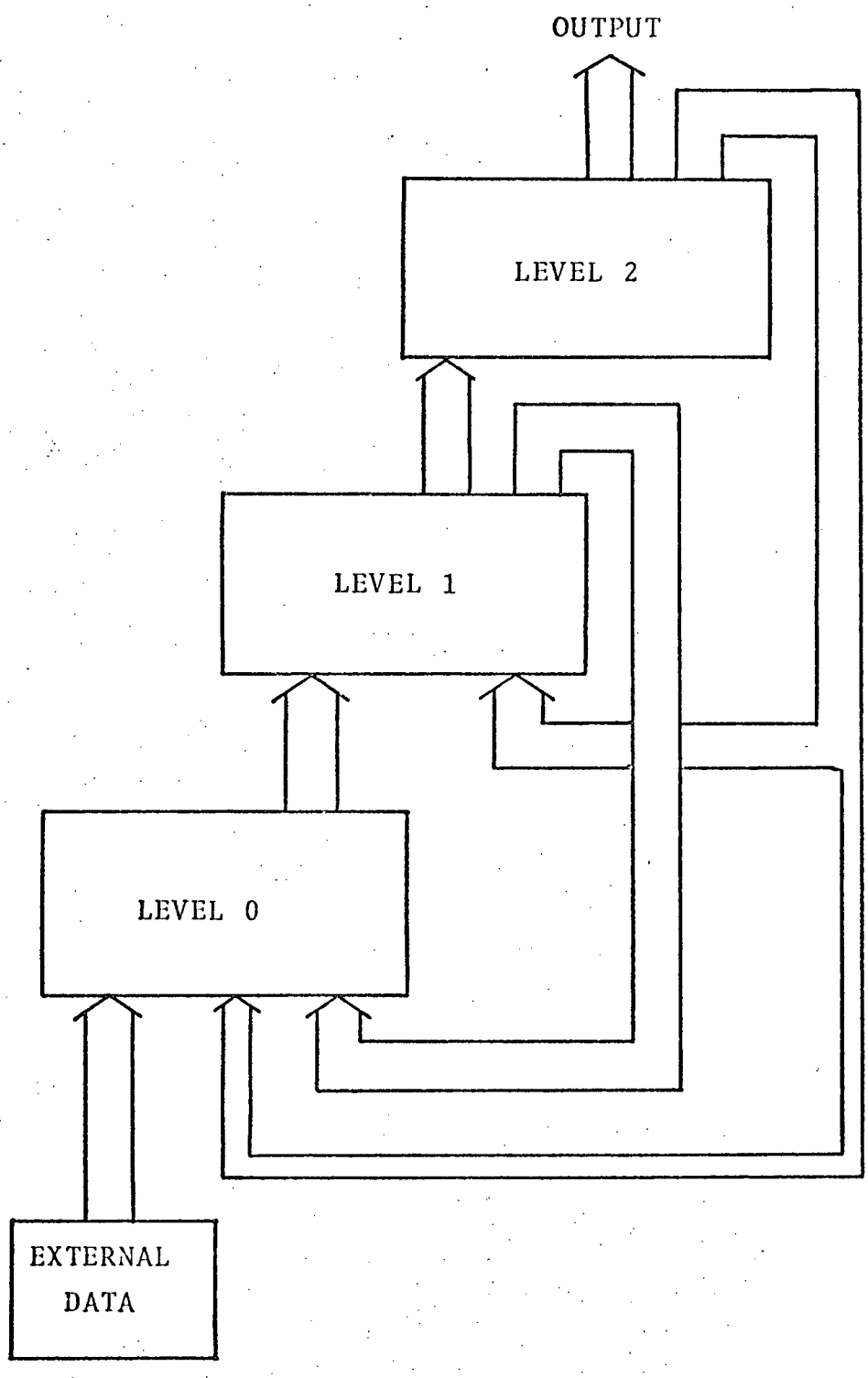


Fig 7.1 A Hierarchical System of Learning Machines

Finally, the parallel processing properties of this learning system are important. It should be remembered that the same basis functions are used to compute all the output functions so that any number of output functions and error corrections may be calculated simultaneously. All output cells are isolated from each other, thus permitting selective adjustment of estimating functions; only those functions whose error is excessively large requiring adjustment at any time.

7.3 Contributions Of This Thesis

The major contribution of this thesis is the development of a system which has the capacity to learn to approximate arbitrary, high order functions of several variables. By representing estimated functions as weighted sums of continuous basis functions, and by employing an iterative solution rather than one which uses matrix inversion, the system requires relatively few variable elements (memory).

