ASSOCIATIVE NEURAL NETWORKS: PROPERTIES, LEARNING, AND APPLICATIONS

By

Chi-Sing Leung

A Dissertation

submitted in partial fulfillment of the requirements

for the Degree of Doctor of Philosophy

Division of Computer Science

The Chinese University of Hong Hong

Dec 1994

Acknowledgement

I would like to express my gratitude to my supervisors, Dr. Lai-wan Chan and Dr. Ming-kit Lai, for their supervision and advice throughout the past three years. I thank also Dr. Wai-ho Mow for helpful discussions on the topics of Trellis Coded Modulation. In addition, I am ever so grateful to Kei-shiu Ho, Richard Wong, and John Sum for assisting the preparation of this manuscript. Also, I am very thankful to the thesis committee, including Dr. Lei Xu, Dr. Chung-ping Kwong, and Prof. Wan-chi Siu, for their valuable suggestions to improve this thesis.

Abstract

Associative neural networks can be classified by their encoding concepts. Accordingly, the main classes of models are distributed encoding (as Hopfield model and bidirectional associative memory (BAM)) and direct encoding (as Kohonen map). All these networks realize the associative recall processing — even with a noisy/partial input of a stored item, the network can recall the whole stored item. Among various models, the BAM and Kohonen map individually has some special features (simultaneous hetero- and autoassociative recollections for the BAM, and ordering preserve for the Kohonen map) which arouse a lot of scholars to investigate in different aspects. The objectives of this thesis are: 1) to examine the statistical properties of BAM and develop the learning algorithms for the BAM, and 2) to develop new applications of Kohonen map based on its ordering property.

The BAM is a two-layer heteroassociative memory. It uses bidirectionality (forward and backward information flow between the two layers) to achieve the hetero- and autoassociative recollections simultaneously. For any real connection matrix, it reaches a steady state during recall. Up to now, only a few theoretical analysis about its statistical properties have been reported. We will examine three statistical properties of BAM: 1)the memory capacity, 2)the number of errors in the retrieval pairs, and 3)the attraction basin for the worst case errors. Also, we will investigate how the ratio $r = \frac{p}{n}$ between the dimensions of the two layers affects its statistical properties, where p and n are the numbers of neurons in the two layers. By studying the statistical properties, one can deal with the recall performance of BAM in a more rigorous manner. When a small number of errors are allowed in the retrieval pairs, the lower bound of the memory capacity can grow as far as $\alpha_r n$, where α_r is a constant which

depends on the value of r. The number of errors in the retrieval pairs is bounded by $O(\exp\left(-\frac{r}{2(1+r)\alpha}\right)n)$ when the number of library pairs is αn. Also, each library pair has a small attraction basin for the worst case errors. For example, when $r = 1$ and the number of library pairs is small, the lower bound of the attraction basin is about $0.0068n$.

Moreover, we extend the analysis to the second order BAM, as well as the general higher order BAM. The lower bound of the memory capacity of the second order BAM can grow as far as $\alpha_r n^2$. The number of errors in the retrieval pairs is bounded by $O(\exp\left(-\frac{r^2}{6(1+r^2)\alpha}\right)n)$ when the number of library pairs is αn^2. Also, each library pair has a small attraction basin for worst case error. For example, when $r = 1$, the lower bound of the attraction basin is about $0.00587n$ for small number of library pairs.

Small memory capacity and no guarantee of correct recall (under noisy initial input) are two main problems of the original BAM encoding strategy. Hence, four new encoding methods are proposed to improve the recall performance, including the memory capacity and the error correction capability . In general, these four methods push the memory capacity to the maximum (or near maximum). The properties of these four encoding methods and the other existing methods will also be discussed and illustrated via computer simulations.

After investigating several aspects of BAM under the non-incremental encoding method, we will also examine its statistical storage behavior under the forgetting learning rule which is an incremental encoding method. The guideline (in favour of the most recent library pairs) for choosing the forgetting constant in the rule is also presented.

The Kohonen map is a self-organized network in which a neighborhood structure is introduced among the codevectors before learning. After learning, the network will have a nice property: ordering preserve. That is, when two codevectors are neighbors of each other, their Euclidean distance is usually small. In the field of communications, there is also a similar neighborhood structure among the channel waveforms. Based on this similarity, we will present three new cross-relative applications of Kohonen map for data compression and communications. The united goal is to design robust transmission systems for vector-quantized data under noisy channel.

Firstly, we use the neighborhood structure of Kohonen map to simplify the design process of a trellis type vector quantizer. The performance of our approach is comparable to that of the conventional trellis type vector quantizer. Secondly, we use the neighborhood structure of Kohonen map to carry out the association between the codevectors and the channel waveforms. By considering the association, the impulsive noise in the received data can be greatly reduced even if the communication channel is noisy. From the computer simulation, with the same root-mean-squared-error in the received data, our approaches achieve about 4-5 dB gain (signal-to-noise-ratio in the channel). Finally, based on the concepts of the above two applications, we present an error control scheme for the transmission of vector-quantized data such that the impulsive noise can be further reduced. To develop this error control scheme, identical trellis is used in both the trellis type vector quantizer and the error control block. To reduce the impulsive noise, the association between the codevectors and channel waveforms in this error control scheme is based on the concept of the second application mentioned above. The advantage of our approaches is that we do not need to design the systems again even though we use a new codebook in the systems.

Contents

Chapter 1

Introduction

This chapter begins with an introduction of the basic concepts of the associative neural networks and then discusses the typical goals in the study of the associative neural networks. Finally, a brief overview of the structure of the rest of the thesis is presented.

1.1 Background of Associative Neural Networks

Data storage methods are generally divided into two classes. The first one is random access memory in which a separate unique address points to each data item. To recall a particular data item, we should provide its address without ambiguity. The other class is associative memory in which the data item can be recalled by its noisy version or partial version [1]-[5]. It is well suited for applications that require the capability to handle noisy input such as pattern matching and pattern classification. For a review of the applications of the associative memory, readers are referred to [6].

It is useful to distinguish between two types of associative memory, the autoassociative memory, whereby an incomplete key pattern (or a noisy version of the stored pattern) is replenished into a complete version, and the heteroassociative memory which selectively produces an output Y_h in response to the input X_h (or a noisy version of X_h); in the latter case the pair (X_h, Y_h) is called library pair.

One way to implement associative memories is by using neural networks. Neural networks generally have a parallel and highly-interconnected computing architecture. They consist of many simple processing units or "neurons" which communicate with each other via connection weights. Each neuron calculates its own activation level by forming a weighted and thresholded sum of the activation levels of the connected neurons. The coefficients in the weighted sum are the connection weights between neurons. The term 'associative neural network' denotes the associative memory implemented by the neural network approach.

As described above, associative neural networks are fundamentally different from the conventional random access memories in which no separate address exists for each stored entity. Instead, the data item itself acts as a pointer either to itself (autoassociative recall) or to another stored data item (heteroassociative recall). With presentation of an initial input, the associative recall will be excited until a decision is reached in a global manner. Associative neural networks also have the error correction capability. An undistorted pattern can be retrieved with a distorted or partial version of input pattern.

Many associative neural networks have been designed and demonstrated for the task of associative recall, for example: Hopfield network [12], bidirectional associative memory (BAM) [13], Kohonen map [1, 2, 3] and many others [5, 6]. Some people may consider Kohonen map as a vector quantizer only. However, vector quantization can be regarded as a special case of associative recall [1], in which the input patterns are directly mapped to a set of finite codevectors. In other words, the codevectors are treated as the stored patterns.

In general, there are two implementation methods of associative neural networks. One is distributed encoding model, in which the stored patterns are encoded into a distributed connection matrix and we cannot directly know the stored patterns from the values of the connection weights. Typical examples of the distributed encoding model are Hopfield network and BAM. Another is direct encoding model, in which the connection weights are the values of the stored patterns and there is an one-to-one association between the connection weights and the values of the stored patterns. A typical example is Kohonen map.

Among various associative neural network models, the BAM (which is a distributed encoding associative neural network) and Kohonen map (which is a direct encoding associative neural network) individually has some special features. These features attracted a greater number of scholars to investigate the two models in different aspects. The thesis consists of two parts. In the first part, we examine the statistical properties of BAM and develop the learning algorithms for the BAM. In the second part, we propose three new cross-relative applications of Kohonen map for data compression and communications.

1.2 A Distributed Encoding Model: Bidirectional Associative Memory

In the distributed encoding models, an associative neural network is designed to map some user-selected pattern vectors $X_1, \cdots, X_m$ into some user-selected pattern vectors $Y_1, \cdots, Y_m$, respectively. These associative pairs (X_h, Y_h) are called *library pairs.* The dimension of the vectors X_h, $h = 1, \cdots, m$, is n and the dimension of the vectors Y_h, $h = 1, \cdots, m$, is p. If the input-output action of the network is described by a function ψ, then our goal is to have $\psi(X_h) = Y_h$ for $h = 1, \cdots, m$. Naturally, this vector association must be logically consistent. It means that there cannot be two different Y_h's assigned to the same X_h.

To understand the distributed encoding models, we briefly introduce two types of distributed encoding models: feedforward and feedback. In a feedforward associative neural network (see Figure 1.1), presentation of an input vector X in the layer F_X leads to the output Y in the layer F_Y in a feedforward pass. For example, the linear associative memory [14] is a feedforward associative neural network. In the feedback case (see Figure 1.2), an initial pattern $X^{(0)}$ presented to F_X will be passed through the connection matrix and then thresholded, and a new state $Y^{(1)}$ in F_Y is obtained; the state $Y^{(1)}$ in F_Y is then passed back through the connection matrix and is thresholded, giving rise to a new state $X^{(1)}$ in F_X; and the process repeats. In other words, the insertion of $X^{(0)}$ will excite the feedback loop of the network. The basic idea is that the state of the network should converge (at least asymptotically) to a "fixed point" which is then read out as the final output of the network. For example, the BAM is a feedback associative neural network.

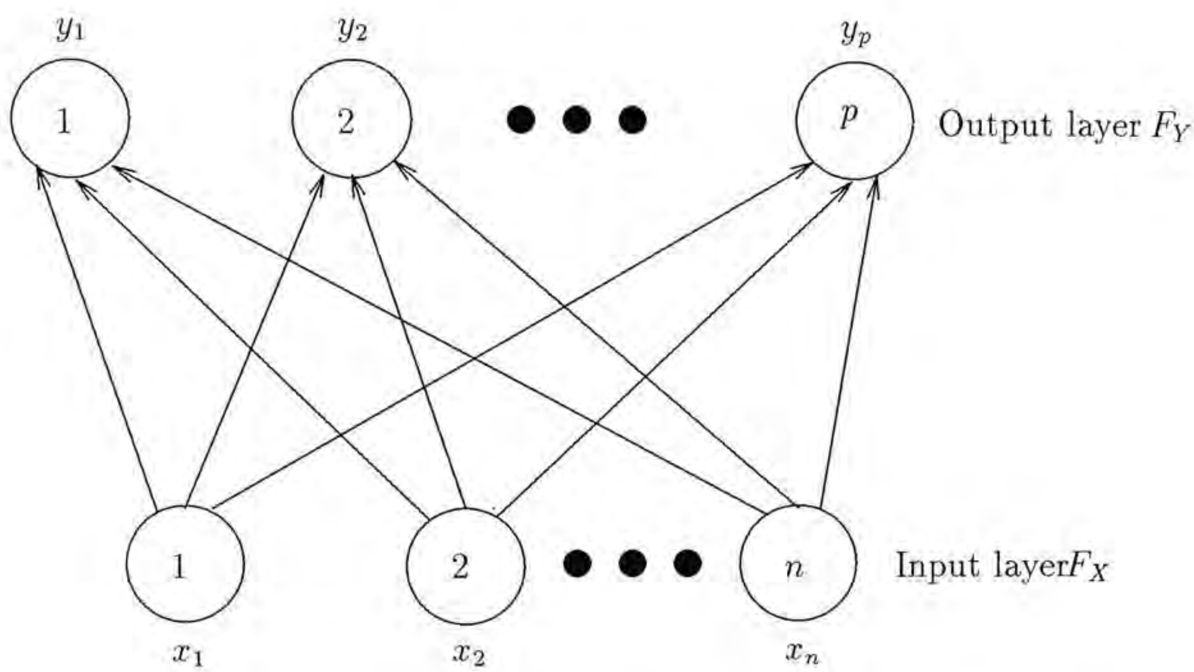


Figure 1.1 A typical feedforward associative network. The input vector X leads to the output vector Y in a single feed-forward pass.

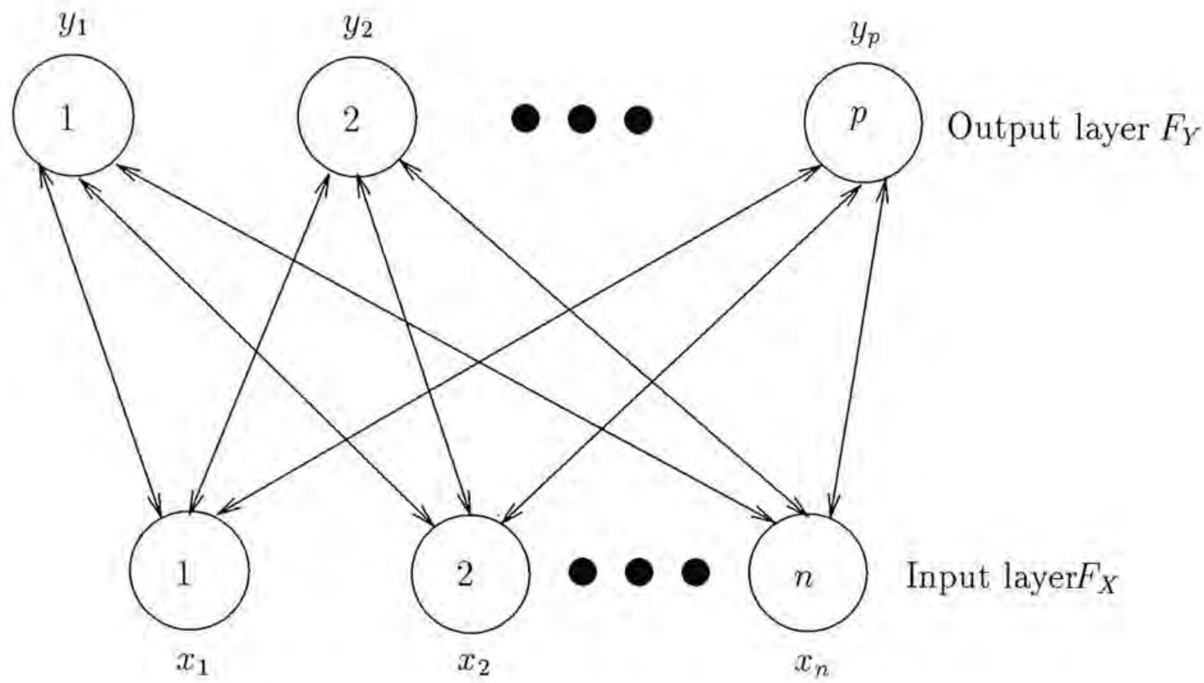


Figure 1.2 A typical layer feedback associative network. The input vector X initiates a feedback evolution between layers F_X and F_Y.

Apart investigating the applications of the distribution encoding models, the typical goals in the study of these models are:

1. To understand how large m (the number of library patterns/pairs) can be, given correct and robust operation, for a model with a given number of neurons and a given learning rule (see Chapter 3, 4, and 6).

2. To understand what the error correction capability is, given m, for a model with a given number of neurons and a given learning rule (see Chapter 3 and 4).

3. For a given model with a given number of neurons, how to construct the connection matrix such that the library patterns/pairs can be correctly stored with a good error correction capability (see Chapter 5). This task can be regarded as *supervised learning.*

The BAM being a distributed encoding model is a two-layer heteroassociative memory. It uses the bidirectionality (forward and backward information flow between the two layers) to search for the library pairs. Firstly, its architecture is able to achieve the simultaneous hetero- and autoassociative recollections. Secondly, the searching is also bidirectional. It means that the initial searching key can be presented in one of the two layers. Also, it is unconditionally stable during recall. In other words, for any real connection matrix, its state will converge to a stable state during recall. Nowadays, several applications of BAM have been reported [20]-[22]. For example, it can be used in intelligent systems [20] and spectral signature recognition [22]. For a brief review of its applications, readers are referred to [6]. The electronic and optical implementation of BAM are also available [55]–[57].

However, only a few theoretical analysis about its statistical properties (for example, the memory capacity, the attraction basin, and the number of errors in the retrieval pairs) have been reported [23, 24, 25, 66]. The lack of the theoretical analysis makes the recall performance of BAM to be an unknown issue. This phenomenon is prominent especially in the higher order BAM. By studying its statistical properties , we can deal with its recall performance in a more rigorous manner. Some preliminary theoretical results and empirical results have shown that its recall performance under the Kokos's encoding method is poor [23]–[24] [32]–[53]. Hence, many researchers and me have reported several modifications [32]–[53]. These modifications usually give improvement on the recall performance with extra costs, such as increased computation and hardware complexity. Note that some modifications are not stable during recall.

1.3 A Direct Encoding Model: Kohonen Map

In the direct encoding, the associative neural network usually divides the space of input vectors into a finite number of disjoint regions (the number of disjoint regions is denoted as m) and a codevector is assigned for each region. Thus, there are m regions and m codevectors. The output of the associative neural network is the codevector whose corresponding region contains the input vector. Then, we can use this codevector to represent the input. This concept is usually regarded as vector quantization [15, 16]. Mathematically, the associative neural network is a mapping Q

from n dimensional Euclidean space $\Re^n$ to a finite subset of $\Re^n$:

$$Q : \Re^n \mapsto \{Y_1, \cdots, Y_m\}, \tag{1.1}$$

where $Y_1, \cdots, Y_m$ are the codevectors. The associative neural network which implements the above function is regarded as vector quantizer. To measure the quality of the codevectors (vector quantizer), we should first define a distortion measure. It is an assignment of a cost $d(X_{in}, Q(X_{in}))$ of reproducing the input X_{in} as the output $Q(X_{in})$. Usually, the squared error distortion is used:

$$d(X_{in}, Q(X_{in})) = (X_{in} - Q(X_{in}))^T (X_{in} - Q(X_{in})). \tag{1.2}$$

Given a distortion measure, the quality of the codevectors is the average distortion between the input vector and the corresponding codevector.

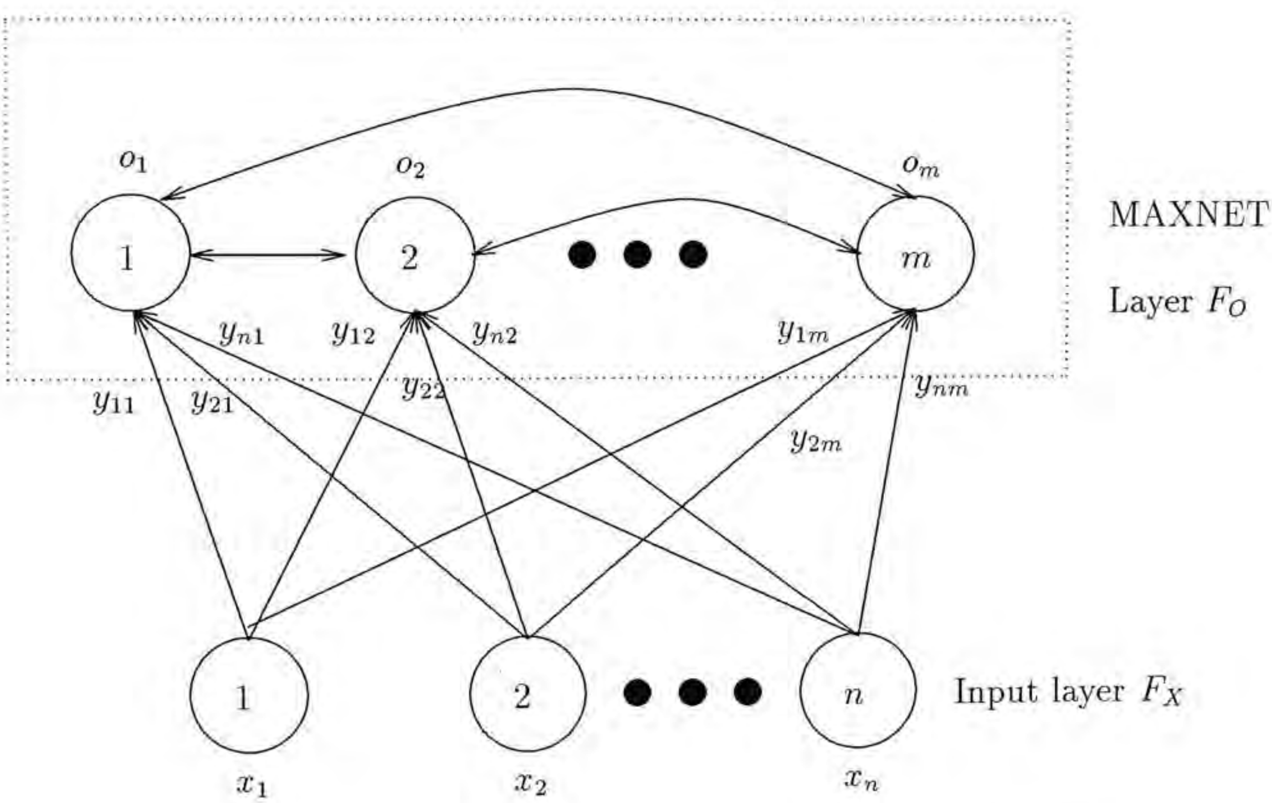


Figure 1.3 A typical direct encoding associative neural network.

Figure 1.3 shows the implementation of this type of associative neural networks. The number of neurons in the layer F_O is m and the number of neurons in the layer F_X is n. Each neuron in the layer F_O corresponds to a codevector. The connection weights from the layer F_X to the h-th neuron in the layer F_O are the components of the codevector Y_h. There are lateral inhibitions in the layer F_O. These lateral inhibitions are used to sort out the largest and to suppress the others. The operation is as follows:

1. An input vector X is presented in F_X.

2. According to the connection weights between the two layers F_X and F_O, the h-th neuron in the layer F_O get a value o_h which is the similarity measure between the input X and the codevector Y_h.

3. The lateral inhibitions, working as a MAXNET [17], find out the neuron in the layer F_O with the largest o_h value.

4. The connection weights (the codevector) whose corresponding o_h value is the largest is then directly feedback to the input layer F_X. Finally, the codevector Y_h appears in the layer F_X.

Since the connection weights are the components of the codevector in the direct encoding, we can easily encode the codevectors into the connection weights. The typical goals in the study of the direct encoding models are:

1. How to construct the codevectors from a set of training samples in the input space such that the average distortion of reproducing the input vector as the corresponding codevector is minimized. The construction process of the codevectors is commonly regarded as *clustering* or *unsupervised learning.*

   There have already been a number of techniques for designing the codevectors, such as Kohonen map [1] and LBG [19]. However, up to now, it is difficult to determine which technique is the best in terms of the distortion minimization.[1]

2. How to utilize this type of associative neural networks for data compression and transmission (see Chapter 8).

[1]We will use a computer simulation to illustrate this in Section 8.2.

Kohonen map is a direct encoding model and can be used as a vector quantizer. In Kohonen map, a neighborhood structure is introduced among the codevectors before learning. After learning, the network will have a nice property: ordering preserve. That is, when two codevectors are neighbors of each other, their Euclidean distance is usually small. This property has been theoretically investigated [8]-[10]. Based on this property, Kohonen map can produce phoneme strings for word recognition in speech recognition [7] and is a good way to reduce the dimensionality of the input in pattern recognition [11].

In the field of communications, there also exists a similar neighborhood structure among the channel waveforms in the signal space. This structure is created on the basis of the concept of the Delaunay neighborhood [82, 83]. Because of this similarity, we believe that Kohonen map has some potential applications in the field of communications [29]–[31].

1.4 Scope and Organization

As shown in Figure 1.4, this thesis consists of two parts. The first part examines the different theoretical aspects of BAM. These aspects include the statistical properties (such as the memory capacity, the attraction basin of each library pair, and the number of errors in the retrieval pair) and the new learning rules which are used to improve the memory capacity and the error correction capability. In the second part, we propose three new cross-relative applications of Kohonen map based on its ordering property.

In Chapter 2, we first give an overview of BAM's encoding algorithm, recall operation, and stability. We also review its memory capacity and error correction capability in the statistical sense when error is not allowed in the retrieval pairs.

Chapter 3 presents the statistical properties of the first order BAM when a small number of errors in the retrieval pairs are allowed. This study is motivated by the well known property of Hopfield network: if a small number of errors in the retrieval pattern are allowed, the memory capacity of Hopfield network can grow as fast as

αn^q, where α is a positive constant, q is the order of the connections and n is the dimension [12, 59]. We want to examine whether the similar results can be obtained for the BAM. Also, we want to understand how the ratio $r = \frac{p}{n}$ of the dimensions affects the the statistical properties of BAM. In Chapter 3, the memory capacity of the first order BAM is first investigated based on the concept of energy barrier, which was originally used to analyze the memory capacity of Hopfield network [59]. Note that the result of such analysis only implies that there exist some stable noisy versions around each library pair. However, the attraction basin of each library pair still cannot be determined. To overcome this limitation, the statistical dynamics of the BAM is presented. We then discuss a way to estimate the attraction basin, the memory capacity, and the number of errors in the retrieval pairs from this statistical dynamics.

Chapter 4 presents the statistical properties of the second order BAM [25]. We first use an example to illustrate that the state of the second order BAM may converge to limit cycles. Also, we explain why the energy function in [59, 45] cannot be used to explain the stability of the second order BAM. Hence, the approach of energy barrier in [59] is not suitable to derive the memory capacity of the second order BAM. The approach of the statistical dynamics used in Chapter 3 is employed again to estimate the attraction basin, the memory capacity, and the number of errors in the retrieval pairs of the second order BAM. The extension of the results to the general higher order BAM is also presented.

In Chapter 5, we propose four new encoding methods [32]-[38] to enhance the recall performance (the memory capacity and the error correction capability) of BAM. Such properties as the memory capacity, the error correction capability, the ease of hardware implementation, the learning speed, the information ratio and the convergent conditions are discussed. Also, we give a comparison on the properties among the four encoding methods and some other existing encoding methods.

After investigating several aspects of BAM under the non-incremental encoding method, we also examine its storage behavior under the forgetting learning [27] in Chapter 6. We first estimate the probability that the last k-th previous library pair is stored as a fixed point and then we derive a guideline for choosing the forgetting constant such that the number of most recent library pairs being correctly stored is near maximal.

As the development of new applications of Kohonen map needs some knowledge on the Kohonen map and vector quantization, we describe the basic concepts of them in Chapter 7.

In Chapter 8, we then introduce our three new cross-relative applications of Kohonen map for data compression and communications based on its ordering property [29, 31]. The united goal is to design robust transmission systems for vector-quantized data under noisy channel.

Firstly, a new trellis type quantizer called the trellis coded Kohonen map (TCKM) is presented [29]. Its design process, which is based on the neighborhood structure of Kohonen map, is simpler than the conventional trellis coded vector quantizer (TCVQ) [81]. Secondly, a novel transmission system for vector-quantized data is presented. Hence, the impulsive noise in the received data can be greatly reduced under a noisy channel [31]. Lastly, we present an error control transmission system for vector-quantized data based on *the concepts of the above two applications.* In this system, the source coding (the vector quantizer), the error control, and the modulation are designed as a whole such that the impulsive noise in the received vector-quantized data can be further reduced.

Remark: A chapter summary is given to summarize the results of each chapter (except Chapter 1 and Chapter 9). Since the thesis contains two parts, the notations of the second part are different from that of the first part. It means that we use a new set of notations in Chapter 7 and Chapter 8.

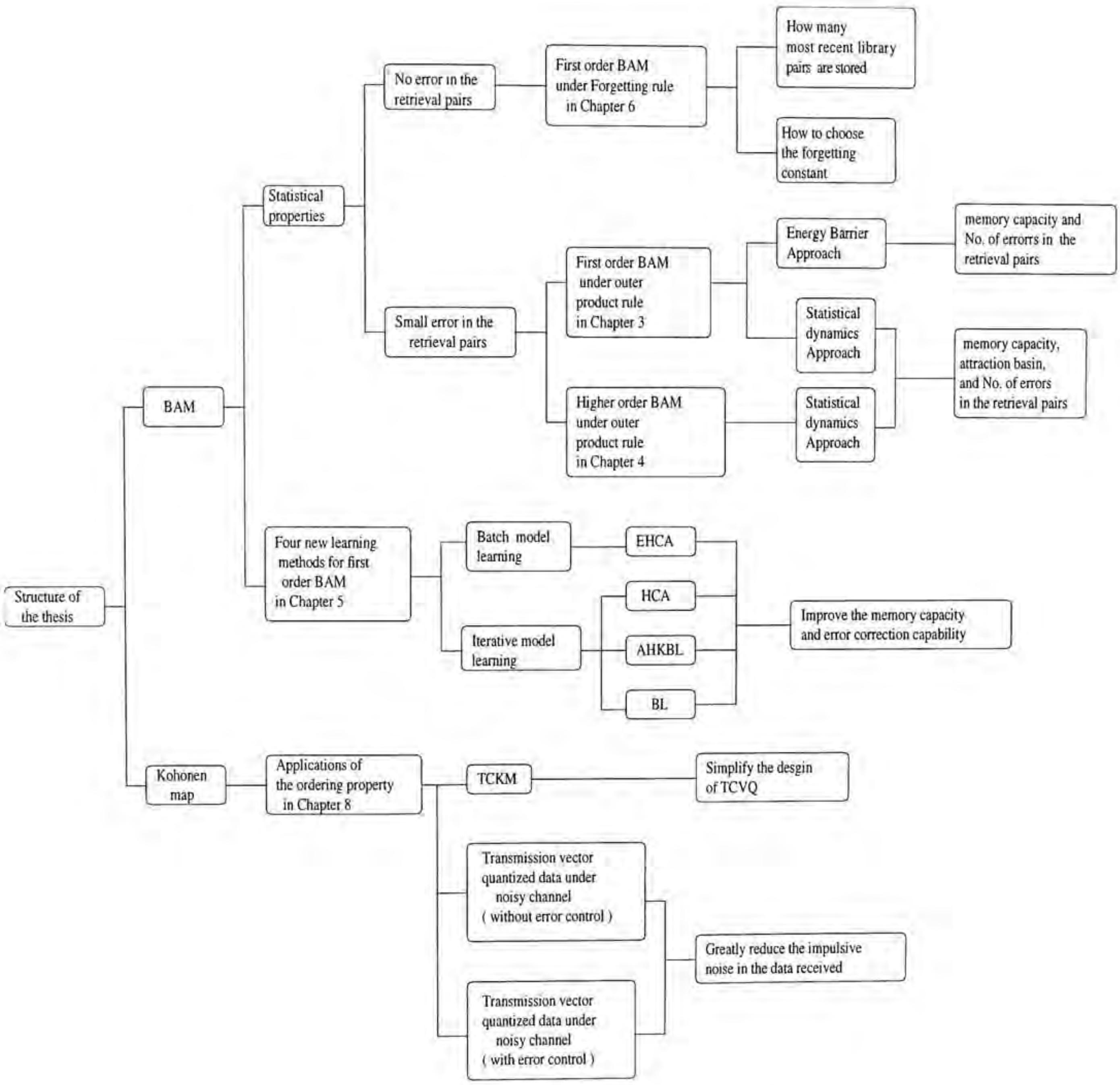


Figure 1.4 The structure of the thesis

1.5 Summary of Publications

- [25] C. S. Leung, L. W. Chan, and M. K. Lai,"Stability, Capacity, and Statistical Dynamics of Second Order BAM Bidirectional Associative Memory," to appear in *IEEE Trans. Syst. Man, and Cybern.*, Vol.25 No. 10, 1995.

  [26] C. S. Leung, L. W. Chan, and M. K. Lai,"Stability and Statistical Properties of Second Order BAM Bidirectional Associative Memory," submitted to *IEEE Trans. Neural Networks.*

  In [25] and [26], the statistical properties of the second order BAM, as well as the general higher order BAM, are presented. We first use an example to illustrate that the state of the second order BAM may converge to limit cycles. We present the statistical dynamics of the second order BAM. From the statistical dynamics, we then discuss a way to estimate the memory capacity, the number of errors in the retrieval pairs, and the attraction basin. Numerical examples are given to illustrate how the ratio of the dimensions affects the statistical properties. The extension of the results to the general higher order BAM is also presented.

- [32] C. S. Leung and K. F. Cheung," Householder Encoding for Discrete Bidirectional Associative Memory Associative Memory," in Proc. IJCNN 91 Singapore, Vol. 1, pp. 237–241,1991.

  [33] C. S. Leung, "Encoding Method for Bidirectional Associative Memory using projection on convex sets," in *Proc. IJCNN 92 Beijing*, Vol. 2 pp. 81–85, 1992.

  [34] C. S. Leung, "Encoding Method for Bidirectional Associative Memory using Projection on Convex Sets," *IEEE Trans. Neural Networks,* Vol. 4, September, pp.879–881, 1993.

  In [32]-[34], we are concerned with two encoding methods, householder encoding algorithm (HCA) and enhanced householder encoding algorithm (EHCA), for the BAM. From the simulations, both the memory capacity and error correction capability can be greatly improved by the HCA. Theoretically, the memory capacity of HCA tends to the dimensions of BAM. However, in BAM with HCA there are two different connection matrices. Hence, the state of BAM with HCA may not converge to fixed points during recall and the number of the connections is double. The EHCA is further developed on the basis of HCA and the projection on convex sets (POCS) [72, 73]. In the EHCA, the two connection matrices found by the HCA are reduced into one matrix by the POCS. Hence, the stable property of BAM can be maintained and the number of connections is the same as that of the original BAM. Simulation results show that the recall performance of EHCA is comparable to that of HCA.

- [35] C. S. Leung, "Optimum Learning for Bidirectional Associative Memory in the Sense of Capacity," *IEEE Trans. Syst. Man, and Cybern.* , Vol 24, No. 5, pp.791-796, 1994.

  [36] C. S. Leung, "Optimum Learning Rule in Bidirectional Associative Memory," in *Proc. of the 2th Pacific Rim International Conference on AI 1992*, Vol 2, pp.940–946, 1992.

  [37] Andrew C. S. Leung and M. Klassen, "A Delta–Rule Encoding for Bidirectional Associative Memory," in *Proc. IJCNN 91 Seattle*, Vol. 2, pp. 954,1991.

  [38] C. S. Leung, "Robust Learning Rule for Bidirectional Associative Memory," in *Proc. IJCNN 93 Nagoya*, Vol. 3 pp. 2686-2689, 1993.

  In [35]-[38], bidirectional learning (BL) is proposed to enhance the recall performance for the BAM based on the perceptron learning [4]. By modifying the proof of convergence of perceptron, we have proved that the BL yields one of the solution connection matrices (such that all library pairs are stored as fixed points) within a finite number of iterations (if the solutions exist). Hence, the memory capacity of BL is larger than or equal to that of other learning rules. Unfortunately, the error correction capability of BL is still poor. A robust learning rule, named adaptive Ho-Kashyap bidirectional learning (AHKBL), is then proposed to enhance the error correction capability. The sufficient convergent conditions of AHKBL are derived. Simulation results show that both the AHKBL and BL greatly improve the memory capacity. In particular, with the

AHKBL the error correction capability is improved.

- [27] C. S. Leung, "Forgetting Learning: Can the last k-th previous pattern be stored as a fixed point in Associative Memory ?", in *Proc. ICONIP'94-Seoul*, pp. 1086–1089, 1994.

  In [27], we have studied the storage behavior of BAM under the forgetting learning. We first estimate the probability that the last k-th previous library pair is stored as a fixed points. Then, we derive a guideline to choose the forgetting constant such that most recent library pairs are correctly stored, with high probability.

- [29] C. S. Leung, "Trellis Coded Kohonen Map", in *Proc. ICONIP'94-Seoul*, pp. 955-959, 1994.

  [30] C. S. Leung, "Design Trellis Coded Vector Quantizer using Kohonen Map", to be submitted .

  In [29, 30], a novel trellis type called the trellis coded Kohonen map (TCKM) is presented. The design process of TCKM is simpler than that of the conventional trellis coded vector quantizer (TCVQ). Although the performance of TCVQ is much better than that of non-trellis type vector quantizer, its design process which is based on the Euclidean distances between codevectors involves a certain amount of both computational and space overhead. In the TCKM, the design process of trellis is based on the neighborhood structure of Kohonen map. As the neighborhood structure of Kohonen map is predefined, different TCKMs with the same neighborhood structure but different codebooks can share the same trellis. Hence, the design process of trellis in the TCKM is simpler than that of TCVQ, and the TCKM is more suitable to operate under adaptive environment. Simulation results show that the performance of TCKM is comparable to that of TCVQ.

- [31] C. S. Leung, "Kohonen Map: Combined Vector Quantization and Modulation", in *Proc. ICONIP'94-Seoul*, pp. 242-247, 1994.

In [31], we present a novel vector quantization transmission system in which the impulsive noise in the received data is greatly reduced under a noisy channel. The key point is that the neighborhood structure of Kohonen map should match the neighborhood structure of the channel waveforms in the communication system. Simulation results show that the impulsive noise in the vector-quantized data received is greatly reduced under a noisy channel even if we do not use an error control scheme. Moreover, we have designed an error control scheme which best fits for the above system. Then, the source coding, the error control, and the modulation are designed as a whole such that the impulsive noise in the received vector-quantized data can be further reduced under a noisy channel.

- [39] C. S. Leung, "The Performance of Dummy Augmentation Encoding," in *Proc. IJCNN 93 Nagoya*, Vol. 3 pp. 2674-2677, 1993.

  In [39], we have investigated the statistical memory capacity of dummy augmentation encoding (DAE) for a given number of additional neurons and then evaluate the efficiency of DAE in terms of information ratio.

- [28] C. S. Leung, "Memory capacity and Statistical Dynamics of the First Order Bidirectional Associative Memory," in preparation.

  In [28], we have analyzed the statistical properties of the first order BAM under the condition that a small number of errors in the retrieval pairs are allowed. The memory capacity and the number of errors in the retrieval pairs are first investigated. The approach used here is the concept of energy barrier which was originally used to analyze the memory capacity of Hopfield network [59]. However, the attraction basin for the first order BAM cannot be determined from this approach. To overcome this limitation, the statistical dynamics of the number of errors has been presented. We then discuss a way to estimate the memory capacity, the number of errors in the retrieval pairs, and the attraction basin from the dynamics. Numerical examples are given to illustrate how the ratio of the dimensions affects the statistical properties.

Part I

Bidirectional Associative Memory: Statistical Properties and Learning

Chapter 2

Introduction to Bidirectional Associative Memory

In this chapter, a review of BAM's encoding algorithm, recall operation, stability, and memory capacity (without error in the retrieval pairs) is first presented. We then discuss its error correction capability.

2.1 Bidirectional Associative Memory and its Encoding Method

Associative memory is one of the major research issues in neural networks with a wide range of applications such as content addressable memory and pattern recognition. Associative memories encode and recall library patterns or pattern pairs. If the associative memory encodes single patterns, it is called an *autoassociative memory.* If it encodes pattern pairs, it is called a *heteroassociative memory.* One form of heteroassociative memories is the bivalent additive BAM [13]. This model is a discrete version of BAM : feedback and two-valued (or bipolar-valued). As we are mainly concerned with the bivalent additive BAM, the term BAM denotes the bivalent additive BAM throughout this thesis.

The BAM, as proposed by Kosko [13], is a two-layer nonlinear feedback heteroassociative memory in which m library pairs $(X_1, Y_1), (X_2, Y_2), \cdots, (X_m, Y_m)$ are stored, where $X_h \in \{-1, 1\}^n$ and $Y_h \in \{-1, 1\}^p$. In this thesis, the term "library pair" denotes the particular pattern pair which we want to store in the BAM. The topology, as shown in Figure 2.1, encodes the interlayer connections between the F_X and F_Y layers in the connection matrix W. The layers F_X and F_Y have n and p neurons, respectively.

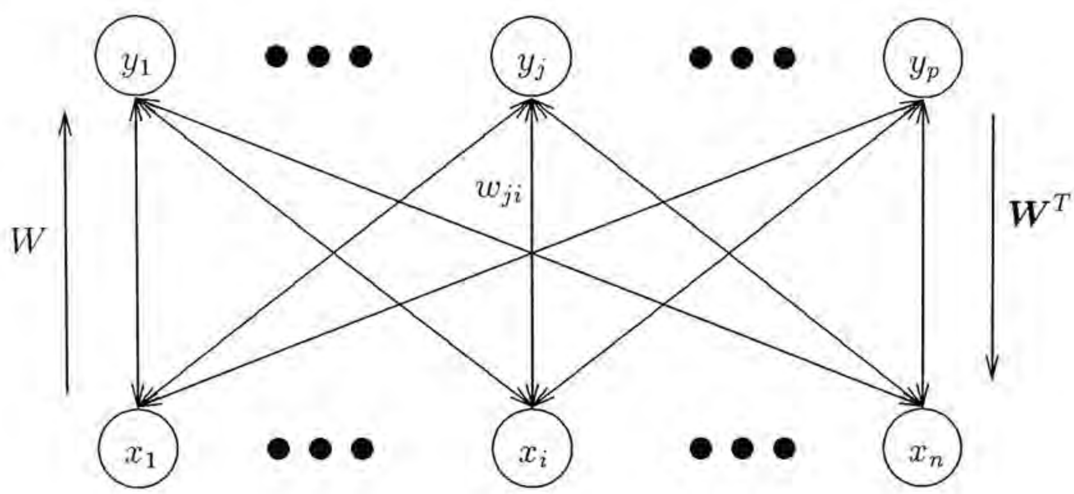


Figure 2.1 The topology of BAM is a two-layer neural network. Flow of signal is inter-layer feedback between the layers F_X and F_Y through the matrix W.

The encoding equation, as proposed by Kosko, is

$$W = \sum_{h=1}^{m} Y_h \, X_h{}^T \tag{2.1}$$

which can be rewritten as

$$w_{ji} = \sum_{h=1}^{m} x_{ih} y_{jh} \, , \tag{2.2}$$

where $X_h = (x_{1h}, x_{2h}, \cdots, x_{nh})^T$ and $Y_h = (y_{1h}, y_{2h}, \cdots, y_{ph})^T$. This encoding method is also called the *outer product rule.*

Although the above construction of W is similar to that employed in the linear associative memory (LAM) [14] and the correlation matrix memory (CMM) [3], the LAM and CMM encode real-valued library pairs and their recall procedure is purely 'one-shot' only. On the other hand, BAM encodes bipolar library pairs and its recall procedure is an interlayer nonlinear feedback process between the layers F_X and F_Y.

2.2 Recall Process of BAM

Conventionally, heteroassocaitve memories are 'one-shot' memories. Input pattern X is presented to the memory, Y is output, and then the process is finished. Hopefully, the output Y will be closer to library pattern Y_h than to all other library patterns $Y_{h'}$ if the input X is the closest to library pattern X_h.

However, the recall process of BAM employs interlayer feedback. An initial pattern $X^{(0)}$ presented to F_X is passed through W and is thresholded, and a new state $Y^{(1)}$ in F_Y is obtained which is then passed back through W^T and is thresholded again, leading to a new state $X^{(1)}$ in F_X. The process repeats until the state of BAM converges. Mathematically, the recall process is:

$$Y^{(t+1)} = \mathrm{sgn}\left(W X^{(t)}\right), \tag{2.3}$$

$$X^{(t+1)} = \mathrm{sgn}\left(W^T Y^{(t+1)}\right), \tag{2.4}$$

where $\mathrm{sgn}(\cdot)$ is the sign operator:

$$\mathrm{sgn}(x) = \begin{cases} +1 & x > 0 \\ -1 & x < 0 \\ \text{state unchanged} & x = 0 \end{cases} .$$

Using an element-by-element notation, the recall process can be written as:

$$y_i^{(t+1)} = \mathrm{sgn}\left(\sum_{i=1}^{n} w_{ji} x_i^{(t)}\right), \tag{2.5}$$

$$x_j^{(t+1)} = \mathrm{sgn}\left(\sum_{j=1}^{p} w_{ji} y_i^{(t+1)}\right), \tag{2.6}$$

where $x_i^{(t)}$ is the state of the ith F_X neuron and $y_j^{(t)}$ is the state of the jth F_Y neuron. The above bidirectional process produces a sequence of pattern pairs $(X^{(t)}, Y^{(t)})$: $(X^{(1)}, Y^{(1)})$, $(X^{(2)}, Y^{(2)})$, $\cdots$. This sequence converges [1] to one of the fixed points

[1]The proof of this property will be shown in the next section.

(X^f, Y^f) and this fixed point ideally should be one of the library pairs or nearly so. A fixed point (X^f, Y^f) has the following properties:

$$Y^f = \operatorname{sgn}(W X^f) \quad (2.7)$$

$$X^f = \operatorname{sgn}(W^T Y^f). \quad (2.8)$$

The output Y^f in F_Y represents heteroassociative recollection and the second output X^f in F_X represents autoassociative recall. Hence, the simultaneous hetero- and autoassociative recollections can be achieved.