The thesis has also shown that, for orthonormal basis sets, learning interference is minimized if estimation errors are reduced by adjusting the weight set {W} according to the expression

$$\triangle W = \frac{\triangle f(H_1)\Phi(H_1)}{k} . \qquad (7.2)$$

This procedure thus reduces the number of iterations which are required before arbitrary functions are approximated to a required accuracy. The procedure has also been shown to result in a learned weight set which produces an estimated function closely approximating the least-square-error estimation of the target function.

Since the system is based on the structure of the mammalian cerebellum which is a very efficient adaptive controller, it shows promise in a number of applications and as the basis of a new class of intelligent machines.

A generalized method of constructing the orthonormal sets of functions required by the system has also been presented.

The second significant contribution of this thesis is to present an improved cerebellar model which, being consistent with current anatomical and physiological data, is more plausible than previous models. Unlike other cerebellar models, this one simulates neural activities as continuous variables, rather than as binary variables, throughout all its operations. This is important since, despite the "all or nothing" character of action potentials, neural information is generally thought to be transmitted as frequency coded values. Also, the model proposes an alogorithm which adjusts synaptic weights strictly according to the activities of the pre and post-synaptic neurons at that synapse. Referring to (7.2), the pre-synaptic activity is $\Phi(H_1)$ while the post-synaptic activity is $\triangle f(H_1)$. This property of localized learning is of critical importance to a plausible cerebellar model as the long, narrow, and widespread structure of Purkinje cell dendrites makes an algorithm which requires computations involving non-local activity exceedingly unlikely.

7.4 Areas of Further Research

This thesis has succeeded in developing an optimized learning algorithm which models the cerebellum. However, as in most research,

while resolving some issues, it leaves many interesting questions to be answered.

In terms of the cerebellum, a number of physiological investigations are indicated:

1. Determine the mathematical form of the mapping from Mossy fiber activity to Granule cell activity. In particular, determine whether the frequency of action potentials in Parallel fibers can be interpreted as a basis set of the form suggested by this thesis:

$$S(H) = A+\Phi(H) \tag{7.3}$$

where $\{\Phi(H)\}$ is an orthonormal set

A is a vector of constants to ensure $S(H)>0$

$\{S(H)\}$ is the set of actual Parallel fiber activity.

2. Determine whether the connectivity of Parallel fiber-Purkinje cell synapses is modifiable. If some form of plasticity is found, determine the mathematical relation which governs this plasticity.
3. Determine whether the cerebellum functions as a phasic or as a tonic device and whether Subcortical Nuclear cells do in fact perform a "sample-and-hold" operation as proposed by this thesis.'