In the above recall process, we assume that the change of state is *layer-synchronous*: an entire layer of neurons is updated at the same time. The other extreme is *random-asynchronous recall*: only one of the neurons is randomly selected and is updated at a time. Another state-processing decision is *deterministic-asynchronous*: neurons are updated one-by-one in a deterministic order. In general, the state-processing decision is the *layer-subset-asynchronous* recall: one subset of the neurons per layer is updated at a time. When the subset represents a whole layer (F_X or F_Y) of neurons, the layer-synchronous process results. In practice we usually use layer-synchronous recall in the BAM because the state of one of the two layers is initially unknown.[2]

To sum up, we have mentioned four recall processes: layer-synchronous, randomly-asynchronous, deterministic-asynchronous, and layer-subset-asynchronous. Here, we do not consider the case: both F_X and F_Y neurons are updated at the same time. It is because this recall process leads to the same problem in the asynchronous recalls: the retrieval pair (X^f, Y^f) becomes nearly totally random. Also, with this recall process, the concept of layer in the BAM is lost and the BAM becomes one kind of *synchronous sparse autoassociative memory*.[3] Hence, without further notice, the layer-synchronous recall is assumed to be use in the BAM throughout this thesis. However, there are two common facts under the four recall processes.

[2]If we use any asynchronous recalls in the BAM, we should randomly initiate the unknown layer. Then the recalled pattern pair (X^f, Y^f) becomes nearly totally random when the known neurons are updated first.

[3]A sparse memory is a memory in which large number of connections are valued zero.

Fact 2.1 *Given any real connection matrix W, the state of BAM will converge to one of the fixed points (X^f, Y^f) under each of the four recall processes.*

Fact 2.2 *Given any real connection matrix W, if a pattern pair (X^f, X^f) is a fixed point under one of the four recall processes, then it is also a fixed point under the other three recall processes.*

The proof of Fact 2.1 will be reviewed in the next section. Fact 2.2 can be easily observed from the definition of fixed point (see (2.7) and (2.8)).

2.3 Stability of BAM

The stability can be proved by considering the energy function [13]

$$E = -Y^T W X \tag{2.9}$$

where Y and X are the current states of F_Y and F_X, respectively. The change of energy with respect to the change of state in F_Y is

$$\begin{aligned} \Delta E_Y &= -(Y^T + (\Delta Y)^T) W X + Y^T W X \\ &= -(\Delta Y)^T W X \,, \end{aligned} \tag{2.10}$$

where $\Delta Y = (\Delta y_1, \Delta y_2, \cdots, \Delta y_p)^T$. Using element-by-element notation, the change of energy is

$$\Delta E_Y = -\sum_{j=1}^{p} \Delta y_j \sum_{i=1}^{n} w_{ji} x_i \,. \tag{2.11}$$

Let us consider the following three cases of Δy_j.

- If Δy_j is zero, then $\Delta y_j \sum_{i=1}^{n} w_{ji} x_i = 0$.
- If $\Delta y_j > 0$, this means that the state of the jth neuron in F_Y changes from -1 to 1 and $\sum_{i=1}^{n} w_{ji} x_i$ must be greater than zero. Hence, $\Delta y_j \sum_{i=1}^{n} w_{ji} x_i$ must be greater then zero.

- If $\Delta y_j < 0$, the state of the jth neuron in F_Y changes from 1 to -1 and $\sum_{i=1}^{n} w_{ji}x_i$ must be less than zero. Hence, $\Delta y_j \sum_{i=1}^{n} w_{ji}x_i$ must be greater then zero.

The change of state in F_Y leads to a reduction in energy no matter how many neurons in F_Y are updated at a time. We can also verify that the change of state in F_X leads to a decrease in energy no matter how many neurons in F_X are updated at a time.[4] Since the energy is lower bounded, the state must converge to one of the fixed points, which is a local minimum of the energy function. The formal definitions of a local minimum are given below.

Definition 2.1 *Given two bipolar pairs (X, Y) and (X', Y'), (X', Y') is called the neighbor of (X, Y) if the Hamming distance between them is one.*

Definition 2.2 *A state (X, Y) is a local minimum of the energy function, if its energy is less than or equal to the energy of all neighborhood pairs (X', Y')'s, i.e., for all neighborhood pairs (X', Y') of (X, Y)*

$$- Y^T W X \leq -Y'^T W X'. \tag{2.12}$$

Definition 2.3 *A state (X, Y) is an isolated local minimum of the energy function, if its energy is less than the energy of all its neighborhood pairs (X', Y')'s, i.e., for all neighborhood pairs (X', Y') of (X, Y)*

$$- Y^T W X < -Y'^T W X'. \tag{2.13}$$

Note that the state of BAM cannot converge to a limit cycle. It is because the change of state leads to a change of energy being 'less than zero', not 'less than or equal to zero'.

[4] Here, we neglect the case: both F_Y and F_X are updated at a time. Therefore, we can only say that the BAM is stable for the four recall processes:layer synchronous, randomly asynchronous, deterministic asynchronous, and layer-subset asynchronous.

2.4 Memory Capacity of BAM

In the field of associative memories, one interesting topic is the maximum number of library pairs/patterns that can be stored as fixed points in the models. This gives us a non-rigorous definition of memory capacity. Without any assumption on the library pairs, we may easily get some discouraging results about the Kosko's encoding method: the memory capacity of BAM under Kosko's encoding method is about 2 or 3 only even for very large p and n.[5]

The conventional definition of the memory capacity is the maximum number of library patterns (or pairs) that can be stored as fixed points with high probability [61]–[63]. The assumption is that each component of the library pairs/patterns is a ± 1 equiprobable independent random variable. In [24], Haines and Hecht-Nielsen *et al.* stated that the memory capacity of BAM is $\frac{\min(n,p)}{2\log\min(n,p)}$, but the formal proof was not given. Similar result was also presented in [23].

In this section, we review the issue of the memory capacity from different perspectives. Two cases of the memory capacity are reviewed in this section. The first one is a stronger concept. That is, with high probability, each library pair is stored as a fixed point if m is less than or equal to $\frac{\min(n,p)}{4\log\min(n,p)}$. The second one is a relatively weaker concept. That is, with high probability, a library pair is stored as a fixed point if m is less than or equal to $\frac{\min(n,p)}{2\log\min(n,p)}$. A separate proof will be given in each case.

Assumptions and notations

- Kosko's encoding method (outer product rule) is used.
- $p = rn$, where r is a positive constant. Also, the dimensions (n and p) are very large.
- Each component of the original library pairs (X_h, Y_h) is a ± 1 equiprobable independent random variable.
- $EU_{j,h}$ is the event that $\mathrm{sgn}(\sum_i^n w_{ji} x_{ih}) = y_{jh}$ and $\overline{EU}_{j,h}$ is the complement event of $EU_{j,h}$.

[5]Examples can be found in [41, 42].

- $EV_{i,h}$ is the event that $\text{sgn}(\sum_j^p w_{ji}y_{jh}) = x_{ih}$ and $\overline{EV}_{i,h}$ is the complement event of $EV_{i,h}$.

- We neglect the cases: $\sum_i^n w_{ji}x_{ih} = 0$ and $\sum_j^p w_{ji}y_{jh} = 0$. Because the probabilities of these cases are very low as $n \to \infty$.

With the above assumptions and notations, we easily get the following two lemmas.

Lemma 2.1 *The probability Prob($\overline{EU}_{j,h}$) is*

$$Q\left(\sqrt{\frac{n}{(m-1)}}\right)$$

for $j = 1, \ldots, p$ and $h = 1, \ldots, m$. $Q(z)$ is defined as

$$Q(z) = \frac{1}{\sqrt{2\pi}} \int_z^{\infty} exp(\frac{-z^2}{2})dz\,.$$

Proof of Lemma 2.1

Without loss of generality, we consider that the library pair (X_h, Y_h) has only positive components. That is, $X_h = (1, \ldots, 1)^T$ and $Y_h = (1, \ldots, 1)^T$. From (2.1),

$$\sum_{i=1}^{n} w_{ji}x_{ih} = n + \sum_{h' \neq h}^{m} y_{jh'} \sum_{i=1}^{n} x_{ih'}. \tag{2.14}$$

The terms $(y_{jh'} \sum_{i=1}^n x_{ih'})$'s can be considered as the noise terms. For large n, the distribution of $\frac{y_{jh'} \sum_{i=1}^n x_{ih'}}{\sqrt{n}}$ approaches standard normal. Since the sum of independent normal random variables is still normal, the distribution of the overall normalized noise $\frac{\sum_{h' \neq h}^m y_{jh'} \sum_{i=1}^n x_{ih'}}{\sqrt{n}}$ approaches normal with mean 0 and variance $(m-1)$. Hence, Prob($\overline{EU}_{j,h}$) is

$$Q\left(\sqrt{\frac{n}{(m-1)}}\right)$$

for $h = 1, \ldots, m$, and $j = 1, \ldots, p$. □

Lemma 2.2 *The probability Prob($\overline{EV}_{i,h}$) is*

$$Q\left(\sqrt{\frac{p}{(m-1)}}\right)$$

for $i = 1, \ldots, n$ and $h = 1, \ldots, m$.

Proof of Lemma 2.2 : similar to the proof of Lemma 2.1. □

For the stronger concept of the memory capacity, we denote the probability that all (X_h, Y_h) are stored as fixed points as

$$\begin{aligned} P_* &= \text{Prob}\left(EU_{11} \cap \cdots \cap EU_{p,m} \cap EV_{11} \cap \cdots \cap EV_{n,m}\right) \\ &= 1 - \text{Prob}\left(\overline{EU}_{11} \cup \cdots \cup \overline{EU}_{p,m} \cup \overline{EV}_{11} \cup \cdots \cup \overline{EV}_{n,m}\right) \\ &\geq 1 - mp\text{Prob}\left(\overline{EU}_{1,1}\right) - mn\text{Prob}\left(\overline{EV}_{1,1}\right) \\ &= 1 - P_A - P_B \ , \end{aligned} \tag{2.15}$$

where $P_A = mpQ\left(\sqrt{\frac{n}{(m-1)}}\right)$ and $P_B = mnQ\left(\sqrt{\frac{p}{(m-1)}}\right)$. Note that

$$P_* \neq \left(1 - \text{Prob}(\overline{EU}_{1,1})\right)^{mp} \left(1 - \text{Prob}(\overline{EV}_{1,1})\right)^{mn} \tag{2.16}$$

because the events $EV_{i,h}$'s and $EU_{j,h}$'s are not mutually independent. This can be easily observed when $m = 2$.

For large value of z [77],

$$Q(z) \approx \exp\left\{-\frac{z^2}{2} - \log z - \frac{1}{2}\log 2\pi\right\} \tag{2.17}$$

which is quite accurate for $z > 3$. Using the above approximation of $Q(z)$,

$$\begin{aligned} P_A &= \exp\left\{\log m + \log p - \frac{n}{2(m-1)} - \frac{1}{2}\log\frac{n}{(m-1)} - \frac{1}{2}\log 2\pi\right\} \\ &\leq \exp\left\{\log m + \log p - \frac{n}{2m} - \frac{1}{2}\log\frac{n}{m} - \frac{1}{2}\log 2\pi\right\} \ . \end{aligned} \tag{2.18}$$

Substituting $p = rn$ and rearranging P_A,

$$P_A \leq \exp\left\{2\log n - \frac{n}{2m} - \frac{3}{2}\log\frac{n}{m} + \log r - \frac{1}{2}\log 2\pi\right\} \ . \tag{2.19}$$

Considering the first two terms, let $m = \frac{n}{4\log n}$. Then,

$$P_A \leq \exp\left\{-\frac{3}{2}\log(4\log n) + \log r - \frac{1}{2}\log 2\pi\right\}. \tag{2.20}$$

Moreover, as $n \to \infty$, then $P_A \to 0$. Since P_A is an increasing function of m, we can conclude that as $n \to \infty$ and $m \leq \frac{n}{4\log n}$, $P_A \to 0$. Using similar method, we can also show that as $p \to \infty$ and $m \leq \frac{p}{4\log p}$, $P_B \to 0$.

To sum up, for large n and p (i.e. $n \to \infty$, $p \to \infty$), if $m \leq \frac{\min(n,p)}{4\log\min(n,p)}$, then $P_* \to 1$. In the case of stronger concept, $\frac{\min(n,p)}{4\log\min(n,p)}$ can be considered a lower bound of the memory capacity of BAM

For the weaker concept, we should consider the probability P_w that a library pair is stored as a fixed point,

$$\begin{aligned} P_w &= \text{Prob}\left(EU_{11} \cap \cdots \cap EU_{p,1} \cap EV_{11} \cap \cdots \cap EV_{n,1}\right) \\ &= 1 - \text{Prob}\left(\overline{EU}_{11} \cup \cdots \cup \overline{EU}_{p,1} \cup \overline{EV}_{11} \cup \cdots \cup \overline{EV}_{n,1}\right) \\ &\geq 1 - p\text{Prob}\left(\overline{EU}_{1,1}\right) - n\text{Prob}\left(\overline{EV}_{1,1}\right) \\ &= 1 - pQ\left(\sqrt{\frac{n}{(m-1)}}\right) - nQ\left(\sqrt{\frac{p}{(m-1)}}\right). \end{aligned} \tag{2.21}$$

Similarly, we can easily show that for large n and p, if $m \leq \frac{\min(n,p)}{2\log\min(n,p)}$, then $P_w \to 1$. Hence, $\frac{\min(n,p)}{2\log\min(n,p)}$ is a lower bound of the memory capacity in the weaker sense.

With more careful analysis (considering the term $\frac{3}{2}\log(4\log n)$ in (2.20)), one can get a better lower bound for the stronger concept:

$$\frac{\min(n,p)}{(4 - c\frac{\log\log\min(n,p)}{\log\min(n,p)})\log\min(n,p)} \tag{2.22}$$

where $0 < c < 3$. In the case of the weaker concept, the better lower bound is

$$\frac{\min(n,p)}{(2 - c\frac{\log\log\min(n,p)}{\log\min(n,p)})\log\min(n,p)} \tag{2.23}$$

where $0 < c < 1$.

These two lower bounds are tight. They can be verified by considering the expectation of the sum of the number of the events $\overline{EV}_{i,h}$'s and $\overline{EU}_{j,h}$'s. For all library pairs being stored as fixed points, if the expectation

$$mnQ\left(\sqrt{\frac{p}{(m-1)}}\right) + mpQ\left(\sqrt{\frac{n}{(m-1)}}\right) \tag{2.24}$$

does not tend to zero as $n \to \infty$, then the probability that all library pairs are stored as fixed points does not tend to one. Clearly, the expectation does not tend to zero if

$$m \geq \frac{\min(n,p)}{(4 - 3\frac{\log\log\min(n,p)}{\log\min(n,p)})\log\min(n,p)}.$$

For a library pair being stored as a fixed point, the expectation is

$$nQ\left(\sqrt{\frac{p}{(m-1)}}\right) + pQ\left(\sqrt{\frac{n}{(m-1)}}\right) \tag{2.25}$$

which does not tend to zero if

$$m \geq \frac{\min(n,p)}{(2 - \frac{\log\log\min(n,p)}{\log\min(n,p)})\log\min(n,p)}.$$

2.5 Error Correction Capability of BAM

So far, we have not yet mentioned the error correction capability of BAM (or the attraction basin of a library pair). In this section, we estimate the probability P_t that a library pair (X_h, Y_h) can be correctly recalled in two shots,[6] given a noisy input X_{noise} of X_h with ρn errors. To have $P_t \to 1$, as $n \to \infty$, the parameters n, m, p, and ρ should be related by

$$m \leq \min\left(\frac{(1-2\rho)^2 n}{2\log n}, \frac{p}{2\log p}\right). \tag{2.26}$$

This condition describes the error correction capability in terms of n, p, m, and ρ. By the definition of P_t,

$$\begin{aligned} P_t &= 1 - \mathrm{Prob}(EN_1 \cup EN_2) \\ &\geq 1 - \mathrm{Prob}(EN_1) - \mathrm{Prob}(EN_2), \end{aligned} \tag{2.27}$$

where EN_1 is the event that

$$\mathrm{sgn}(WX_{noise}) \neq Y_h$$

and EN_2 is the event that

$$\mathrm{sgn}(W^T Y_h) \neq X_h.$$

[6] "Two shots" means that the noisy input is first presented in F_X, the state of F_Y is then obtained. This state of F_Y is feedback to F_X, and finally a new state of F_X is obtained.

Using a similar method presented in Section 2.4, we can easily show that

$$\text{Prob}(EN_1) \leq pQ\left(\sqrt{\frac{(1-2\rho)^2 n}{m-1}}\right) \tag{2.28}$$

$$\text{and Prob}(EN_2) \leq nQ\left(\sqrt{\frac{p}{m-1}}\right). \tag{2.29}$$

Hence, if

$$m \leq \min\left(\frac{(1-2\rho)^2 n}{2\log n}, \frac{p}{2\log p}\right),$$

then $P_t \to 1$, as $n \to \infty$.

The inequality (2.26) only gives us a weak sense of the error correction capability. The reason is that the considerations here are: 'for a library pair', 'for a noisy input with ρn errors', and 'for two shots recall'. Note that the term 'for a noisy input with ρn errors' reflects the random case errors. In most applications, correcting random errors may be satisfactory. However, it is interesting to find out whether the library pairs can attract **all** the initial noisy input within a distance of ρn for some positive constant ρ. This leads to the consideration: *worst case errors* ('for all noisy initial input with ρn errors'). If we consider the worst case errors, then (2.26) becomes

$$m < -\rho\log\rho - (1-\rho)\log(1-\rho) \tag{2.30}$$

which is not an increasing function of n and p.

The requirement that all library pairs are stored as fixed points may be too tight. In the following two chapters, we will examine the memory capacity and the error correction capability when a small number of errors are allowed in the retrieval pairs. The considerations in the next chapter are: 'for every library pair' and 'for all noisy input with ρn errors'.

2.6 Chapter Summary

In this chapter, we have presented an overview of BAM's encoding algorithm, recall process, and stability. Its main features are, 1) its state always converges to one of the fixed points for any real connection matrix, and 2) it is able to achieve the simultaneous hetero- and autoassociative recollections.

Also, the memory capacity of BAM is reviewed. Under the stronger concept (each library pair is stored as fixed points), the memory capacity is

$$\frac{\min(n,p)}{4\log\min(n,p)}.$$

or

$$\frac{\min(n,p)}{(4-c\frac{\log\log\min(n,p)}{\log\min(n,p)})\log\min(n,p)}$$

where $0 < c < 3$. With the weaker concept (a library pair is stored as a fixed point), the memory capacity becomes

$$\frac{\min(n,p)}{2\log\min(n,p)}.$$

or

$$\frac{\min(n,p)}{(2-c\frac{\log\log\min(n,p)}{\log\min(n,p)})\log\min(n,p)}$$

where $0 < c < 1$.

For the error correction capability of BAM, an initial input of X_h with ρn errors can recall the whole pair (X_h, Y_h) within two recall steps if the inequality

$$m \leq \min\left(\frac{(1-2\rho)^2 n}{2\log n}, \frac{p}{2\log p}\right)$$

is satisfied.

Chapter 3

Memory Capacity and Statistical Dynamics of First Order BAM

In this chapter, we investigate the statistical properties of the first order BAM under the outer product rule when a small number of errors in the retrieval pairs are allowed. Firstly, we use the concept of energy barrier which was originally used to analyze the memory capacity of Hopfield Network [59] to estimate the memory capacity of BAM and the number of errors in the retrieval pairs.

However, the above approach can only tell us that there exist some stable noisy versions of library pair around each library pair but cannot tell us about the attraction basin of each library pair. Hence, we then present the statistical dynamics of the number of errors of the first order BAM. From the dynamics, we can estimate the attraction basin, the memory capacity, and the number of errors in the retrieval pairs.

3.1 Introduction

One of the research topics in the field of associative memories is to understand the recall process in statistical sense [64] when a small number of errors are allowed in the retrieval patterns/pairs. However, most of the existing results are concerned with the cases of the Hopfield network and higher order Hopfield network only [59]-[68].

When error is not allowed in the retrieval pairs, the theoretically statistical memory capacity of BAM [23, 24], as well as the empirical memory capacity of BAM [21, 45, 46], have been reported. The theoretical memory capacity of BAM is about

$$\frac{\min(n,p)}{4\log\min(n,p)}.$$

If the BAM is used to store sparse-format library pairs [24], the theoretical statistical memory capacity is

$$\frac{0.68\min(n,p)^2}{(\log_2\min(n,p)+4)^2}.$$

The deterministic memory capacity of BAM which depends on the correlation of the library pairs was investigated in [50]. The result is

$$m < \min\left(\frac{n+a}{2k+a}, \frac{p+b}{2k'+b}\right)$$

where a and b are, respectively, the maximum correlations among X_h's and Y_h's. Also, k and k' are the number of errors allowed in the initial input.

In [64, 65], based on the law of large number, Amari has studied the dynamic behavior of the first-order Hopfield network. The dynamic behavior of the first-order Hopfield network with hysteretic response (The nonzero diagonal terms in the connection matrix are introduced.) was investigated in [67]. The extension of the results to the BAM was presented in [66]. The considerations of the dynamics in [64]-[67] are 'for a library pattern' and 'for a error pattern'. In other words, the dynamics proposed by Amari is the average dynamics of the number of errors.

Under the considerations that "for every library pair" and "for every error pattern", the memory capacity of Hopfield network, as well as the general higher order Hopfield network, has been studied based on the concept of energy barrier (see Section 2 in [59]) when error in the retrieval pairs is allowed. In the formulation of [59], the memory capacity and the number of errors in the retrieval pairs can be numerically estimated. The results rely on the existence of a suitable energy function (which always decreases during recall) for the general higher order Hopfield network.

The statistical convergence results of the first order Hopfield network have been studied in [60]. That is, given the number of errors in the present state, what the confidence interval of the number of errors is in the next state. In [60] the main concern is the order of the convergence rather than the actual dynamics (see the Main Lemma in [60]). However, if we directly follow the formulation used in [60] to handle BAM, we will get a relative poorer estimation on the memory capacity and the attraction basin (That will be discussed more details in Section 3.4.). Moreover, the formulation in [60] is mainly concerned with the first order Hopfield network only. Hence, generalization to the higher order BAM from this formulation may not be suitable.

This chapter addresses the statistical properties of BAM under the consideration: 'for every library pair' and 'for every error pattern (the worst case errors)'. We first estimate the memory capacity and the number of errors in the library pairs based on the concept of energy barrier [59]. Since the attraction basin cannot be estimated from the approach of energy barrier, we then develop the statistical dynamics of BAM: Starting with an initial state close to the library pairs (there are some errors in the initial state), how the confidence interval of the number of errors changes during recall.

Instead of using the formulation originated for the Hopfield network [60], we will develop another formulation to estimate the confidence interval of the number of errors in the next state for the BAM. In our formulation, we directly estimate the probability that the number of errors in the next state is less than $\rho^{new}n$. The minimum value of $\rho^{new}n$ such that the probability tends to one defines the confidence interval of the number of errors in the next state. Therefore, we can obtain a sequence of the confidence intervals. If the sequence converges to a very small value (It means that for each initial noisy version input with a given noisy level, the final noisy level in the retrieval pair is less than this small value with probability one.), then each library pair can be recalled with small number of errors. The limit value of the sequence represents the upper bound of the number of errors in the retrieval pairs. The maximum number of errors in the initial state, such that the sequence converges to a small value, represents the lower bound of the attraction basin. Also, the maximum number of the library pairs, such that the confidence interval of the number of errors converges to a small value, represents the lower bound of the memory capacity.