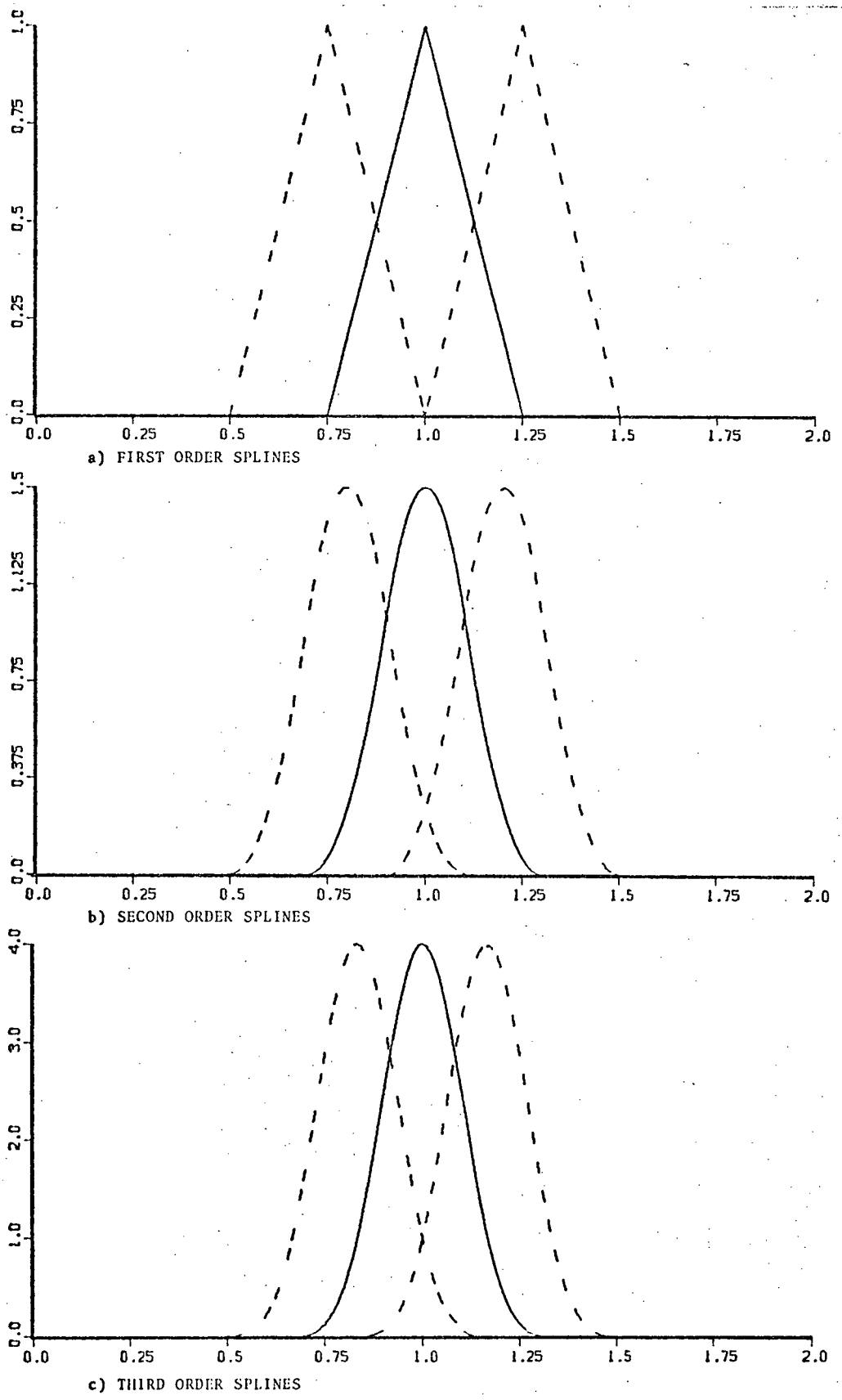


Fig 7.2 One Dimensional Spline Functions

There are also a number of mathematical questions, relating to learning machines, posed by this research. The following investigations could prove most interesting:

1. There are several properties of spline functions, in terms of basis functions for learning machines, which appear promising. As shown in Figure 7.2, each such function has a limited region of non-zero support. When considering learning algorithms, this means that weight adjustments produce error correcting functions of similarly limited support. In other words, spline-based error correcting functions would cause no interference outside of a small region. The major problem with splines is to generate a useful set of functions which produce continuous functions and do not require excessive numbers of elements [41]. Another possible problem, as discussed in Section 6.3, is the potential system failure which could result from the failure of only a few elements of the basis set generator.
2. The programs which demonstrate the effectiveness of this learning system use extended precision (64 bit), floating point, digital values in all computations. The effects of using analog, reduced precision, or noisy variables require further investigation.
3. The operational characteristics of the system in a realistic control system environment require testing.
4. There is much theoretical work required to determine strategies for training or "programming" hierarchical networks of learning systems.

5. The system is relatively expensive and slow when modeled on a general purpose, digital computer. To be most effective, the system should be constructed as a single, special purpose, device.

BIBLIOGRAPHY

Cited Literature

[1] J.S. Albus, A Theory of Cerebellar Function, Math. Biosci., 10, pp25-61, Feb 1971.

[2] J.S. Albus, Theoretical and Experimental Aspects of a Cerebellar Model, Ph.D. Thesis, University of Maryland, Dec 1972.

[3] J.S. Albus, A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC), Trans. ASME Ser.G., 97, pp200-227, Sept 1975.

[4] J.S. Albus, Data Storage in the Cerebellar Model Articulation Controller (CMAC), Trans. ASME Ser.G., 97, pp228-233, Sept 1975.

[5] J.S. Albus and J.M. Evans,Jr., Robot Systems, Scientific American, 234, pp77-86b, Feb 1976.

[6] L.A. Alekseeva and Y.F. Golubev, An Adaptive Algorithm for Stabilization of Motion of an Automatic Walking Machine, Eng. Cybern., 14, No.5, pp51-59, 1976.

[7] C.C. Bell and R.S. Dow, Cerebellar Circuitry, Neurosci. Res. Prog. Bul., 5, No.2, pp121-222, 1967.

[8] S. Blomfield, Arithmetic Operations Performed by Nerve Cells, Brain Res., 69, pp115-124, 1974.

[9] V. Braitenberg and R.P. Atwood, Morphological Observations on the Cerebellar Cortex, J. Comp. Neurol., 109, pp1-27, 1958.

[10] R.J. Brown, Adaptive Multiple-Output Threshold Systems and Their Storage Capacities, Technical Report 6771-1, Stanford Electronics Laboratories (SU-SEL-64-018), June 1964, AD 444 110.

[11] T.W. Calvert and F. Meno, Neural Systems Modeling Applied to the Cerebellum, IEEE Trans. Syst., Man, & Cybern., SMC-2, pp363-374, July 1972.

[12] J.C. Eccles, M. Ito, J. Szentagothai, The Cerebellum as a Neuronal Machine, Springer-Verlag, New York, 1967.

[13] J.C. Eccles, The Dynamic Loop Control Hypothesis of Movement Control, in Information Processing in the Nervous System, K.N. Leibovic, ed., Springer-Verlag, New York, 1969.

[14] J.C. Eccles, The Cerebellum as a Computer: Patterns in Space and Time, J. Physiol. Lond., 229, No.1, pp1-32, 1973.

[15] B.R. Gaines, Stochastic Computing Systems, in Advances in Information Systems Science, J.T. Tou, ed., Plenum Press, New York, 1969.

[16] A.R. Gardner-Medwin, The recall of events through the learning of associations between their parts, Proc. R. Soc. Lond. Ser.B, 194, pp375-402, 1976.

[17] P.F.C. Gilbert, A Theory of Memory that Explains the Function and Structure of the Cerebellum, Brain Res., 70, pp1-18, 1974.

[18] G.H. Glasser and D.C. Higgins, Motor Stability, Stretch Responses and the Cerebellum, in Nobel Symposium, I. Muscular Afferents and Motor Control, R. Granit, ed., Almquist and Wiksell, Stockholm, pp121-138, 1966.

[19] P.G. Guest, Numerical Methods of Curve Fitting, Cambridge University Press, London, 1961.

[20] M. Hassul and P.D. Daniels, Cerebellar Dynamics: The Mossy Fiber Input, IEEE Trans. Bio-Med, BME-24, pp449-456, Sept 1977.

[21] D.O. Hebb, The Organization of Behaviour: A Neuropsychological Theory, Wiley, New York, 1949.

[22] J.K.S. Jansen, K. Nicolaysen, T. Rudjord, Discharge Patterns of Neurons of the Dorsal Spinocerebellar Tract Activated by Static Extension of Primary Endings of Muscle Spindles, J. Neurophysiol., 29, pp1061-1086, 1966.

[23] J.K.S. Jansen, K. Nicolaysen, T. Rudjord, On the Firing Pattern of Spinal Neurons Activated from the Secondary Endings of Muscle Spindles, Acta Physiol. Scand., 70, pp183-193, 1967.

[24] H.H. Kornhuber, Motor Functions of Cerebellum and Basal Ganglia: The Cerebellocortical Saccadic (Ballistic) Clock, the Cerebellonuclear Hold Regulator, and the Basal Ganglia Ramp (Voluntary Speed Smooth Movement) Generator, Kybernetic, 8, No.4, pp157-162, 1971.

[25] Y. Kosugi and Y. Naito, An Associative Memory as a Model for the Cerebellar Cortex, IEEE Trans. Syst., Man, & Cybern., SMC-7, pp94-98, Feb 1977.

[26] P.D. Lawrence and W-C. Lin, Statistical Decision Making in the Real-Time Control of an Arm Aid for the Disabled, IEEE Trans. Syst., Man, & Cybern., SMC-2, pp35-42, Jan 1972.

[27] H.C. Longuet-Higgins, D.J. Willshaw, O.P. Buneman, Theories of Associative Recall, Q. Rev. Biophys., 3, No.2, pp223-244, 1970.

[28] D. Marr, A Theory of Cerebellar Cortex, J. Physiol. Lond., 202, pp437-470, 1969.

[29] W.S. Meisel, Potential Functions in Mathematical Pattern Recognition, IEEE Trans. Comput., C-18, pp911-918, Oct 1969.

[30] M.D. Mesarovic, The Control of Multivariable Systems, Technology Press and John Wiley and Sons, New York, 1960.

[31] M.D. Mesarovic, D. Macko, and Y. Takahara, Theory of Hierarchical, Multilevel Systems, Academic Press, New York, 1970.

[32] M. Minsky and S. Papert, Perceptrons, MIT Press, Cambridge, 1969.