Section 3.2 presents the theoretical development of the statistical properties from the approach of energy barrier and Section 3.3 shows the corresponding numerical results. The statistical dynamics is presented in Section 3.4. The corresponding numerical examples are shown in Section 3.5. Lastly, a concluding remark is made in Section 3.6.

3.2 Existence of Energy Barrier

The concept of barrier energy is usually used to explain the stability of a neural network model [2, 3]. An energy function is associated with the state of the network. Starting from some initial states, the state is changed (this lowers the energy) until a fixed point (i.e. local minimum) is reached. As shown in Figure 3.1, if there exists an energy barrier around a library pattern, then the noise versions of this library pattern can be stored as a fixed point or a limit cycle. In this approach, we first need to form a close boundary for each library item and then we determine whether the energy of each point in the boundary is larger than that of the library item. If it is true, then there exists an energy barrier for the library item.

By using the theory of large deviation, Newman explained the nature of the energy barriers in the Hopfield model [59]. That is, under what condition (i.e. how many library patterns) each library pattern contains energy barriers with high probability. The smallest distance from the energy barriers to the library pattern reflects the upper bound of the number of errors in the retrieval pattern.

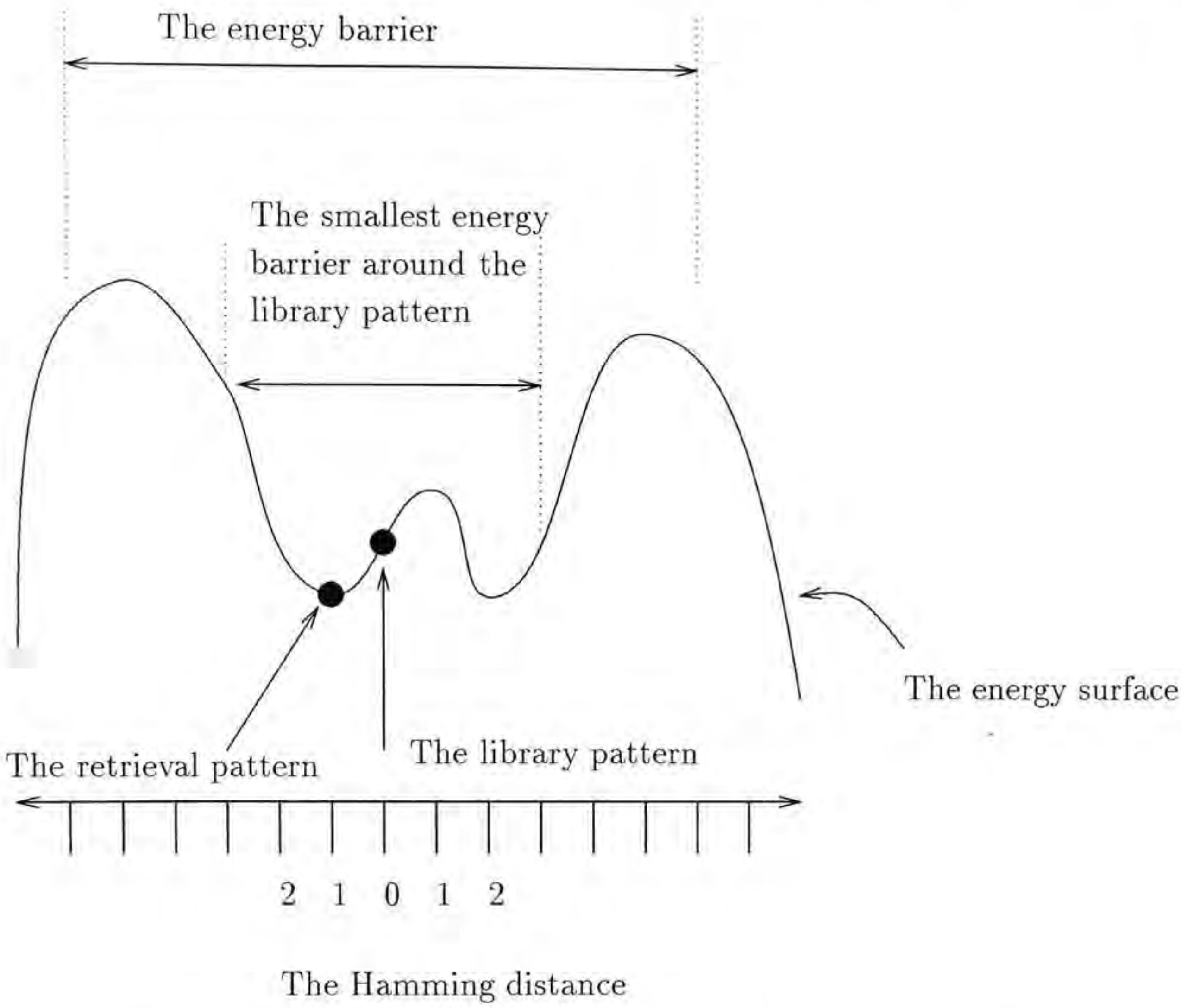


Figure 3.1 The illustration of the energy barrier. If the model is stable and the energy barriers exist, there is at least a stable noise version of the library pattern. The distance between the noise version and the library pattern is upper bounded by the energy barrier whose radius is the smallest.

The key point of this approach is that the change of the energy should be less than or equal to zero during recall (if the state of the model changes). For the 'less than zero' case, the state always converges to a fixed point. For the 'less than or equal to zero' case, the state may converge to fixed points or limit cycles. Therefore, finding out a suitable energy function in this approach is very important.

In this section, the memory capacity and the number of errors in the retrieval pairs of BAM are estimated based on Newman's approach.

Assumptions and Notations:

1. The dimensions (n and p) of BAM are large and $p = rn$, where r is a positive constant.
2. The number of library pairs is $m = \alpha n$, where α is a positive constant.
3. Each component of the library pairs (X_h, Y_h) is a ± 1 equiprobable independent random variable.
4. $d(X, X')$ is the Hamming distance between the two bipolar vectors X and X'.
5. Given the state of BAM (X, Y), $H(X, Y)$ is defined as the energy of this state

$$H(X,Y) = -Y^T W X = -Y^T \sum_{h=1}^{m} Y_h \, {X_h}^T X \,. \tag{3.1}$$

   Note that the energy of BAM always decreases during recall.

6. For the BAM, we consider the following close boundary:

$$S_{X_h,Y_h,\delta} = \left\{ (X', Y') : \max\left(\frac{d(X_h, X')}{n}, \frac{d(Y_h, Y')}{p} \right) = \delta \right\} .$$

7. Given δ, the sphere $S_{X_h,Y_h,\delta}$ is union of the sets

$$S_{X_h,Y_h,\delta_x,\delta} = \{ (X', Y') : d(X_h, X') = \delta_x n \text{ and } d(Y_h, Y') = \delta p \ \}$$

   for all $\delta_x \leq \delta$, and the sets

$$S_{X_h,Y_h,\delta,\delta_y} = \{ (X', Y') : d(X_h, X') = \delta n \text{ and } d(Y_h, Y') = \delta_y p \ \}$$

   for all $\delta_y \leq \delta$.

   For small δ, the size of the sphere $S_{X_h,Y_h,\delta}$ is less than

$$\left\{ 1 + \frac{2\delta}{1 - 2\delta} \right\} \binom{n}{\delta n} \binom{p}{\delta p}$$

   because

$$\tbinom{n}{\delta n - k} \leq \left\{ \frac{\delta}{1 - \delta} \right\}^k \tbinom{n}{\delta n} \,.$$

8. Given a library pair (X_h, Y_h), a fixed δ and a $\delta_x \leq \delta$, $E_{h,\delta_x,\delta,\alpha,r,all}$ is the event that

$$H(X_h, Y_h) < H(X', Y') \text{ for all } (X', Y') \in S_{X_h,Y_h,\delta_x,\delta}$$

where $H(X,Y)$ is the energy of the state (X,Y). Also, $E_{h,\delta_x,\delta,\alpha,r,a}$ is the event that

$$H(X_h, Y_h) < H(X', Y') \text{ for a } (X', Y') \in S_{X_h,Y_h,\delta_x,\delta} \ .$$

9. Given a library pair (X_h, Y_h), a fixed δ and a fixed $\delta_y \leq \delta$, $E_{h,\delta,\delta_y,\alpha,r,all}$ is the event that

$$H(X_h, Y_h) < H(X', Y') \text{ for all } (X', Y') \in S_{X_h,Y_h,\delta,\delta_y} \ .$$

Also, $E_{h,\delta,\delta_y,\alpha,r,a}$ is the event that

$$H(X_h, Y_h) < H(X', Y') \text{ for a } (X', Y') \in S_{X_h,Y_h,\delta_x,\delta} \ .$$

10. Given δ and α, $E_{\delta,\alpha,r}$ is the event that

$$E_{\delta,\alpha,r} = \bigcap_{h=1}^{m} \left(\bigcap_{\delta_x \leq \delta} E_{h,\delta_x,\delta,\alpha,r,all} \right) \left(\bigcap_{\delta_y \leq \delta} E_{h,\delta,\delta_y,\alpha,r,all} \right) \ .$$

It is the event that for each library pair, the energy of each boundary's point in $S_{X_h,Y_h,\delta}$ is larger than that of the library pair.

If $\text{Prob}(E_{\delta,\alpha,r})$ tends to one as $n \to \infty$, then for each library pair there exist some stable states (noisy versions of the library pair) around it. That means each library pair can be recalled with a certain noisy level.

We first estimate a lower bound of $\text{Prob}(E_{\delta,\alpha,r})$. Then we can find out a maximum value of α (denoted as $\alpha_{r,max}$) such that the lower bound of the probability tends to one as $n \to \infty$. Therefore, $\alpha_{r,max}n$ can be considered as a lower bound of the memory capacity of BAM when a small number of errors in the retrieval pairs are allowed. With a given α, if we can find out the minimum value of δ (denoted as $\delta_{\alpha,r,min}$) such that the lower bound of $\text{Prob}(E_{\delta,\alpha,r})$ tends to one as $n \to \infty$, then $\delta_{\alpha,r,min}n$ and $\delta_{\alpha,r,min}p$ can be considered as the upper bound of the number of errors in the retrieval pairs.

Lemma 3.1 *Chebyshev's inequality : For any random variable* χ *and* $u \geq 0$,

$$Prob(\chi \geq u) \leq \inf_{\tau \geq 0} e^{-\tau u} E(e^{\tau \chi}) .$$

Lemma 3.2 *Let* $v_1, v_2, \ldots, v_n$ *be independent* ± 1 *equiprobable random variables. Also,* $S_I = \sum_{i \in I} v_i$, *where* I *and* I' *are disjoint subsets of indices* $\{1, 2, \ldots, n\}$. *Then,*

$$E\left(\exp\left\{\tau \frac{S_I S_{I'}}{\sqrt{|\, I \,||\, I' \,|}}\right\}\right) \leq E\left(\exp\left\{\tau N_I N_{I'}\right\}\right) = \frac{1}{\sqrt{1-\tau^2}} ,$$

for $-1 < \tau < 1$, *where* N_I *and* $N_{I'}$ *are standard normal variables. In general,*

$$E\left(\exp\left\{\sqrt{a}\tau \frac{S_I S_{I'}}{\sqrt{|\, I \,||\, I' \,|}}\right\}\right) \leq \frac{1}{\sqrt{1-a\tau^2}} ,$$

for $\frac{-1}{\sqrt{a}} < \tau < \frac{1}{\sqrt{a}}$.

Proof of Lemma 3.2

The lemma follows from expanding each exponential as a power series in τ together with the fact that

$$E\left(\left(\frac{S_I}{\sqrt{|\, I \,|}}\right)^k\right) \leq E\left(N_I^k\right)$$

for all $k = 0, 1, \cdots,$. $\Box$

Lemma 3.3 *Stirling's asymptotic formula for factorial: let $\delta \in (0, 0.5)$ and n is a large integer, then*

$$\binom{n}{\delta n} \sim \exp\{n\hbar(\delta)\},$$

where

$$\hbar(\delta) = -\delta\log\delta - (1-\delta)\log(1-\delta).$$

Theorem 3.1 *For large n and p,*

$$\begin{aligned}\mathrm{Prob}(E_{\delta,\alpha,r}) \;\geq\; & 1 - \exp\Bigg\{(1+r)n\hbar(\delta) + \log\alpha n + \log\frac{1}{1-2\delta} - (m-1)\left(\sqrt{1+\frac{r\delta(1-\delta)n^2}{(m-1)^2}}\right.\\ & \left.-1 + \log\left(\sqrt{1+\frac{r\delta(1-\delta)n^2}{(m-1)^2}}-1\right) + \log\frac{r\delta(1-\delta)n^2}{2(m-1)^2}\right)\Bigg\}.\end{aligned}$$

Proof of Theorem 3.1

By the definition of $E_{\delta,\alpha,r}$,

$$\mathrm{Prob}(E_{\delta,\alpha,r}) \geq 1 - m\frac{1}{1-2\delta}\binom{n}{\delta n}\binom{p}{\delta p}P_u, \tag{3.2}$$

where P_u is an upper bound of the probability that the energy of a point in the sphere is less than that of the library pair. This is,

$$P_u > \mathrm{Prob}(\overline{E}_{h,\delta_x,\delta,\alpha,r,a})$$

and

$$P_u > \mathrm{Prob}(\overline{E}_{h,\delta,\delta_y,\alpha,r,a})$$

where $\overline{E}_{h,\delta_x,\delta,\alpha,r,a}$ and $\overline{E}_{h,\delta,\delta_y,\alpha,r,a}$ are the complement events of $E_{h,\delta_x,\delta,\alpha,r,a}$ and $E_{h,\delta,\delta_y,\alpha,r,a}$, respectively.

Clearly,

$$\text{Prob}(\overline{E}_{h,\delta_x,\delta,\alpha,r,a}) = \text{Prob}(\overline{E}_{h,\delta,\delta_y,\alpha,r,a}),$$

for $\delta_y = \delta_x$. Also, due to the homogeneous structure of BAM, we only need to find the upper of $\text{Prob}(\overline{E}_{1,\delta_x,\delta,\alpha,r,a})$.

We let I be the set of indices in which X and X_1 differ. Also, we let J be the set of indices in which Y and Y_1 differ. Without loss of generality, we consider that the library pair (X_1, Y_1) has all components positive: $X_1 = (1, \cdots, 1)^T$ and $Y_1 = (1, \cdots, 1)^T$. Then,

$$\text{Prob}(\overline{E}_{1,\delta_x,\delta,\alpha,r,a}) =$$

$$\text{Prob}\left\{-2\sum_{h\neq 1}^{m}\sum_{i\in I_{\delta_x}}\sum_{j\notin J_\delta} x_{ih}y_{jh} \;+\; 2\sum_{h\neq 1}^{m}\sum_{i\notin I_{\delta_x}}\sum_{j\in J_\delta} x_{ih}y_{jh} \;\geq\; 2\delta_x(1-\delta)rn^2 + 2\delta(1-\delta_x)rn^2\right\} <$$

$$\text{Prob}\left\{-\sum_{h\neq 1}^{m}\sum_{i\in I_{\delta_x}}\sum_{j\notin J_\delta} x_{ih}y_{jh} \;+\; \sum_{h\neq 1}^{m}\sum_{i\notin I_{\delta_x}}\sum_{j\in J_\delta} x_{ih}y_{jh} \;\geq\; \delta(1-\delta_x)rn^2\right\} <$$

$$\text{Prob}\left\{\frac{\sum_{h\neq 1}^{m} U^h_{I_{\delta_x}} V^h_{\overline{J}_\delta} \;+\; U^h_{\overline{I}_{\delta_x}} V^h_{J_\delta}}{\sqrt{\delta(1-\delta_x)rn^2}} \;\geq\; \sqrt{\delta(1-\delta_x)rn^2}\right\} <$$

$$\text{Prob}\left\{\frac{\sum_{h\neq 1}^{m} U^h_{I_{\delta_x}} V^h_{\overline{J}_\delta} \;+\; U^h_{\overline{I}_{\delta_x}} V^h_{J_\delta}}{\sqrt{\delta(1-\delta_x)rn^2}} \;\geq\; \sqrt{\delta(1-\delta)rn^2}\right\} =$$

$$P_u\,.$$

where

$$\begin{aligned}
U^h_{I_{\delta_x}} &= -\sum_{i\in I_{\delta_x}} x_{ih} \\
U^h_{\overline{I}_{\delta_x}} &= \sum_{i\notin I_{\delta_x}} x_{ih} \\
V^h_{J_\delta} &= \sum_{j\in J_\delta} y_{jh} \\
V^h_{\overline{J}_\delta} &= \sum_{j\notin J_\delta} y_{jh}\,.
\end{aligned}$$

With Lemma 3.1 and Lemma 3.2,

$$P_u \leq \inf_{1>\tau\geq 0}\left(e^{-u'\tau}\frac{1}{\sqrt{1-\tau^2}}\right)^{(m-1)}\left(e^{-u'\tau}\frac{1}{\sqrt{1-a\tau^2}}\right)^{(m-1)},$$

where

$$a = \frac{\delta_x(1-\delta)}{\delta(1-\delta_x)} < 1$$

and

$$u' = \frac{\sqrt{\delta(1-\delta)rn}}{2(m-1)}.$$

Hence,

$$P_u \leq \inf_{1>\tau\geq 0} \left(e^{-u'\tau} \frac{1}{\sqrt{1-\tau^2}} \right)^{2(m-1)} . \tag{3.3}$$

When τ is equal to

$$\tau = \frac{\sqrt{1+4u'^2}-1}{2u'},$$

(3.3) becomes

$$\begin{aligned} \mathrm{Prob}(\overline{E}_{1,\delta,\alpha,r}) \; &\leq \; \binom{n}{\delta n} \binom{p}{\delta p} \exp\Big\{-(m-1)\Big(\sqrt{1+4u'^2}-1 \\ &\quad + \log \frac{\sqrt{1+4u'^2}-1}{2u'^2}\Big)\Big\} . \end{aligned} \tag{3.4}$$

With Lemma 3.3, Theorem 3.1 can be obtained. □

From Theorem 3.1, if

$$\begin{aligned} &(1+r)\hbar(\delta) - \alpha \left(\sqrt{1+\frac{r\delta(1-\delta)}{\alpha^2}} - 1 + \right. \\ &\left. \log \left(\sqrt{1+\frac{r\delta(1-\delta)}{\alpha^2}} - 1 \right) - \log \frac{r\delta(1-\delta)}{2\alpha^2} \right) < 0 \ , \end{aligned} \tag{3.5}$$

then $\mathrm{Prob}(E_{\delta,\alpha,r})$ tends to one as $n \to \infty$. For small[1] δ,

$$\begin{aligned} &\alpha \left(\sqrt{1+\frac{r\delta(1-\delta)}{\alpha^2}} - 1 + \log \left(\sqrt{1+\frac{r\delta(1-\delta)}{\alpha^2}} - 1 \right) - \log \frac{r\delta(1-\delta)}{2\alpha^2} \right) \\ &\to \; \frac{r\delta(1-\delta)}{4\alpha}. \end{aligned} \tag{3.6}$$

Also,

$$\frac{\frac{r\delta(1-\delta)}{\alpha}}{4(1+r)\hbar(\delta)} \to \infty \text{ as } \alpha \to 0. \tag{3.7}$$

Hence, the following corollary is obtained. It means that for a sufficient small α, the probability that there exist energy barriers for each library pair tends to one.

[1] See the proof of Corollary 3.2. We can choose $\delta = O(\alpha^3)$

Corollary 3.1 *For a small δ, there exist some small α such that (3.5) holds.*

Given a fixed r, define

$$\begin{aligned}\Omega(\alpha,\delta) &= (1+r)\hbar(\delta) - \alpha\left(\sqrt{1+\frac{r\delta(1-\delta)}{\alpha^2}} - 1\right.\\ &\left.\quad + \log\left(\sqrt{1+\frac{r\delta(1-\delta)}{\alpha^2}} - 1\right) - \log\frac{r\delta(1-\delta)}{2\alpha^2}\right). \qquad (3.8)\end{aligned}$$

Hence, the lower bound of the memory capacity is $\alpha_{r,max}n$, where $\alpha_{r,max}$ is **the maximum value of α such that $\Omega < 0$ (for some $\delta < 0.5$)** Based on (3.8), $\alpha_{r,max}$ can be numerically solved. A typical plot of $\Omega(\alpha,\delta)$ is shown in Figure 3.2. Intuitively, $\Omega(\alpha,\delta)$ should be an increasing function of α. With careful analysis (by using binomial expansion on the square root), the following additional feature of $\Omega(\alpha,\delta)$ can be obtained.

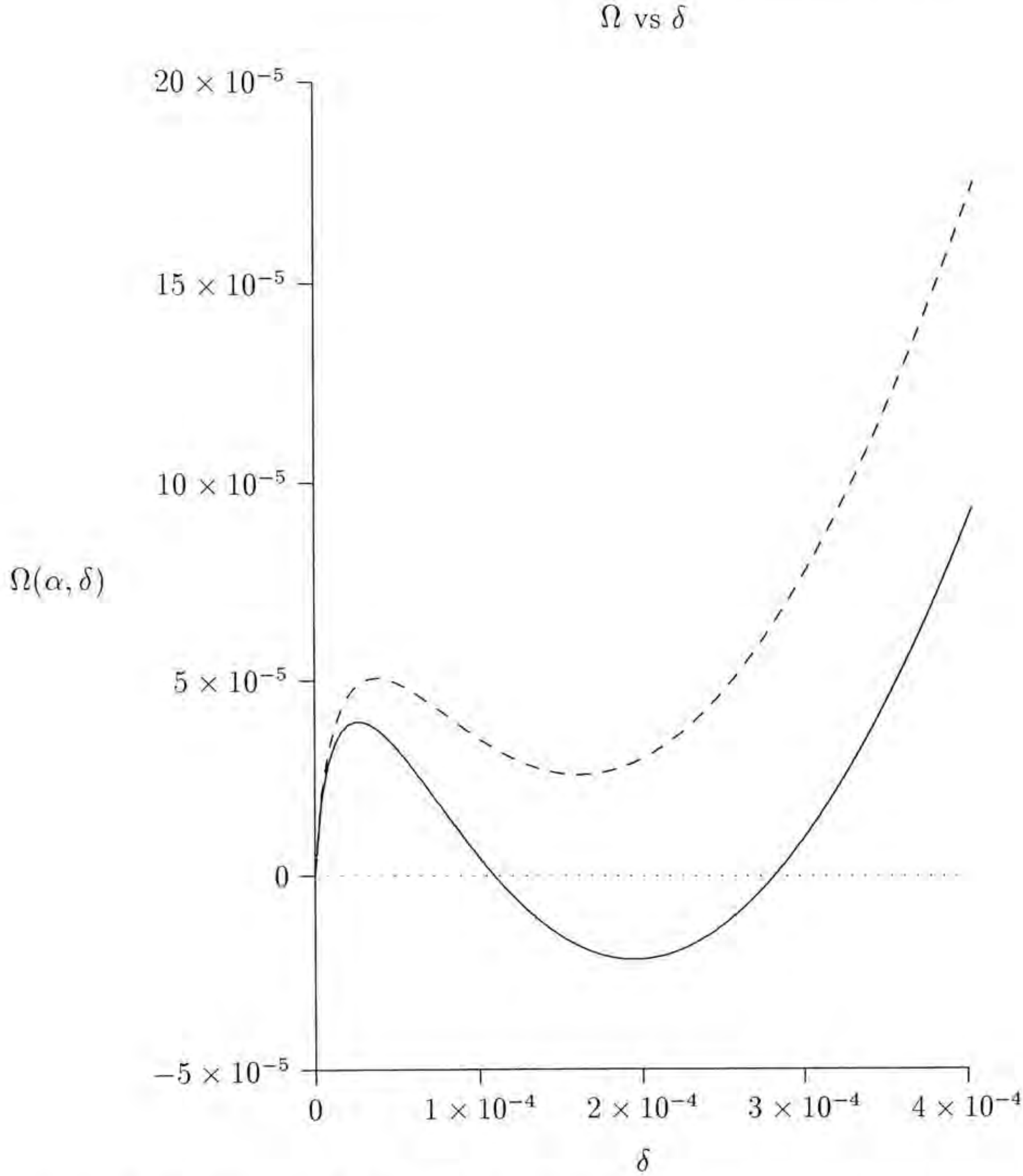


Figure 3.2 A typical Plot of Ω vs δ, where $r = 1$, $\alpha = 0.0113$ for solid-line and $\alpha = 0.0115$ for dashes-line.

Corollary 3.2 *With a small $\delta << \alpha^2$, $\Omega(\alpha,\delta)$ is an increasing function of α.*

Proof of Corollary 3.2

Using binomial theorem, if δ is small, then

$$\Omega(\alpha,\delta) \approx (1+r)\hbar(\delta) - \alpha\left(\frac{v''}{2} + \log(1-\frac{v''}{4})\right)$$

where

$$v'' = \frac{r\delta(1-\delta)}{\alpha^2}\,.$$

Since $\log(1-x) \approx -x$ for small positive x,

$$\Omega(\alpha,\delta) \approx (1+r)\hbar(\delta) - \frac{r\delta(1-\delta)}{4\alpha}\,.$$

Hence, $\Omega(\alpha,\delta)$ is an increasing function of α for a small δ. □

Remark: In fact, it is not difficult to show that $\Omega(\alpha,\delta)$ is an increasing function of α because

$$f_1(u) = \frac{1}{u}\left(\sqrt{1+4u^2} - 1 + \log\frac{\sqrt{1+4u^2}-1}{2u^2}\right)$$

is an increasing function.

3.3 Memory Capacity from Energy Barrier

According to Corollary 3.2 (also see Corollary 3.3 shown below), if there exist some small δ's such that $\Omega < 0$ with α', then $\Omega < 0$ for all $\alpha'' < \alpha'$. Hence, we can say that $\alpha_{r,max}n$ is a lower bound of the memory capacity. Table 3.1 summarizes the values of $\alpha_{r,max}$ that are numerically solved. From the table, when we first increase r, $\alpha_{r,max}$ also increases. But $\alpha_{r,max}$ will decrease after a certain value of r is reached ($r > 20$). Also, there is a symmetry property between $\alpha_{r,max}$ and $\alpha_{\frac{1}{r},max}$. That is,

$$\alpha_{\frac{1}{r},max}r \approx \alpha_{r,max}\,.$$

This symmetry property can be verified by interchanging r to $\frac{1}{r}$ and α to $\frac{\alpha}{r}$ in (3.8). This property means that interchanging p and n does not affect the overall estimated lower bound of the memory capacity.

Given fixed α and r, the minimum value of δ, such that (3.8) (i.e. $\Omega(\alpha, \delta) < 0$) holds, is denoted as $\delta_{\alpha,r,min}$. Hence, $\delta_{\alpha,r,min}n$ and $\delta_{\alpha,r,min}p$ can be regarded as **the upper bound of the number of errors in the the retrieval pair**. Figure 3.3 summarizes $\delta_{\alpha,r,min}$ which is numerically solved based on (3.8). From Figure 3.3, $\delta_{\alpha,r,min}$ decreases exponentially as α decreases. This can be mathematically verified by the following corollary.

Corollary 3.3 *For a small* $\delta_{\alpha,r,min} << \alpha^2$,

$$\delta_{\alpha,r,min} \approx \exp(-\frac{r}{4(1+r)\alpha} + 1) .$$

Proof of Corollary 3.3

When δ is small, Ω can be approximated by

$$\Omega \approx (1+r)\hbar(\delta) - \frac{r\delta(1-\delta)}{4\alpha} .$$

Hence, $\delta_{\alpha,r,min}$ is the solution of the following equation

$$(1+r)\hbar(\delta) - \frac{r\delta(1-\delta)}{4\alpha} \approx 0 . \tag{3.9}$$

As δ is small,

$$\hbar(\delta) \approx -\delta \log \delta + (1-\delta)\delta \approx -\delta \log \delta + \delta$$

and

$$\frac{r\delta(1-\delta)}{4\alpha} \approx \frac{r\delta}{4\alpha} .$$

Then (3.9) becomes

$$\log \delta \approx -\frac{r}{4(1+r)\alpha} + 1 .$$

Immediately, $\delta_{\alpha,r,min} \approx \exp(-\frac{r}{4(1+r)\alpha}+1)$. □

As $\delta_{\alpha,r,min}$ is very similar to $\delta_{\frac{\alpha}{r},\frac{1}{r},min}$ (with a suitable change on *alpha*), we only show the cases of $r = 1, 2, 5, 10$ in Figure 3.3. This can also be mathematically verified by Corollary 3.3. If we change r to $\frac{1}{r}$ and α to $\frac{\alpha}{r}$ in Corollary 3.3, a new corollary is immediately obtained.

Corollary 3.4 *For a small* $\delta_{\frac{\alpha}{r},\frac{1}{r},min} << \alpha^2$,

$$\delta_{\frac{\alpha}{r},\frac{1}{r},min} \approx \exp(-\frac{r}{4(1+r)\alpha}+1) .$$

The above corollary means that interchanging p and n does not affect the estimated $\delta_{\frac{\alpha}{r},\frac{1}{r},min}$.

Table 3.1 The lower bound of memory capacity of BAM.

r	$\alpha_{r,max}$
50	0.0169
20	0.0178
10	0.0181
5	0.0175
2	0.0149
1	0.0113
0.5	0.00746
0.2	0.00352
0.1	0.00181
0.05	0.000875
0.02	0.000336

Note that the maximum value of δ (denoted as $\delta_{\alpha,r,max}$) such that Ω is less zero does not reflect the attraction basin (see the energy barrier whose radius is larger in Figure 3.1). It is because the following inequality has not been proved yet:

$$H(X', Y') < H(X'', Y''), \tag{3.10}$$

for all $(X', Y') \in S_{X_h, Y_h, \delta'}$ and for all $(X'', Y'') \in S_{X_h, Y_h, \delta''}$ where $\delta_{\alpha,r,min} < \delta' < \delta'' < \delta_{\alpha,r,max}$. One might think that if (3.10) holds, then a bipolar library, with errors less than $\delta_{\alpha,r,max}n$, can correctly recall the desired pair with high probability. In fact, even if we can prove that (3.10) is true, we still cannot claim that the lower bound of the attraction basin is $\delta_{\alpha,r,max}n$. It is because the BAM is a heteroassociative memory and the initial state of one of layers is unknown.

Based on the method originally applied to Hopfield network [59], we cannot determine the attraction basin for the BAM. It is because Theorem 3.1 only tells us that for a certain number of library pairs (αn), there exists energy barriers for each library pair. However, the attraction basin for the worst case errors has not been mentioned yet.

In the next section, we will present the statistical dynamics of the number of errors for the BAM. From the statistical dynamics, we can estimate the memory capacity when a small number of errors in the retrieval pairs are allowed. Moreover, the attraction basin and the number of errors in the retrieval pairs can also be estimated.

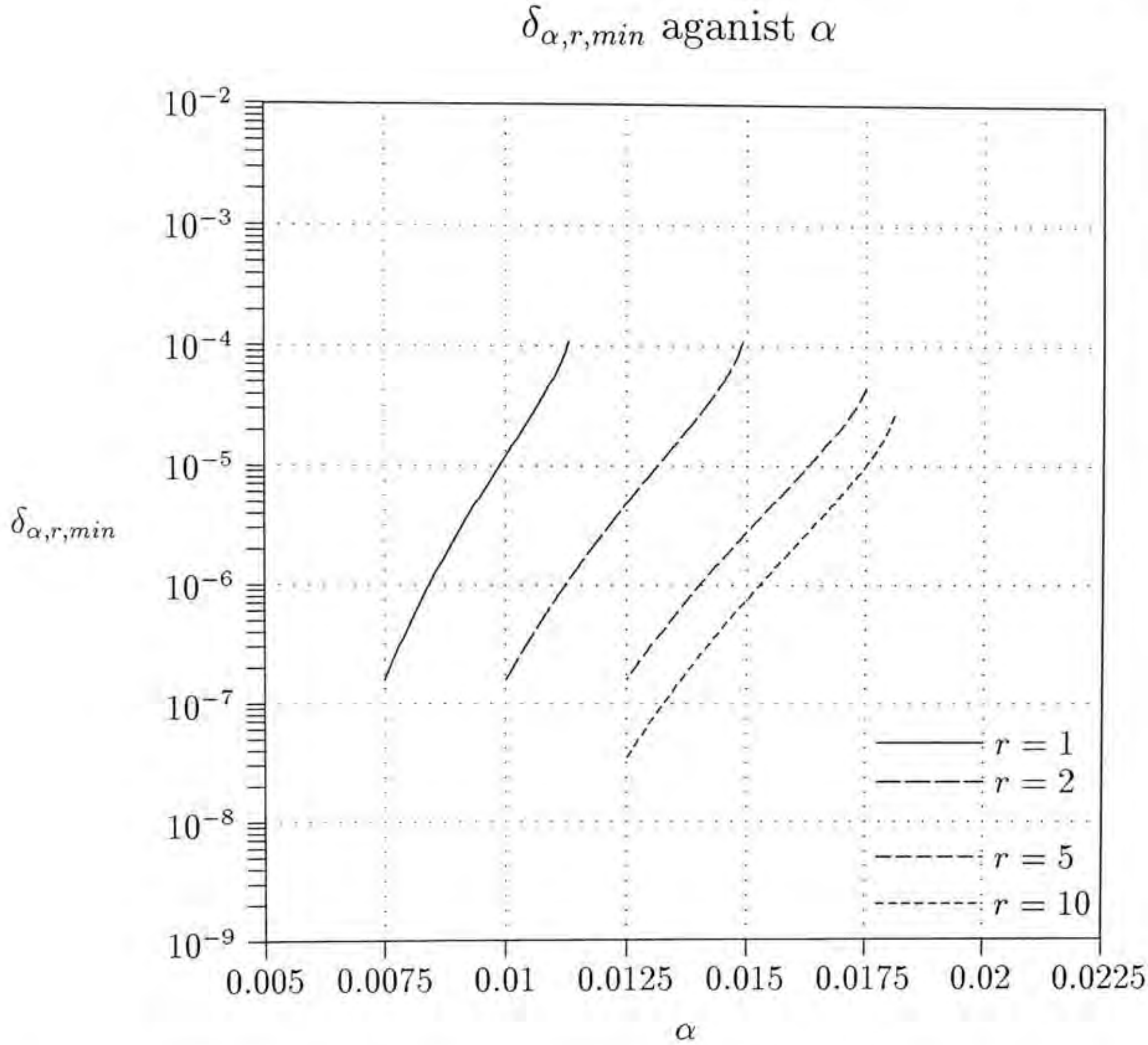


Figure 3.3 The upper bound of the number of errors in the retrieval pairs, $\delta_{\alpha,r,min}$, on the basis of the concept of energy barrier.

3.4 Confidence Dynamics

In this section, we first present the statistical dynamics of the number of errors of BAM. Then, we discuss the way to estimate the memory capacity, the attraction basin, and the number of errors in the retrieval pairs from the statistical dynamics. Before we introduce our formulation, we will briefly review the formulation used in the Hopfield network [60].

In [60], let A be the event that (given the number of errors ρn in the present state and given the number of library pattern $m = \alpha n$) the number of errors in the next state is more than $\rho^{new} n$, where n is the dimension. Let $\overline{A}$ be the complement event of A. Furthermore, Komlos [60] proved that if

$$a + T_1 > 0 \tag{3.11}$$

then $\overline{A}$ holds, where $a = f_a(\rho, \rho^{new}, \alpha)$ is positive function, and T_1 is a random number. Also,

$$T_1 < -f_1(\rho, \rho^{new}, \alpha),$$

with probability $1 - e^{-\triangle}$, where f_1 is a positive function. Hence, if

$$a - f_1 > 0\,, \tag{3.12}$$

then the Prob(A) is upper bounded by $me^{-\triangle}$. Under this formulation, the minimum value of ρ^{new}, such that (3.12) holds, defines the confidence interval of the number of errors in the next state. Note that we will get a relative poorer estimation on the statistical properties of BAM (see the end of this section) if we follow the above formulation which is originated for Hopfield network of [60]. At the end of this section, we will discuss the results of BAM from this formulation.

Hence, we develop another formulation to estimate the number of errors in the next state for the BAM. In our formulation, we first define two probabilities P_Y^{**} and P_X^{**} for the BAM.

Definition 3.1 *Given that $p = rn$ and $m = \alpha n$, P_Y^{**} is the probability that the number of errors in the layer F_Y in the next state is less than $\rho_y p$ (i.e. the Hamming distance between Y_h and $Y^{(t)}$ is less than $\rho_y p$), for every library pair (X_h, Y_h) and for any $\rho_x^{(t)} n$ errors in the layer F_X in the present state (i.e. the Hamming distance between X_h and $X^{(t)}$ is equal to $\rho_x^{(t)} n$).*

Definition 3.2 *Given that $p = rn$ and $m = \alpha n$, P_X^{**} is the probability that the number of errors in the layer F_X in the next state is less than $\rho_x n$, for every library pair (X_h, Y_h) and for any $\rho_y^{(t+1)} p$ errors in the layer F_Y in the present state.*

We will first estimate a lower bound of P_Y^{**}. Then, we can find the minimum value of ρ_y, denoted as ρ'_y, such that P_Y^{**} tends to one. The above means that given $\rho_x^{(t)}$, the probability that the number of errors in F_Y in the next state is less than $\rho'_y p$ tends to one. Also, we can find the minimum value of ρ_x, denoted as ρ'_x, such that P_X^{**} tends to one. From ρ'_x and ρ'_y, we can construct the dynamics about the confidence interval of the number of errors. The notations and assumptions used here are:

- The dimensions (n and p) are large and $p = rn$, where r is a positive constant.
- Each component of the library pairs (X_h, Y_h) is a ± 1 equiprobable independent random variable.
- $EA_{h,g}$ is the event

$$d(Y^{(t+1)}, Y_h) < \rho_y p$$

  for a given library pair (X_h, Y_h) and a given present state $X^{(t)}$ which is an element of the set

$$S_{h,t} = \left\{ X \in \{+1, -1\}^n \text{ such that } d(X, X_h) = \rho_x^{(t)} n \right\}.$$

Note that the number of elements in the set $S_{h,t}$ is $\binom{n}{\rho_x^{(t)}n}$. Thus the index g has the range from 1 to $\binom{n}{\rho_x^{(t)}n}$. Also, $\overline{EA_{h,g}}$ is the complement event of $EA_{h,g}$:

$$d(Y^{(t+1)}, Y_h) \geq \rho_y p \,.$$

- EA is the event that

$$d(Y^{(t+1)}, Y_h) < \rho_y p \,,$$

for every library pair (X_h, Y_h) and for every $X^{(t)} \in S_{h,t}$. Also, $\overline{EA}$ is the complement event of EA. Hence,

$$\overline{EA} = \bigcup_{h,g} \overline{EA_{h,g}} \,.$$

and

$$\begin{aligned} P_Y^{**} &\equiv \text{Prob}(EA) \\ &\geq 1 - m \binom{n}{\rho_x^{(t)}n} \text{Prob}(\overline{EA_{g,h}}) \,. \end{aligned}$$

Lemma 3.4 *For large n and p,*

$$Prob(\overline{EA_{g,h}}) \leq \exp\left\{ rn\hbar(\rho_y) - \frac{m-1}{2}\left(\sqrt{1+4u^2} - 1 + \log\frac{\sqrt{1+4u^2}-1}{2u^2}\right)\right\},$$

where

$$u = \frac{\sqrt{\rho_y r}(1 - 2\rho_x^{(t)})n}{m-1}$$

for $g = 1, \ldots, \binom{n}{\rho_x^{(t)}n}$ and $h = 1, \ldots, m$.

Proof of Lemma 3.4

Without loss of generality, we consider that the library pair (X_h, Y_h) is: $X_h = (1,\ldots,1)^T$ and $Y_h = (1,\ldots,1)^T$. Let I be the set of indices in which $X^{(t)}$ and X_h differ. For a given $X^{(t)} \in S_{h,t}$, there is only one I where $|I| = \rho_x^{(t)} n$. Also, let J be the set of indices of Y_h and $Y^{(t+1)}$ such that $|J| = \rho_y p$. Note that there are $\binom{p}{\rho_y p}$ such sets of J.

The event $\overline{EA_{g,h}}$ implies that there is at least one J, where $|J| = \rho_y p$, such that

$$\sum_{j\in J}\sum_{i=1}^{n} w_{ji}\, x_i^{(t)} < 0\,. \tag{3.13}$$

Hence,

$$\begin{aligned}
\mathrm{Prob}(\overline{EA_{g,h}}) \;&\le\; \mathrm{Prob}\Big(\text{ there is at least one } J \text{ where } |J| = \rho_y p \text{ such that} \\
&\qquad \sum_{j\in J}\sum_{i=1}^{n} w_{ji}\, x_i^{(t)} < 0\Big) \\
&\le\; \binom{p}{\rho_y p}\,\mathrm{Prob}\left(\sum_{j\in J}\sum_{i=1}^{n} w_{ji}\, x_i^{(t)} < 0 \text{ for a given } J\right).
\end{aligned} \tag{3.14}$$

Let

$$P'' = \mathrm{Prob}\left(\sum_{j\in J}\sum_{i=1}^{n} w_{ji}\, x_i^{(t)} < 0 \text{ for a given } J\right). \tag{3.15}$$

Replacing w_{ji} with $\sum_{h'=1}^{m} y_{jh} x_{ih}$,

$$P'' \;=\; \mathrm{Prob}\left(\rho_y p(1-2\rho_x^{(t)})n + \sum_{h'\neq h}^{m}\sum_{j\in J} y_{jh'}\left(\sum_{i\notin I}^{n} x_{ih'} - \sum_{i\in I} x_{ih'}\right) < 0\right). \tag{3.16}$$

Due to the symmetrical properties of random variables in (3.16),

$$P'' = \mathrm{Prob}\left(\sum_{h'\neq h}^{m}(\sum_{j\in J} y_{jh'})(\sum_{i\notin I} x_{ih'} - \sum_{i\in I} x_{ih'}) > \rho_y p(1-2\rho_x^{(t)})n\right). \tag{3.17}$$

Applying Lemma 3.1 and 3.2, (3.17) becomes

$$P'' \le \inf_{1>\tau>0}\left(e^{-\tau u}\frac{1}{\sqrt{1-\tau^2}}\right)^{m-1}. \tag{3.18}$$

Taking $\tau = \frac{1}{2u}(\sqrt{1+4u^2}-1)$, (3.18) becomes

$$P'' \le \exp\left\{-\frac{m-1}{2}\left(\sqrt{1+4u^2} - 1 + \log\frac{\sqrt{1+4u^2}-1}{2u^2}\right)\right\}. \tag{3.19}$$

Replacing the binomial coefficient in (3.14) and substituting (3.19) back into (3.14), Lemma 3.4 can be immediately obtained. □

With Lemma 3.4, we can immediately obtain the following theorem.

Theorem 3.2 *For large n and p,*

$$P_Y^{**} \geq 1 - \exp\left\{ n\hbar(\rho_x^{(t)}) + \log \alpha n + rn\hbar(\rho_y) - \frac{m-1}{2}\left(\sqrt{1 + \frac{4r\rho_y(1-2\rho_x^{(t)})^2 n^2}{(m-1)^2}} - 1 + \log \frac{\sqrt{1+\frac{4r\rho_y(1-2\rho_x^{(t)})^2n^2}{(m-1)^2}} - 1}{\frac{2\rho_y r(1-2\rho_x^{(t)})^2 n^2}{(m-1)^2}}\right)\right\},$$

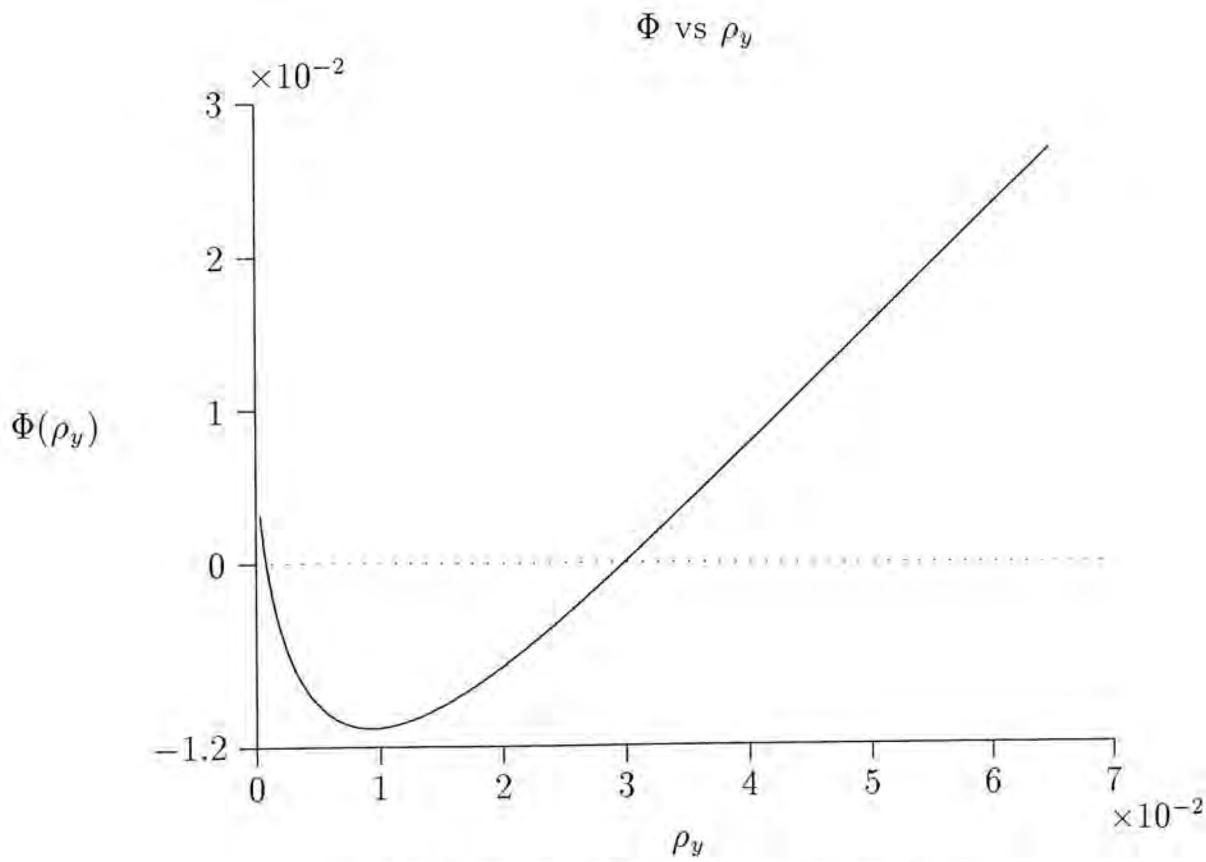


Figure 3.4 A Typical Plot of Φ.