[33] J.A. Mortimer, A Computer Model of Mammalian Cerebellar Cortex, Comput. Biol. & Med., 4, pp59-78, 1974.

[34] N.J. Nilsson, Learning Machines, McGraw-Hill, New York, 1965.

[35] E. Parzen, Modern Probability Theory and its Applications, John Wiley and Sons, New York, 1960.

[36] E. Parzen, On Estimation of a Probability Density Function and Mode, Ann. Math. Stat., 33, pp1065-1076, Sept 1962.

[37] R.G. Peddicord, A Computational Model of Cerebellar Cortex and Peripheral Muscle, Int. J. Bio-Med. Comput., 8, pp217-237, 1977.

[38] A. Pellionisz, Computer Simulation of the Pattern Transfer of Large Cerebellar Neuronal Fields, Acta Bioch. & Biophys. Hung., 5, No.1, pp71-79, 1970.

[39] A. Pellionisz and J. Szentagothai, Dynamic Single Unit Simulation of a Realistic Cerebellar Network Model, Brain Res., 49, pp83-99, 1973.

[40] A. Pellionisz and J. Szentagothai, Dynamic Single Unit Simulation of a Realistic Cerebellar Network Model. II. Purkinje Cell Activity within the Basic Circuit and Modified by Inhibitory Systems, Brain Res., 68, pp19-40, 1974.

[41] P.M. Prenter, Splines and Variational Methods, John Wiley and Sons, New York, 1975.

[42] M.H. Raibert, a Model for Sensorimotor Control and Learning, Biol. Cybern., 29, No.1, pp29-36, 1978.

[43] A. Rapoport, "Addition" and "Multiplication" Theorems for the Inputs of Two Neurons Converging on a Third, Bul. Math. Biophys., 13, pp179-188, 1951.

[44] F. Rosenblatt, the Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Psycol. Rev., 65, No.6, pp386-408, 1958.

[45] F. Rosenblatt, Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms, Spartan Books, Washington, 1961.

[46] T.C. Ruch, Basal Ganglia and Cerebellum, in Medical Physiology and Biophysics, T.C. Ruch and J.F. Fulton, eds., Saunders, Philadelphia, 1960.

[47] N.H. Sabah and J.T. Murphy, Reliability of Computations in the Cerebellum, Biophys J., 11, pp429-445, 1971.

[48] N.H. Sabah, Aspects of Cerebellar Computation, in Proceedings of the European Meeting on Cybernetics and System Research, Vienna, Transcripta, London, 1972, pp230-239.

[49] M. Simaan, Stackelberg Optimization of Two-Level Systems, IEEE Trans. Syst., Man, & Cybern., SMC-7, pp554-557, July 1977.

[50] D.F. Specht, Generation of Polynomial Discriminant Functions for Pattern Recognition, Technical Report No.6764-5, Stanford Electronics Laboratories (SU-SEL-66-029), May 1966, AD 487 537.

[51] D.F. Specht, A Practical Technique for Estimating General Regression Surfaces, Lockheed Palo Alto Research Laboratory, June 1968, AD 672 505.

[52] B. Stott, Power System Dynamic Response Calculations, Proc. IEEE, 67, pp219-241, Feb 1979.

[53] D.F. Stubbs, Frequency and the Brain, Life Sciences, 18, No.1, pp1-14, 1976.

[54] B. Widrow, N.K. Gupta, S. Maitra, Punish/Reward: Learning with a Critic in Adaptive Threshold Systems, IEEE Trans. Syst., Man, & Cybern., SMC-3, pp455-465, Sept 1973.

General References

[1] P. Cress, P. Dirksen, J.W. Graham, Fortran IV with Watfor and Watfiv, Prentice-Hall, Englewood Cliffs, New Jersey, 1970.

[2] H.W. Fowler and F.G. Fowler, eds., The Concise Oxford Dictionary, University Press, Oxford, 1964.

[3] A.C. Guyton, Textbook of Medical Physiology, W.B. Saunders, Philadelphia, 1976.

[4] P.H. Lindsay and D.A. Norman, Human Information Processing, Academic Press, New York, 1977.

[5] B. Noble, Applied Linear Algebra, Prentice-Hall, Englewood Cliffs, New Jersey, 1969.

[6] B. Rust, W.R. Burrus and C. Schneeberger, A Simple Algorithm for Computing the Generalized Inverse of a Matrix, Comm. ACM, 9, pp381-387, May 1966.

[7] S.M. Selby, ed., Standard Mathematical Tables, Chemical Rubber Co., Cleveland, 1968.