From Theorem 3.2, if

$$\Phi(\rho_y) = \hbar(\rho_x^{(t)}) + r\hbar(\rho_y) - \frac{\alpha}{2}\left(\sqrt{1 + \frac{4r\rho_y(1-2\rho_x^{(t)})^2}{\alpha^2}} - 1\right.$$

$$+\log \frac{\sqrt{1+\frac{4r\rho_y(1-2\rho_x^{(t)})^2}{\alpha^2}}-1}{\frac{2\rho_y r(1-2\rho_x^{(t)})^2}{\alpha^2}}\Bigg) < 0\,, \tag{3.20}$$

then $P_Y^{**} \to 1$ as $n \to \infty$ and $p \to \infty$ (note $\frac{m-1}{n} \to \alpha$). Hence, the minimum value of ρ_y (denoted as ρ'_y), such that (3.20) holds, defines the confidence interval of the number of errors in F_Y in the next state. A typical plot of $\Phi(\rho_y)$ is shown in Figure 3.4. For a given $\rho_x^{(t)} \in [0, 0.5)$, ρ'_y can be solved numerically from (3.20). If ρ_y or ρ'_y is small ($<< \alpha^2$), $\Phi(\rho_y)$ can be approximated by

$$\Phi(\rho_y) \approx \Phi'(\rho_y) = \hbar(\rho_x^{(t)}) + r\hbar(\rho_y) - \frac{\alpha}{2}\left(\frac{1}{2}u'' + \log(1-\frac{u''}{4})\right) \tag{3.21}$$

where

$$u'' = \frac{4r\rho_y r(1-2\rho_x^{(t)})^2}{\alpha^2}\,.$$

Note that

$$\log(1-x) \approx -x\,,$$

when x is a small positive number. Hence, (3.21) becomes

$$\Phi'(\rho_y) \approx \hbar(\rho_x^{(t)}) + r\hbar(\rho_y) - \frac{\rho_y r(1-2\rho_x^{(t)})^2}{2\alpha}\,. \tag{3.22}$$

Numerically solving ρ'_y based on (3.22) is much easier than directly solving ρ'_y based on (3.20). Apparently, for fixed $\rho_x^{(t)}$, ρ'_y can be easily solved (see Figure 3.5). Let ρ_y^* be the intersection of the line

$$L_{11} \,:\, y = \frac{\rho_y(1-2\rho_x^{(t)})^2}{2\alpha} - \frac{\hbar(\rho_x^{(t)})}{r} \tag{3.23}$$

and the curve

$$C_{11} \,:\, y = \hbar(\rho_y)\,. \tag{3.24}$$

Then,

$$\rho'_y = \rho_y^* + \varepsilon$$

where ε is an arbitrarily small positive number. Note that $\hbar(\rho_x^{(t)})$ is an increasing function of $\rho_x^{(t)} \in [0, 5)$ and $(1-2\rho_x^{(t)})^2$ is a decreasing function of $\rho_x^{(t)} \in [0, 5)$. From (3.23) and Figure 3.5, for a smaller $\rho_x^{(t)} \in (0, 0.5)$ (the line is shifted up and the slope of the line increases.), a smaller ρ'_y can be obtained. Thus, the following corollary can be obtained.

Corollary 3.5 *If both ρ'_{y1} and ρ'_{y2} are small ($<< \alpha^2$), $\rho^{(t)}_{x1} < \rho^{(t)}_{x2} < 0.5$ implies that $\rho'_{y1} < \rho'_{y2}$.*

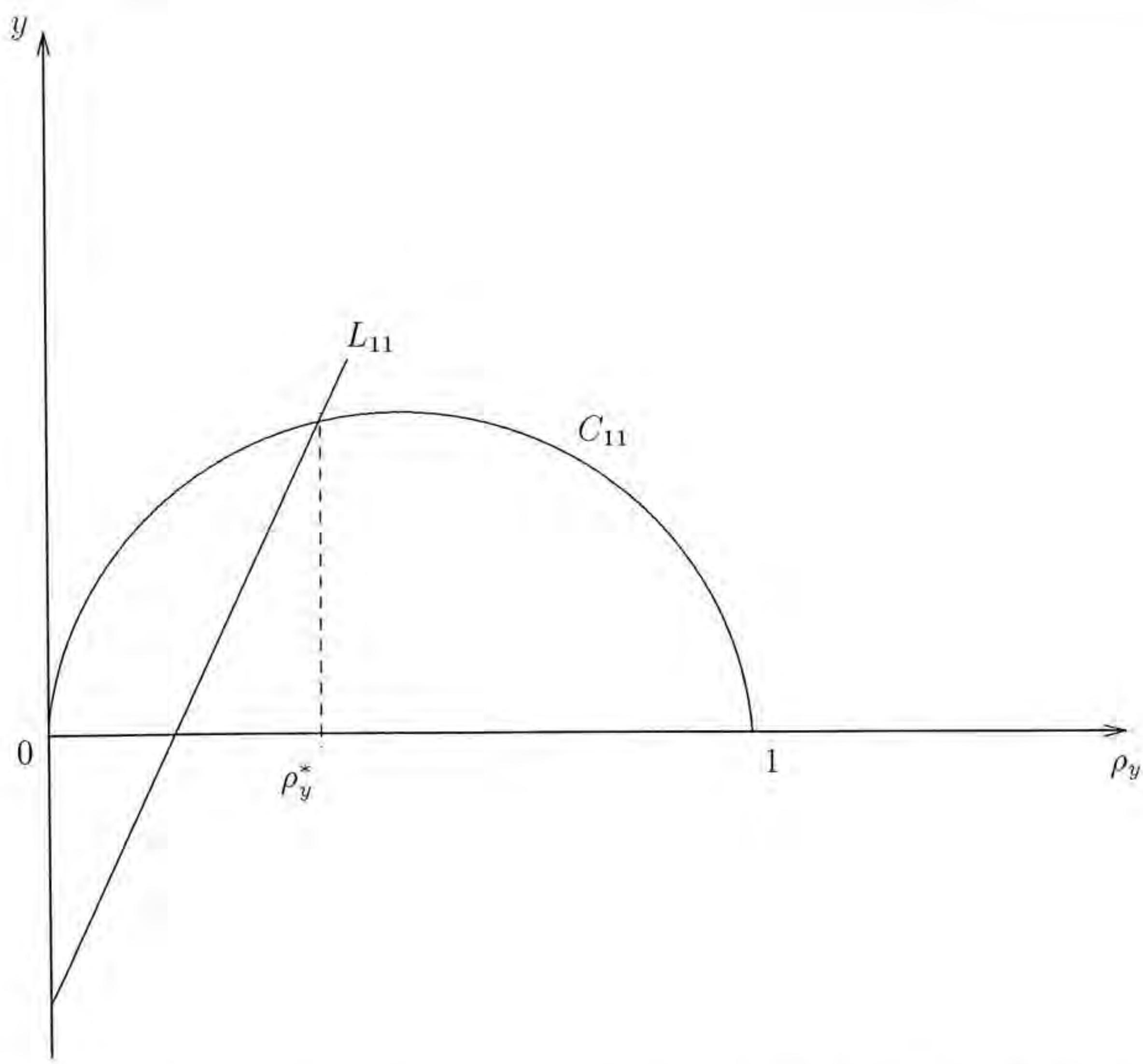


Figure 3.5 Graphical Implication of Solving ρ'_y. Note ρ'_y can be solved by finding out the intersection of C_{11} and L_{11}.

From Corollary 3.5, as $n \to \infty$ and $p \to \infty$, for every library pair (X_h, Y_h) and for every $X^{(t)}$ such that $d(X_h, X^{(t)}) \leq \rho^{(t)}_x n$ (i.e. the number of errors in the present state of F_X is less than or equal to $\rho^{(t)}_x n$), where $\rho^{(t)}_x \in [0, 5)$, the probability that the number of errors in the next states of F_Y is less than $\rho'_y p$ tends to one. We can restate the above statement as:

Corollary 3.6 *As $n \to \infty$ and $p \to \infty$, for every library pair (X_h, Y_h) and every $X^{(t)}$ such that $d(X_h, X^{(t)}) \leq \rho_x^{(t)} n$ ($\rho_x^{(t)} < 0.5$), the probability that $d(Y_h, Y^{(t+1)}) < \rho'_y p$ tends to one provided that ρ'_y is small, where $\rho'_y = \rho^*_y + \varepsilon$, ρ^*_y is the intersection of L_{11} and C_{11} as shown in Figure 3.6, and ε is an arbitrarily small positive number.*

Corollary 3.5 and Corollary 3.6 mean that if the number of errors in F_X in the present state is less than or equal to $\rho_x^{(t)} n$, then the number of errors in F_Y in the next state will be less than $\rho'_y p$ (denoted as $\rho_y^{(t+1)} p$) provided that ρ'_y is small.

When ρ'_y is small, we surely can use Corollary 3.5 and 3.6 to find ρ'_y. When ρ'_y is not small, we should directly solve ρ'_y based on (3.20) and then put it as $\rho_y^{(t+1)}$. **Besides, when ρ'_y is not small, for a smaller $\rho_x^{(t)}$, a smaller ρ'_y can be also obtained.** It is because the partial derivative of Φ with respect to $\rho_x^{(t)}$

$$\frac{\partial \Phi}{\partial \rho_x^{(t)}} = \log \frac{1-\rho_x^{(t)}}{\rho_x^{(t)}} + \frac{4r\rho_y(1-2\rho_x^{(t)})}{\alpha \mho} + \frac{4r\rho_y(1-2\rho_x^{(t)})}{\alpha \mho(\mho - 1)} - \frac{2\alpha}{1-2\rho_x^{(t)}} \tag{3.25}$$

is positive, [2] where

$$\mho = \sqrt{1 + \frac{4r\rho_y(1-2\rho_x^{(t)})}{\alpha^2}}.$$

In fact, Φ is an increasing function of $\rho_x^{(t)}$ because

$$f_2(u) = \left(\sqrt{1+4u^2} - 1 + \log \frac{\sqrt{1+4u^2}-1}{2u^2}\right)$$

is an increasing function. Hence, we get the following corollary.

Corollary 3.7 *As $n \to \infty$ and $p \to \infty$, for every library pair (X_h, Y_h) and every $X^{(t)}$ such that $d(X_h, X^{(t)}) \leq \rho_x^{(t)} n$ ($\rho_x^{(t)} < 0.5$), the probability that $d(Y_h, Y^{(t+1)}) < \rho'_y p$ tends to one, where ρ'_y is minimum value of ρ_y such that (3.20) holds.*

---

[2] The first three terms of (3.25) are always positive for $0 < \rho_x^{(t)} < 0.5$ and the last term is a small negative number. Also, in our interest range (such as: $\rho_x^{(t)} < 0.1$ and $\alpha < 0.2$) the first term is much larger than the last term.

The lower bound of P_X^{**} can also be obtained based on a method similar to the one above.

Theorem 3.3 *For large n and p,*

$$P_X^{**} \ \geq \ 1-\exp\left\{n\hbar(\rho_x)+\log\alpha n+rn\hbar(\rho_y^{(t+1)})-\frac{m-1}{2}\left(\sqrt{1+\frac{4r\rho_x(1-2\rho_y^{(t+1)})^2n^2}{(m-1)^2}}\right.\right.$$
$$\left.\left.-1+\log\frac{\sqrt{1+\frac{4r\rho_x(1-2\rho_y^{(t+1)})^2n^2}{(m-1)^2}}-1}{\frac{2\rho_x r(1-2\rho_y^{(t+1)})^2n^2}{(m-1)^2}}\right)\right\}.$$

From Theorem 3.3, if

$$\Theta(\rho_x) \ = \ r\hbar(\rho_y^{(t+1)}) + \hbar(\rho_x) - \frac{\alpha}{2}\left(\sqrt{1+\frac{4r\rho_x(1-2\rho_y^{(t+1)})^2}{\alpha^2}}-1\right.$$
$$\left.+\log\frac{\sqrt{1+\frac{4r\rho_x(1-2\rho_y^{(t+1)})^2}{\alpha^2}}-1}{\frac{2\rho_x r(1-2\rho_y^{(t+1)})^2}{\alpha^2}}\right) < 0, \qquad (3.26)$$

then $P_X^{**}\to 1$ as $n\to\infty$ and $p\to\infty$ (note $\frac{m-1}{n}\to\alpha$). Hence, the minimum value of ρ_x (denoted as ρ_x'), such that (3.26) holds, defines the confidence interval of number of errors in F_X in the next state.

The typical plot of $\Theta(\rho_x)$ is similar to that of $\Omega(\rho_y)$. Also, for a given $\rho_y^{(t+1)}\in[0,0.5)$, ρ_x' can be solved numerically. Also, if ρ_x' or ρ_x is small, $\Theta(\rho_x)$ can be approximated by

$$\Theta(\rho_x)\approx\Theta'(\rho_x)=r\hbar(\rho_y^{(t+1)})+\hbar(\rho_x)-\frac{\rho_x r(1-2\rho_y^{(t+1)})^2}{2\alpha}. \qquad (3.27)$$

With the above approximation, one can find out ρ_x' by considering the intersection of the line

$$L_{21}\ :\ y=\frac{\rho_x r(1-2\rho_y^{(t+1)})^2}{2\alpha} \ - \ r\hbar(\rho_y^{(t+1)}) \qquad (3.28)$$

and the curve

$$C_{21}\ :\ y=\hbar(\rho_x). \qquad (3.29)$$

Then the following two corollaries are obtained.

Corollary 3.8 *If both ρ'_{x1} and ρ'_{x2} are small ($<< \alpha^2$), $\rho_{y1}^{(t+1)} < \rho_{y2}^{(t+1)} < 0.5$ implies that $\rho'_{x1} < \rho'_{x2}$.*

Corollary 3.9 *As $n \to \infty$ and $p \to \infty$, for every library pair (X_h, Y_h) and every $Y^{(t+1)}$ such that $d(Y_h, Y^{(t+1)}) \leq \rho_y^{(t+1)}p$ ($\rho_y^{(t+1)} < 0.5$), the probability that $d(X_h, X^{(t+1)}) < \rho'_x n$ tends to one provided that ρ'_x is small, where $\rho'_x = \rho^*_x + \varepsilon$, ρ^*_x is the intersection of L_{21} and C_{21}, and ε is arbitrarily small positive number.*

It means that if the number of errors in F_Y in the present state is less than or equal to $\rho_y^{(t+1)}p$, then the number of errors in F_X in the next state is less than $\rho'_x n$ (denoted as $\rho_x^{(t+1)}n$). **When ρ'_x is not small, we should use the following corollary and then put ρ'_x as $\rho_x^{(t+1)}$.**

Corollary 3.10 *As $n \to \infty$ and $p \to \infty$, for every library pair (X_h, Y_h) and every $Y^{(t+1)}$ such that $d(Y_h, Y^{(t+1)}) \leq \rho_y^{(t+1)}p$ ($\rho_y^{(t+1)} < 0.5$), the probability that $d(X_h, X^{(t+1)}) < \rho'_x n$ tends to one, where ρ'_x is the minimum value of ρ_x such that (3.26) holds.*

It should be emphasized that $d(X_h, X^{(t+1)}) < \rho'_x n$ implies $d(X_h, X^{(t+1)}) \leq \rho'_x n$ and $d(Y_h, Y^{(t+1)}) < \rho'_y p$ implies $d(X_h, y^{(t+1)}) \leq \rho'_y p$. Hence, by iteratively solving ρ'_x and ρ'_y, we can construct two sequences, $\rho_x^{(t)}$ and $\rho_y^{(t)}$. These sequences are the statistical dynamics about the confidence interval of the number of errors. A typical dynamics is shown in Figure 3.6. From the figure, the sequences rapidly converge to the stable states[3] (ρ_x^f, ρ_y^f). We can use the dynamics to estimate the memory capacity, the attraction basin, and the number of errors in the retrieval pairs.

[3]Here, the stable states are referred with respect to the sequences ($\rho_x^{(t)}$, $\rho_y^{(t)}$) only and are not referred respect to the states of neurons.

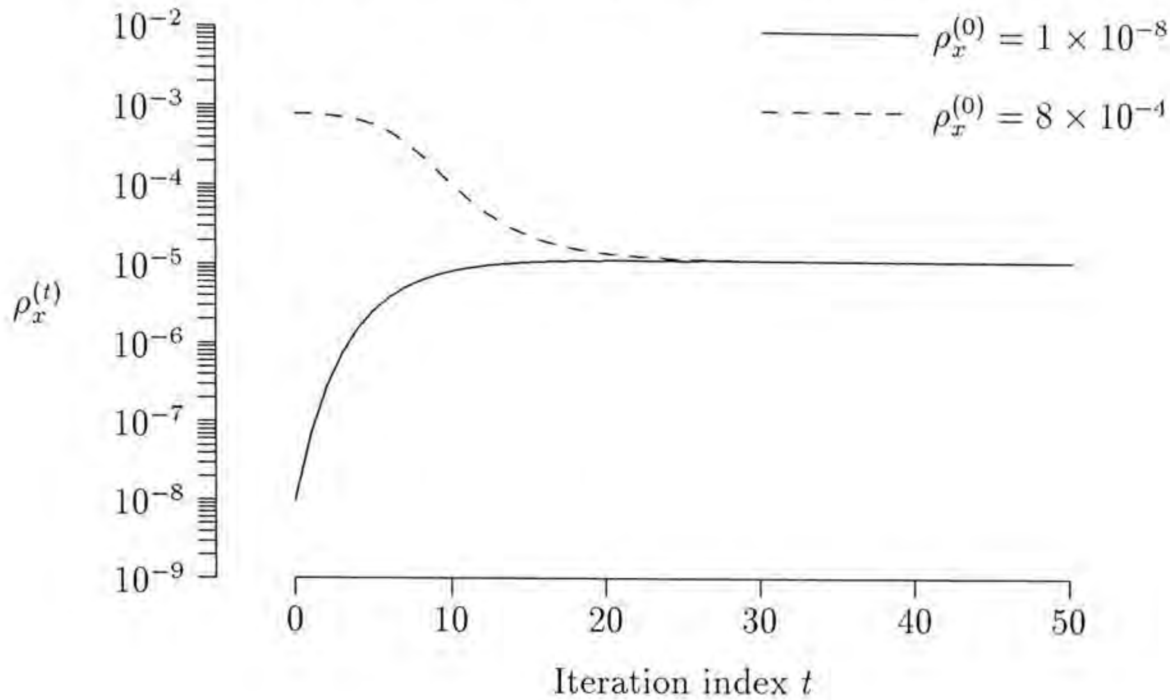
Given $m = \alpha n$ and $\rho_x^{(0)}$, if the sequences $(\rho_x^{(t)}, \rho_y^{(t)})$ converge to (ρ_x^f, ρ_y^f) which are less than $\rho_x^{(0)}$ (i.e. the library pairs can be retrieved with a small number of errors.), then the memory capacity of BAM is at least equal to αn. Note that with a smaller α, the sequences will also converge to the values which are less than (ρ_x^f, ρ_y^f). It is because Φ' and Θ' are the increasing functions of α. The above claim of Φ' and Θ' comes from the following fact :

$$f_1(u) = \frac{1}{u}\left(\sqrt{1+4u^2} - 1 + \log\frac{\sqrt{1+4u^2}-1}{2u^2}\right)$$

is an increasing function. Because

$$\frac{df_1}{du} = -\log(\frac{\sqrt{1+4u^2}-1}{2u^2})u^{-2} > 0.$$

(a) Dynamics of $\rho_x^{(t)}$ for $r = 1$ and $\alpha = 0.02$



(b) Dynamics of $\rho_y^{(t)}$ for $r = 1$ and $\alpha = 0.02$

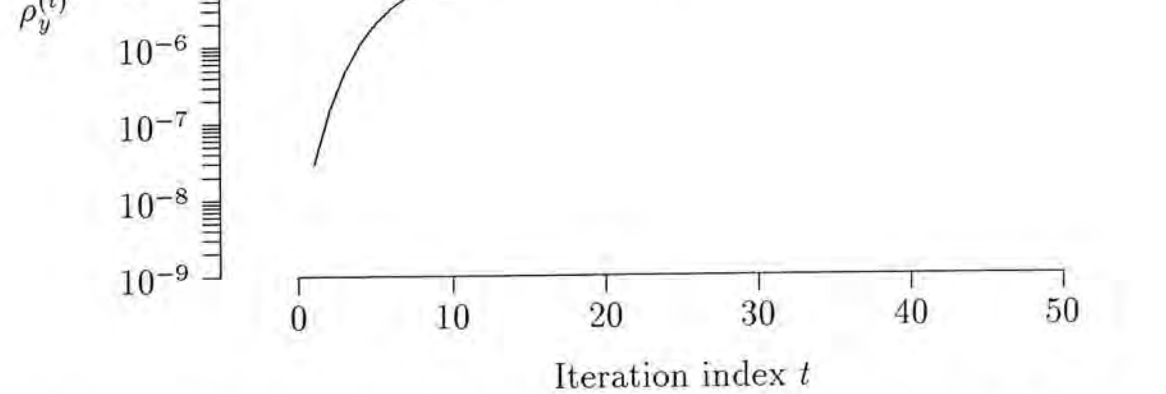


Figure 3.6 The Confidence Dynamics of the number of errors of BAM, where $r = 1$ and $\alpha = 0.02$.

With the initial number of errors ($\rho_x^{(0)}n$) being nonzero, if the sequences converge to the values (ρ_x^f and ρ_y^f) which are less than 0.5 and $\rho_x^{(0)}$ (It means the library pairs can be retrieved with a small number of errors when the initial errors is less than or equal to $\rho_x^{(0)}n$.), the attraction basin of each library pair is at least equal to $\rho_x^{(0)}n$. The maximum value of $\rho_x^{(0)}$, such that the sequences converge to (ρ_x^f, ρ_x^f) which values are less than $\rho_x^{(0)}$, reflects the lower bound of the attraction basin. We denote this maximum value as $\rho_{maxinit,r}$. Also, ρ_x^f and ρ_y^f reflect the upper bound of the number of errors in the retrieval pairs. Based on the (3.22) and (3.27), it is not difficult to see the following relation between ρ_x^f and ρ_y^f.

Corollary 3.11 *If the sequences $(\rho_x^{(t)}, \rho_y^{(t)})$ converge to (ρ_x^f, ρ_y^f) which values are much less then α^2, then*

$$\rho_x^f \approx \rho_y^f$$

no matter what value r takes.

Proof of Corollary 3.11

At the limit point $(0 < \rho_x^f < 0.5,\ 0 < \rho_y^f < 0.5)$ of the sequences (the intersection of L_{11} and C_{11} is equal to the intersection of L_{21} and C_{21}),

$$\frac{\rho_x^f}{\rho_y^f} \approx \frac{(1-2\rho_x^f)^2}{(1-2\rho_x^f)^2}.$$

If $\rho_x^f > \rho_y^f$, then the left hand side is greater than one (since $0 < \rho_x^f$ and $0 < \rho_y^f$) but the right hand side is less than one (since $0 < \rho_x^f < 0.5$ and $0 < \rho_y^f < 0.5$). Hence, the only solution of the above equation is $\rho_x^f \approx \rho_y^f$. □

With Corollary 3.11 and (3.22), we can easily obtain Corollary 3.12 and Corollary 3.13 which can be used to directly estimate ρ_x^f and ρ_y^f instead of solving the dynamics. Also, Corollary 3.13 shows the symmetric property of ρ_x^f between α and r. The symmetric property means that interchanging p and n does not affect the estimated ρ_x^f and ρ_y^f.

Corollary 3.12 *Given α and r, for small ρ_x^f and ρ_y^f,*

$$\rho_x^f \approx \rho_y^f \approx \exp(-\frac{r}{2(1+r)\alpha}+1) \; .$$

Corollary 3.13 *Given $\alpha = \frac{\alpha'}{r'}$ and $r = \frac{1}{r'}$, for small ρ_x^f and ρ_y^f,*

$$\rho_x^f \approx \rho_y^f \approx \exp(-\frac{r'}{2(1+r')\alpha'}+1) \; .$$

Result from other approach

If we directly follow the approach used in the Hopfield network [60] to estimate the confidence interval of the number of errors for the BAM, we will obtain the following two corollaries. Note that in [60], the approach is only employed to estimate the confidence interval of the number of errors for the Hopfield network. The following two corollaries for the BAM are derived by us based on the approach in [60]. However, we find that the estimation from the following two corollaries is not good. Hence, we derive our formulation to estimate the statistical properties.

Corollary 3.14 *As $n \to \infty$ and $p \to \infty$, for every library pair (X_h, Y_h) and every $X^{(t)}$ such that $d(X_h, X^{(t)}) \leq \rho_x^{(t)} n$ ($\rho_x^{(t)} < 0.5$), $d(Y_h, Y^{(t+1)}) > \rho_y p$ with probability $me^{-\triangle}$ if*

$$r\rho_y(1-2\rho_x^{(t)}) - (z+\sqrt{2z})\alpha\sqrt{r\rho_y} > 0 \, , \tag{3.30}$$

where

$$z = \frac{\hbar(\rho_x^{(t)}) + r\hbar(\rho_y)}{\alpha} \, .$$

Corollary 3.15 *As $n \to \infty$ and $p \to \infty$, for every library pair (X_h, Y_h) and every $Y^{(t+1)}$ such that $d(Y_h, Y^{(t+1)}) \leq \rho_y^{(t+1)} p$ ($\rho_y^{(t+1)} < 0.5$), $d(X_h, X^{(t+1)}) > \rho_x n$ with probability $me^{-\Delta}$ if*

$$r\rho_x(1 - 2\rho_y^{(t+1)}) - (z' + \sqrt{2z'})\alpha\sqrt{r\rho_x} > 0\,, \tag{3.31}$$

where

$$z' = \frac{r\hbar(\rho_y^{(t+1)}) + \hbar(\rho_x)}{\alpha}\,.$$

We can numerically find the minimum values of ρ_y and ρ_x such that (3.30) and (3.31) hold. These values define the confidence interval of the number of errors in the next state. However, the lower bound of the memory capacity and the lower bound of the attraction basin estimated from (3.30) and (3.31) are usually poorer than those estimated from our approach. For example, from (3.30) and (3.31), the lower bound of the memory capacity is $0.017n$ (for $r = 1$) and the lower bound of the attraction basin is $0.0007n$ (for $r = 1$ and $\alpha = 0.01$). In our approach, the lower bound of the memory capacity is $0.022n$ (for $r = 1$) and the lower bound of the attraction basin is $0.003n$ (for $r = 1$ and $\alpha = 0.01$). Hence, in the next section we use our approach to estimate the memory capacity, the attraction basin and the number of errors in the retrieval pairs for the BAM.

3.5 Numerical Results from the Dynamics

Numerical Example a:

In this example, we numerically estimate the lower bound of the memory capacity of BAM for different values of r. For a given r, let α_r be the largest value of α such that the sequences converge. Then $\alpha_r n$ can be considered as a lower bound of the memory capacity of BAM. Here, we estimate the dynamics based on the Corollary 3.7 and Corollary 3.10. The result is summarized in Table 3.2. From the table, the lower bound increases with r when $r < 10$. However, the lower bound decreases as r increases for large r ($r > 10$). Also, there are some symmetrical results about α_r. That is

$$\alpha_{\frac{1}{r}} \approx \frac{\alpha_r}{r}.$$

It means that interchanging p and n does not affect the overall estimated lower bound.

The advantages of the dynamics approach are, 1) it can be used to estimate the attraction basin, and 2) we do not need to find a suitable energy function.[4]

Table 3.2 The lower bound of memory capacity of BAM from the dynamic approach.

r	α'
50	0.0334
20	0.0353
10	0.036
5	0.035
2	0.029
1	0.022
0.5	0.0147
0.2	0.00705
0.1	0.00362

[4]This approach will be used to estimate the properties of the higher order BAM in the next chapter.

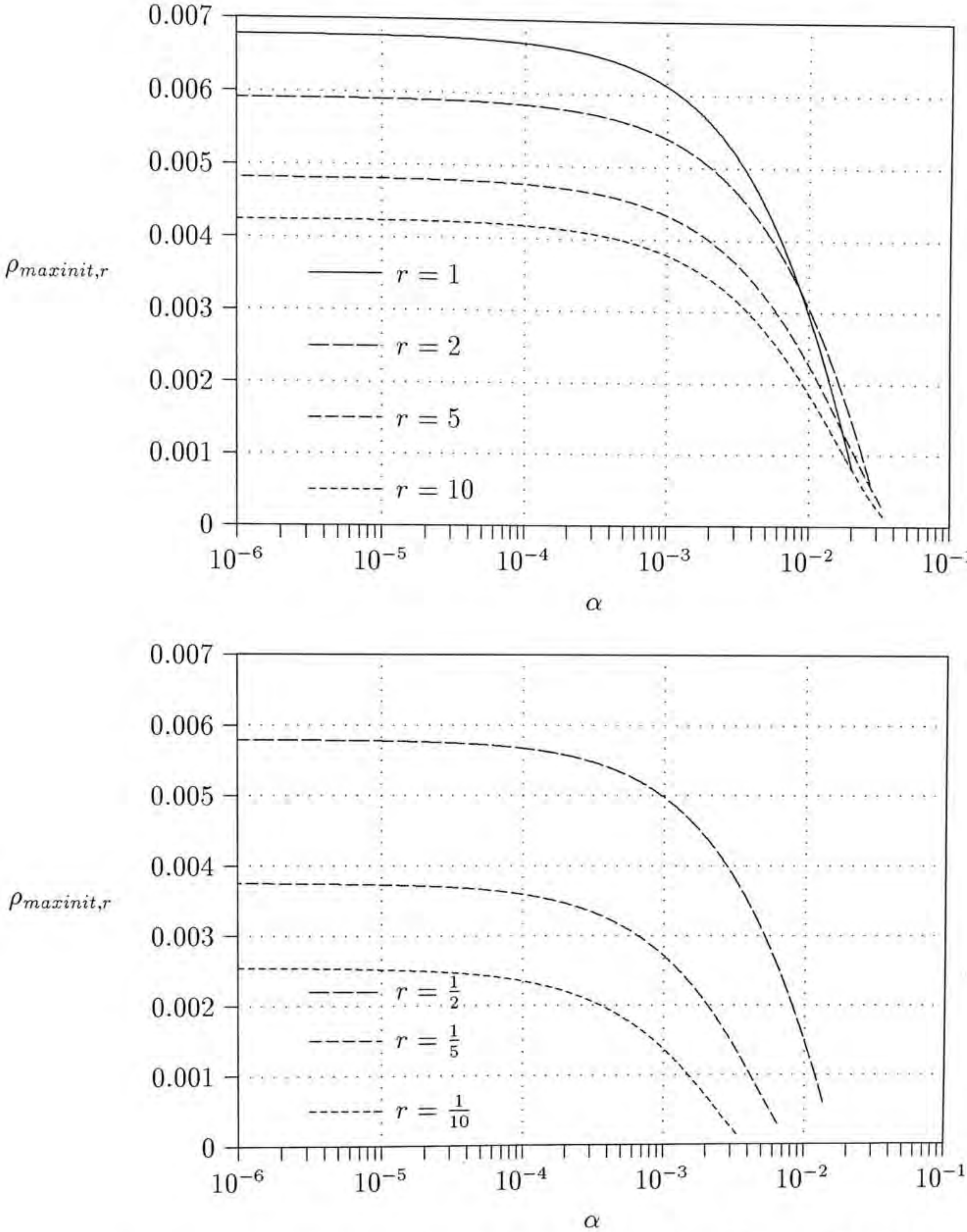


Figure 3.7 The lower bound of the attraction basin of BAM.

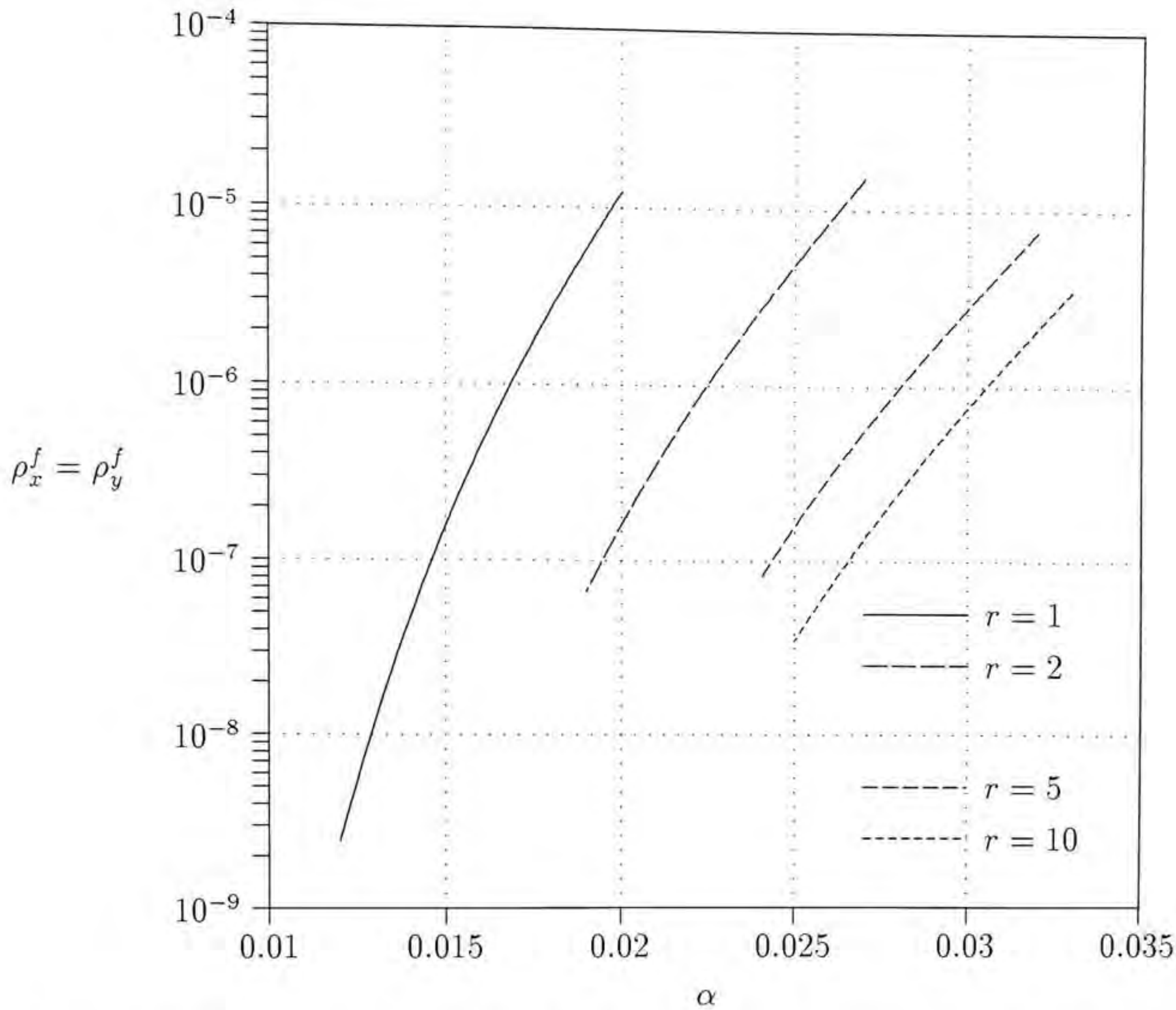


Figure 3.8 The upper bound of the number of errors in the retrieval pairs of BAM based on Corollary 3.7 and 3.10.

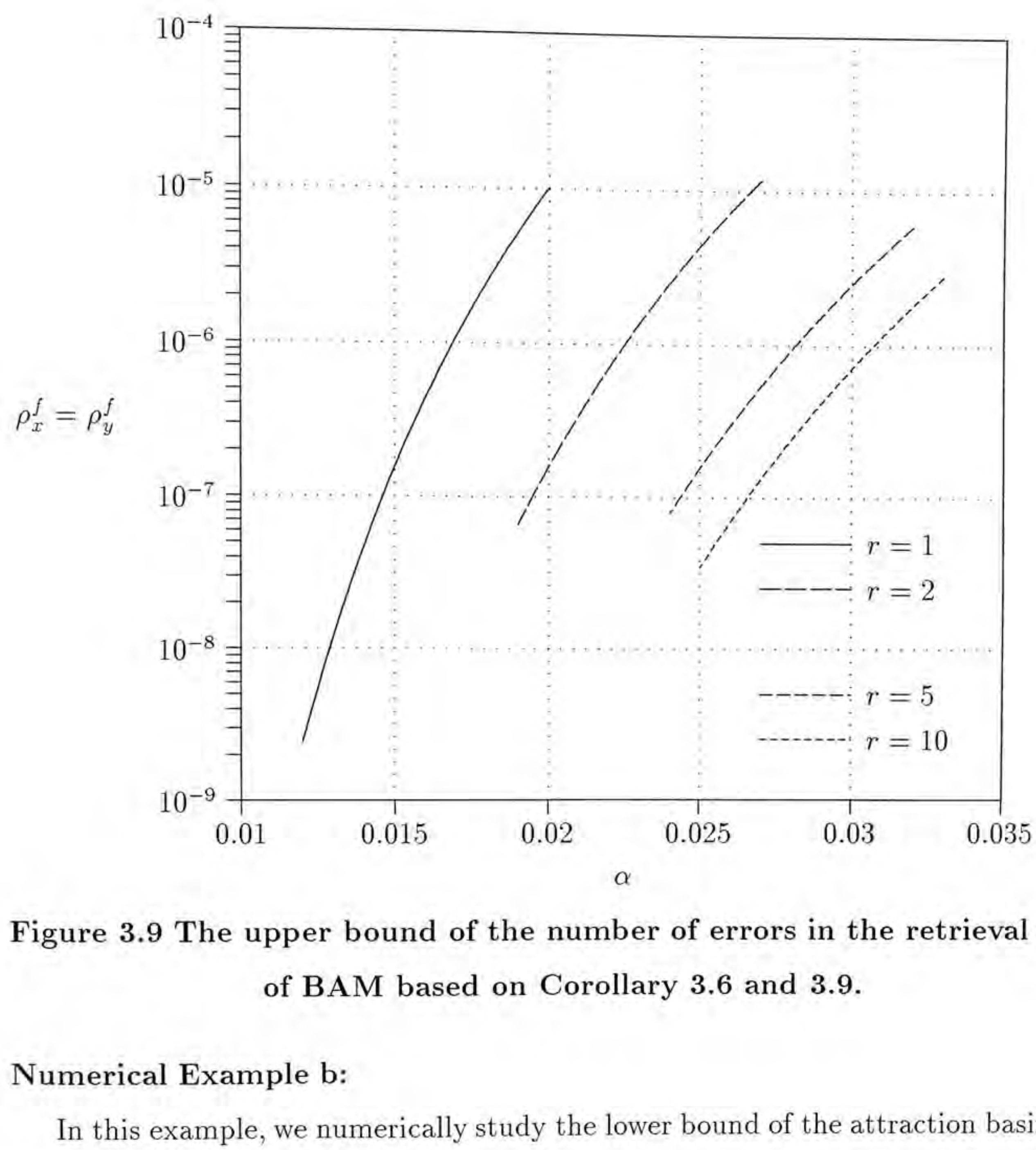


Figure 3.9 The upper bound of the number of errors in the retrieval pairs of BAM based on Corollary 3.6 and 3.9.

Numerical Example b:

In this example, we numerically study the lower bound of the attraction basin and the upper bound of the number of errors in the retrieval pairs. For a given α, let $\rho_{maxinit,r}$ be the largest value of $\rho_x^{(0)}$ such that the sequences converge. Figure 3.7 and Figure 3.8 summarize the lower bounds of the attraction basin and the upper bounds of the number of errors in the retrieval pairs at different values of α, respectively. In Figure 3.8 we do not show the cases of $r = \frac{1}{2}$, $\frac{1}{5}$, and $\frac{1}{10}$. It is because such cases (with suitable change in α) are very similar to those of $r = 2$, 5, and 10, respectively (see Corollary 3.11). [5]

[5] To change α in Figure 3.8 to $\frac{\alpha}{r}$ and to change r to $\frac{1}{r}$ will get the results of $r = \frac{1}{2}$, $\frac{1}{5}$, and $\frac{1}{10}$.

- Construction of Figure 3.8 is from Corollary 3.7 and Corollary 3.10 since the value of the lower bound of the attraction basin is large. As α decreases, the lower bound of the attraction basin starts increase. When α is very small (about 10^{-5}), the lower bound reach a upper limit and will no longer increase. From Figure 3.7, the lower bound of the attraction basin is the best when $r = 1$.

- From Figure 3.8, the upper bound of the number of errors in the retrieval pairs ρ_x^f (or ρ_y^f) exponentially decreases as α decreases. This property of ρ_x^f (or ρ_y^f), estimated from the dynamics shown in Figure 3.8, agrees with Corollary 3.12.

- As a comparison, we also use Corollary 3.6 and Corollary 3.9 to estimate ρ_x^f. The results are summarized in Figure 3.9. Comparing Figure 3.8 with Figure 3.9, we can conclude that Corollary 3.6 and Corollary 3.9 are good approximation of Corollary 3.7 and Corollary 3.10 when the values of ρ_x^f and ρ_y^f are small.

3.6 Chapter Summary

Under the considerations 'for every library pair' and 'for every error pattern', we have examined the memory capacity and the number of errors in the retrieval pairs of BAM based on the concept of the energy barrier when a small number of errors are allowed in the retrieval pairs. The results are,

- The memory capacity can grow as far as $\alpha_r n$ which depends on the ratio of the dimensions: $r = \frac{p}{n}$ (see Table 3.1).

- The number of errors in the retrieval pairs is bounded by $O(\exp\{-\frac{r}{4(1+r)\alpha}\}n)$, where the number of the library pairs is αn (see Figure 3.3).

Also, we have pointed out the limitation of this approach. That is, the attraction basin cannot be known.

Then, the statistical dynamics of BAM is introduced. Based on this dynamics, we can estimate the memory capacity, the attraction basin, and the number of errors in the retrieval pairs.

- The memory capacity can grow as far as $\alpha_r n$ which depends on the ratio of the dimensions: $r = \frac{p}{n}$ (see Table 3.2).

- The lower bound of the attraction basin is a function of α and r (see Figure 3.8).

- The number of errors in the retrieval pairs is bounded by $O(\exp\{-\frac{r}{2(1+r)\alpha}\}n)$, where the number of the library pairs is αn (see Figure 3.8 and Figure 3.9).

Although the energy approach can directly estimate the memory capacity, the advantage of the dynamics approach is that it can be used to estimate the attraction basin and it is suitable to analyze the associative memories without finding out a suitable energy function.[6] In the next chapter, the statistical dynamics of the higher order BAM will be developed based on the dynamics approach.

[6] In the dynamics approach, we first construct the sequences and then we use the the properties of the sequences to estimate the statistical properties. So, the dynamics approach is an indirect approach.

Chapter 4

Stability and Statistical Dynamics of Second order BAM

The second order BAM, as well as the general higher order BAM, are the enhanced versions of BAM. The interesting point is whether the second order BAM, as well as the higher order BAM, have the similar statistical properties attributed to the first order BAM. In this chapter, the statistical dynamics of the second order BAM is first presented. From the dynamics, we can estimate the attraction basin, the memory capacity, and the number of errors in the retrieval pairs for the second order BAM. Finally, we extend the results to the general higher order BAM.

4.1 Introduction

As mentioned in Chapter 1, several encoding methods have been developed to improve the memory capacity of the BAM. One of these is to introduce higher order connections [45] (resulted in the so-called higher order BAM). In [45, 46], the empirical memory capacity of the second order BAM, as well as the higher order BAM, was studied. However, the theoretical memory capacity of the second/higher order BAM has not been given yet. In [45], the second order BAM was proved to be stable, i.e., its state always converges to a fixed point and the energy function which is similar to the energy function used in [59] always decreases during the recall. But as we will show, its state may converge to limit cycles and the energy function may increase during recall.

This chapter presents the stability and statistical behavior of the second order BAM. We will first use an example to demonstrate that its state may converge to limit cycles. Hence, the stabilization of the second order BAM is not guaranteed during recall. Also, we will point out a mistake in [45], which leads to its wrong conclusion that the state of the second order BAM always converges to a fixed point and that the energy function in [45] (the format of which is similar to that in [59]) always decreases during recall. Hence, we cannot use the approach of energy barrier in [59] (or similar energy function) to estimate the memory capacity of the second order BAM. Instead of finding out another suitable energy function, we will examine the statistical properties of the second order BAM based on the statistical dynamics used in Chapter 3. Note that it is difficult to find a suitable energy function which always decreases during recall and can be used to estimate a good memory capacity.

Section 4.2 presents the stability. The statistical dynamics of the second order BAM is introduced in Section 4.3. The numerical examples are shown in Section 4.4. In Section 4.5, we will discuss how to generalize the above results to the general higher order cases. In Section 4.4 and 4.5, we use a theory of large deviation, which is derived by Newman [59], to develop the dynamics. In the theory, some conditions must be fulfilled. The checking procedure of these conditions is put in Section 4.6 as a supplementary reading material. A concluding remark will be made in Section 4.7.

4.2 Second order BAM and its Stability

The second order BAM is a heteroassociative memory that stores bipolar library pairs, (X_h, Y_h), $h = 1, \ldots, m$, where $X_h \in \{+1, -1\}^n$, $Y_h \in \{+1, -1\}^p$, and m is the number of the library pairs. It encodes them into two separate matrices. The first matrix, U, is a $n \times n \times p$ lattice that holds the second order connections from F_X to F_Y. The second matrix, V, is a $p \times p \times n$ lattice that holds the connections from F_Y to F_X.

The matrix $U = [u_{kji}]$ is constructed according to the correlation rule:

$$u_{kji} = \sum_{h=1}^{m} y_{kh}\, x_{jh}\, x_{ih} \text{ for } j = 1, \cdots, n,\ i = 1, \cdots, n, \text{ and } k = 1, \cdots, p. \tag{4.1}$$

Also, the matrix $V = [v_{jkl}]$ is

$$v_{jkl} = \sum_{h=1}^{m} x_{jh}\, y_{kh}\, y_{lh} \text{ for } l = 1, \cdots, p,\ k = 1, \cdots, p, \text{ and } j = 1, \cdots, n. \tag{4.2}$$

Note that $v_{jkl} = v_{jlk}$ and $u_{kji} = u_{kij}$. The recall process of the second order BAM works in the same fashion as the first order BAM. That is

$$y_k^{(t+1)} = \operatorname{sgn}\left(\sum_{j=1}^{n}\sum_{i=1}^{n} u_{kji}\, x_j^{(t)}\, x_i^{(t)}\right) \tag{4.3}$$

$$x_j^{(t+1)} = \operatorname{sgn}\left(\sum_{k=1}^{p}\sum_{l=1}^{p} v_{jkl}\, y_k^{(t+1)}\, y_l^{(t+1)}\right). \tag{4.4}$$

The second order BAM belongs to the class of finite state autonomous systems. The number of states is finite and the next state only depends on its present state. One can easily verify that a finite-state autonomous system either converges to fixed points or limit cycles.

Unlike the first order BAM, the state of the second order BAM may converge to limit cycles. Consider the following library pairs:

$$\begin{aligned}
X_1 &= (-1,1,-1,1,1,-1)^T \\
Y_1 &= (-1,1,1,-1,1,-1)^T \\
X_2 &= (1,1,-1,-1,1,-1)^T \\
Y_2 &= (-1,-1,1,-1,-1,-1)^T \\
X_3 &= (-1,-1,-1,-1,1,-1)^T \\
Y_3 &= (-1,-1,1,-1,-1,1)^T \\
X_4 &= (-1,1,-1,-1,1,-1)^T \\
Y_4 &= (1,-1,-1,-1,1,-1)^T \\
X_5 &= (1,1,-1,1,1,1)^T \\
Y_5 &= (1,1,-1,1,-1,-1)^T
\end{aligned}$$

and the initial state is:

$$
\begin{aligned}
X^{(0)} &= (-1,-1,1,-1,1,1)^T \\
Y^{(0)} &= (1,1,-1,1,1,1)^T.
\end{aligned}
$$

According to the updating rule, the following sequence can be obtained:

$$
\begin{aligned}
Y^{(1)} &= \operatorname{sgn}(-8,0,8,-8,-8,-8)^T &&= (-1,1,1,-1,-1,-1)^T \\
X^{(1)} &= \operatorname{sgn}(-8,32,-40,-8,40,-40)^T &&= (-1,1,-1,-1,1,-1)^T \\
Y^{(2)} &= \operatorname{sgn}(-12,-52,12,-84,20,-52)^T &&= (-1,-1,1,-1,1,-1)^T \\
X^{(2)} &= \operatorname{sgn}(8,48,-56,8,56,-24)^T &&= (1,1,-1,1,1,-1)^T \\
Y^{(3)} &= \operatorname{sgn}(-12,12,12,-20,-12,-52)^T &&= (-1,1,1,-1,-1,-1)^T \\
X^{(3)} &= \operatorname{sgn}(-8,32,-40,-8,40,-40)^T &&= (-1,1,-1,-1,1,-1)^T \\
Y^{(4)} &= \operatorname{sgn}(-12,-52,12,-84,20,-52)^T &&= (-1,-1,1,-1,1,-1)^T \\
X^{(4)} &= \operatorname{sgn}(8,48,-56,8,56,-24)^T &&= (1,1,-1,1,1,-1)^T \\
Y^{(5)} &= \operatorname{sgn}(-12,12,12,-20,-12,-52)^T &&= (-1,1,1,-1,-1,-1)^T \\
X^{(5)} &= \operatorname{sgn}(-8,32,-40,-8,40,-40)^T &&= (-1,1,-1,-1,1,-1)^T \\
Y^{(6)} &= \operatorname{sgn}(-12,-52,12,-84,20,-52)^T &&= (-1,-1,1,-1,1,-1)^T \\
X^{(6)} &= \operatorname{sgn}(8,48,-56,8,56,-24)^T &&= (1,1,-1,1,1,-1)^T \\
\vdots \;\; & \vdots \;\; \vdots && \vdots \;\; \vdots
\end{aligned}
$$

Clearly, the sequence $(X^{(t)}, Y^{(t)})$ converges to a limit cycle. Hence, the stabilization of the second order BAM is not guaranteed.

In [45], an energy function, which is similar to the energy function used in [59], is proposed to explain the stability of the second order BAM. Let us first review the work in [45] and then point out the mistake. The energy function of the second order BAM is expressed in [45] as

$$
\begin{aligned}
E_2 &= -(E_{2Y} + E_{2X}) \\
&= -\sum_{h=1}^{m}(X_h^T X)^2 (Y_h^T Y) - \sum_{h=1}^{m}(Y_h^T Y)^2 (X_h^T X)
\end{aligned}
\tag{4.5}
$$

where (X, Y) is the current state of the neurons, the term E_{2Y} is the second order energy contribution of F_Y neurons, and the term E_{2X} is the second order energy contribution of F_X neurons. If some neurons in F_X change state, the change of energy in F_X is

$$\Delta E_{2X} = -\sum_{h=1}^{m}(Y_h^T Y)^2(X_h^T \Delta X), \tag{4.6}$$

where $\Delta X = (\Delta x_1, \cdots, \Delta x_n)^T$ and Δx_i is the change in the ith element of X. If some neurons in F_Y change state, the change of energy in F_Y is

$$\Delta E_{2Y} = -\sum_{h=1}^{m}(X_h^T X)^2(Y_h^T \Delta Y). \tag{4.7}$$

By showing that ΔE_{2X} and ΔE_{2Y} are always negative, [45] claimed that the second order BAM is always stable.

From our counter example, the stabilization of the second order BAM is not guaranteed but [45] claimed that it is always stable. The flaw in [45] is that the change of the energy in F_Y due to the change of state in F_X (and the change of the energy in F_Y due to the change of state in F_X) had been neglected. In fact, if the state of F_X is changed, the total change of energy is

$$\begin{aligned}\Delta E_{2X} &= -\sum_{h=1}^{m}(Y_h^T Y)^2(X_h^T \Delta X) - \sum_{h=1}^{m}(X_h^T \Delta X)^2(Y_h^T Y) \\ &\quad -\sum_{h=1}^{m} 2(X^T \Delta X)(X_h^T \Delta X)(Y_h^T Y).\end{aligned} \tag{4.8}$$

The last two terms in the above equation may be negative or positive. Hence, the energy function, proposed in [45], may *increase* during recall and cannot be used to explain the stabilization. As shown by our example, its state may converge to limit cycles. Hence, the second order BAM is an unstable model.

The above consideration is based on the layer-synchronous recall process in which one of the two layers is updated at a time. In fact, the layer-synchronous recall process is also the asynchronous recall process with the updating order: each neuron in a layer is updated one by one and then each neuron in the other layer is updated one by one. Therefore, the stabilization of the second order BAM is not guaranteed under both layer-synchronous and asynchronous recall processes.

In [59], Newman used a similar energy function (similar to the energy function used in [45]) to estimate the memory capacity of the higher order Hopfield network based on the stabilization of the higher order Hopfield network (or the stabilization of the energy function). Since the stabilization of the second order BAM is not guaranteed (under the energy function in [45]), we cannot use the Newman's approach (under the similar energy function) to estimate the memory capacity of the second order BAM. Instead of finding out another suitable energy function, we will use the statistical dynamics to examine the statistical properties of the second order BAM in the next section.

4.3 Confidence Dynamics of Second Order BAM

The notations and assumptions used here are similar to those used in Section 3.4. The only exception is that

$$m = \alpha n^2 . \tag{4.9}$$

We first introduce two probabilities P_Y^{**} and P_X^{**} for the second order BAM.

Definition 4.1 *For the second order BAM, given that $p = rn$ and $m = \alpha n^2$, let P_Y^{**} be the probability that for every library pair (X_h, Y_h) and for any $\rho_x^{(t)} n$ errors in F_X in the present state (i.e. the Hamming distance between X_h and $X^{(t)}$ is equal to $\rho_x^{(t)} n$), the number of errors in F_Y in the next state is less than $\rho_y p$ (i.e. the Hamming distance between Y_h and $Y^{(t)}$ is less than $\rho_y p$).*

Definition 4.2 *For the second order BAM, given that $p = rn$ and $m = \alpha n^2$, let P_X^{**} be the probability that for every library pair (X_h, Y_h) and for any $\rho_y^{(t+1)} p$ errors in F_Y in the present state (i.e. the Hamming distance between Y_h and $Y^{(t+1)}$ is equal to $\rho_y^{(t+1)} p$), the number of errors in F_X in the next state is less than $\rho_x n$ (i.e. the Hamming distance between X_h and $X^{(t+1)}$ is less than $\rho_x n$).*

The estimation of P_Y^{**} and P_X^{**} is based on an existing theory from large deviation (Proposition 3.4 in [59]). Here, we restate it as the following lemma.

Lemma 4.1 *Newman's Lemma: Suppose $\chi_N^1, \chi_N^2, \cdots$ are, for each N, independent, identically distributed and symmetric random variables satisfying:*

1.

$$\lim_{N\to\infty} Var(\chi_N^1) = \sigma^2 \in (0, \infty). \tag{4.10}$$

2. For some real $L > 2$ and $t_o > 0$,

$$\limsup_{N\to\infty} \left\{ E(\exp(t_o \mid \chi_N^1 \mid^{2/L})) \right\} < \infty. \tag{4.11}$$

For any $\gamma \in (0, \infty)$ and

$$\Re = \frac{\gamma^2}{2\sigma^2}, \tag{4.12}$$

a sufficient condition for

$$Prob\left(M^{-1} \sum_{s=1}^{M} \chi_N^s \geq \gamma M^{-\frac{L-2}{2L-2}} \right) \leq \exp\left(-\Re M^{\frac{1}{L-1}} \right) \tag{4.13}$$

as M, $N \to \infty$, is

$$\gamma^{2L-2} < 2^{L-2}(\sigma^2 t_o)^L. \tag{4.14}$$

The proof of the above lemma can be found in [59] and we will not show it here again. Note that in [59] the above lemma is used to estimate the probability of the existence of an energy barrier around a library pattern in the higher order Hopfield network. Here we use the lemma to estimate P_Y^{**} and P_X^{**}. Then we can create the dynamics of the fraction of errors of the second order BAM.[1] Based on Lemma 4.1, we can get the estimation about Prob($\overline{EA_{h,g}}$).

[1] As mentioned in Section 4.1, we cannot use the similar energy function [45, 59] to estimate the memory capacity of the second order BAM.

Lemma 4.2 *For the second order BAM, as $n \to \infty$, and $p \to \infty$*

$$Prob(\overline{EA_{g,h}}) < \exp\left\{ rn\hbar(\rho_y) - \frac{\rho_y rn(1-2\rho_x^{(t)})^4}{6\alpha} \right\}$$

provided that

$$\left(\frac{\sqrt{\rho_y r}(1-2\rho_x^{(t)})^2}{\sqrt{3}\alpha^{3/4}} \right)^4 < \frac{3}{2},$$

for $g = 1, \cdots, \binom{n}{\rho_x^{(t)}}$ and $h = 1, \cdots, m$.

Proof of Lemma 4.2

Without loss of generality, we consider that the library pair (X_h, Y_h) are: $X_h = (1, \ldots, 1)^T$ and $Y_h = (1, \ldots, 1)^T$. Note that $\overline{EA_{g,h}}$ is the event that for a given $X^{(t)} \in S_{h,t}$[2], the number of errors in F_Y in the next state is larger than or equal to $\rho_y p$. Let J be the set of indices in which $X^{(t)}$ and X_h differ. Note that there are $\binom{n}{\rho_x^{(t)} n}$ such sets. Also, let K be the set of indices of Y_h and $Y^{(t)}$ such that $|K| = \rho_y p$. Note that there are $\binom{p}{\rho_y p}$ such sets.

The event $\overline{EA_{g,h}}$ implies that there is at least one K such that

$$\sum_{k \in K} \sum_{j=1}^{n} \sum_{i=1}^{n} u_{kji}\, x_i^{(t)}\, x_j^{(t)} < 0 ,$$

where $|K| = \rho_y p$. Clearly, we have

$$\begin{aligned} \mathrm{Prob}(\overline{EA_{g,h}}) &\leq \mathrm{Prob}\Big(\text{ there is at least one } K \text{ where } |K| = \rho_y p \text{ such that} \\ &\qquad \sum_{k \in K} \sum_{j=1}^{n} \sum_{i=1}^{n} u_{kji}\, x_i^{(t)}\, x_j^{(t)} < 0 \Big) \\ &\leq \binom{p}{\rho_y p} \mathrm{Prob}\left(\sum_{k \in K} \sum_{j=1}^{n} \sum_{i=1}^{n} u_{kji}\, x_i^{(t)}\, x_j^{(t)} < 0 \text{ for a given } K \right) . \end{aligned} \tag{4.15}$$

Let

$$P'' = \mathrm{Prob}\left(\sum_{k \in K} \sum_{j=1}^{n} \sum_{i=1}^{n} u_{kji}\, x_i^{(t)}\, x_j^{(t)} < 0 \text{ for a given } K \right) . \tag{4.16}$$

[2] $S_{h,t}$ is the set of $X \in \{1, -1\}^n$ such that $d(X, X_h) = \rho_x^{(t)} n$

Substituting (4.1) and (4.2) into above,

$$P'' = \text{Prob}\left(\rho_y p(1-2\rho_x^{(t)})^2 n^2 + \sum_{h'\neq h}^{m}\sum_{k\in K} y_{kh'}\left(\sum_{j=1}^{n} x_{jh'}x_j^{(t)}\right)^2 < 0\right). \tag{4.17}$$

One can easily find that

$$E\left[\sum_{k\in K} y_{kh'}\left(\sum_{j\notin J}^{n} x_{jh'} - \sum_{j\in J} x_{jh'}\right)^2\right] = 0$$

and

$$E\left[\left(\sum_{k\in K} y_{kh'}\left(\sum_{j\notin J}^{n} x_{jh'} - \sum_{j\in J} x_{jh'}\right)^2\right)^2\right] = \rho_y p(3n^2-2n).$$

Since the random variables in (4.17) are symmetric,

$$P'' = \text{Prob}\left(\frac{1}{m-1}\sum_{h'\neq h}^{m} \chi_{h'} > \gamma(m-1)^{-1/4}\right) \tag{4.18}$$

where

$$\gamma = \frac{\sqrt{\rho_y r}(1-2\rho_x^{(t)})^2 n^{(5/2)}}{\sqrt{3n^2-2n}(m-1)^{(3/4)}}, \tag{4.19}$$

and

$$\chi_{h'} = \frac{\sum_{k\in K} y_{kh'}\left(\sum_{j\notin J}^{n} x_{jh'} - \sum_{j\in J} x_{jh'}\right)^2}{\sqrt{(3n^2-2n)(\rho_y p)}}. \tag{4.20}$$

Also, as n, $p \to \infty$ (then $\frac{m-1}{n} \to \alpha$),

$$\gamma \to \frac{\sqrt{\rho_y r}(1-2\rho_x^{(t)})^2}{\sqrt{3}\alpha^{(3/4)}}.$$

Applying Lemma 4.1 to (4.18), putting $L = 3$ and t_o being slight less than $2^{-2/3}3^{1/3}$ (Checking whether $\chi_{h'}$'s satisfy the two conditions, (4.10) and (4.11), will be shown in Section 4.6.),

$$P'' \leq \exp\left\{-\frac{\rho_y rn(1-2\rho_x^{(t)})^4}{6\alpha}\right\}.$$

Then

$$\text{Prob}(\overline{EA_{g,h}}) \leq \binom{p}{\rho_y p}\exp\left\{-\frac{\rho_y rn(1-2\rho_x^{(t)})^4}{6\alpha}\right\}. \tag{4.21}$$

Replacing the binomial coefficient in (4.21) with Stirling's asymptotic formula,

$$\text{Prob}(\overline{EA_{g,h}}) \leq \exp\left\{rn\hbar(\rho_y) - \frac{\rho_y rn(1-2\rho_x^{(t)})^4}{6\alpha}\right\}.$$

The proof is completed. □

By the definition of the event EA,

$$\begin{aligned}\text{Prob}(EA) = P_Y^{**} &= 1 - \text{Prob}(\overline{EA}) \\ &\geq 1 - \sum_{h=1}^{m} \sum_{g=1}^{\binom{n}{\rho_x^{(t)} n}} \text{Prob}(\overline{EA_{g,h}}) \\ &= 1 - m \binom{n}{\rho_x^{(t)} n} \text{Prob}(\overline{EA_{1,1}}). \end{aligned} \tag{4.22}$$

With Lemma 4.2, we can immediately get the lower bound of P_Y^{**}.

Theorem 4.1 *For the second order BAM, for large n and p,*

$$P_Y^{**} \geq 1 - \exp\left\{ n\hbar(\rho_x^{(t)}) + \log \alpha n^2 + rn\hbar(\rho_y) - \frac{\rho_y rn(1-2\rho_x^{(t)})^4}{6\alpha} \right\} \tag{4.23}$$

provided that

$$\left(\frac{\sqrt{\rho_y r}(1-2\rho_x^{(t)})^2}{\sqrt{3}\alpha^{3/4}} \right)^4 < \frac{3}{2}. \tag{4.24}$$

If

$$\hbar(\rho_x^{(t)}) + r\hbar(\rho_y) - \frac{\rho_y r(1-2\rho_x^{(t)})^4}{6\alpha} < 0, \tag{4.25}$$

then $P_Y^{**} \to 1$ as $n \to \infty$ and $p \to \infty$. Hence, the minimum value of ρ_y (denoted as ρ_y'), such that (4.25) holds, defines the confidence interval of the number of errors in F_Y in the next state. Apparently, for a given $\rho_x^{(t)} \in [0, 0.5)$, ρ_y' can be numerically solved. Let ρ_y^* be the intersection of the line

$$L_{12} : y = \frac{\rho_y(1-2\rho_x^{(t)})^4}{6\alpha} - \frac{\hbar(\rho_x^{(t)})}{r} \tag{4.26}$$

and the curve

$$C_{12} : y = \hbar(\rho_y). \tag{4.27}$$

Then

$$\rho_y' = \rho_y^* + \varepsilon$$

where ε is an arbitrarily small positive number. According to the feature of L_{12} and C_{12}, the following corollary can be obtained.

Corollary 4.1 *For the second order BAM, $\rho_{x1}^{(t)} < \rho_{x2}^{(t)} < 0.5$ implies that $\rho'_{y1} < \rho'_{y2}$.*

According to Corollary 4.1, as $n \to \infty$ and $p \to \infty$, for every library pair (X_h, Y_h) and every $X^{(t)}$ such that $d(X_h, X^{(t)}) \leq \rho_x^{(t)} n$, the probability that the number of errors in F_Y in the next state is less than $\rho'_y p$ tends to one. We can restate the above statement as:

Corollary 4.2 *For the second order BAM, as $n \to \infty$ and $p \to \infty$, for every library pair (X_h, Y_h) and every $X^{(t)}$ such that $d(X_h, X^{(t)}) \leq \rho_x^{(t)} n$, the probability that $d(Y_h, Y^{(t+1)}) < \rho'_y p$ tends to one, where $\rho'_y = \rho^*_y + \varepsilon$, ρ^*_y is the intersection of L_{12} and C_{12}, and ε is an arbitrary small positive number.*

The above corollary means that given the number of errors in F_X in the present state being less than or equal to $\rho_x^{(t)} n$, the number of errors in F_Y in the next state is less than or equal to $\rho'_y p$ (denoted as $\rho_y^{(t+1)} p$). Note that $d(Y_h, Y^{(t+1)}) < \rho'_y p$ implies $d(Y_h, Y^{(t+1)}) \leq \rho'_y p$.

Similarly, we can easily get the following theorem.

Theorem 4.2 *For the second order BAM, for large p and n,*

$$P_X^{**} \geq 1 - \exp\left\{ p\hbar(\rho_y^{(t+1)}) + \log \alpha (\frac{p}{r})^2 + \frac{p}{r}\hbar(\rho_x) - \frac{\rho_x r p (1 - 2\rho_y^{(t+1)})^4}{6\alpha} \right\} \tag{4.28}$$

provided that

$$\left(\frac{\sqrt{\rho_x} r (1 - 2\rho_y^{(t+1)})^2}{\sqrt{3}\alpha^{3/4}} \right)^4 < \frac{3}{2}. \tag{4.29}$$

Let ρ'_x be the minimum value of ρ_x such that the right hand side of (4.28) tends to one. Also, one can find ρ'_x by considering the intersection of the line

$$L_{22} : y = \frac{\rho_x r^2(1 - 2\rho_y^{(t+1)})^4}{6\alpha} - r\hbar(\rho_y^{(t+1)}) \tag{4.30}$$

and the curve

$$C_{22} : y = \hbar(\rho_x). \tag{4.31}$$

Hence, Corollary 4.3 and Corollary 4.4 are obtained.

Corollary 4.3 *For the second order BAM, $\rho_{y1}^{(t+1)} < \rho_{y2}^{(t+1)} < 0.5$ implies that $\rho'_{x1} < \rho'_{x2}$.*

Corollary 4.4 *For the second order BAM, as $n \to \infty$ and $p \to \infty$, for every library pair (X_h, Y_h) and every $Y^{(t+1)}$, such that $d(Y_h, Y^{(t+1)}) \le \rho_y^{(t+1)} p$, the probability that $d(X_h, X^{(t+1)}) < \rho'_x n$ tends to one, where $\rho'_x = \rho^*_x + \varepsilon$, ρ^*_x is the intersection of L_{22} and C_{22}, and ε is an arbitrarily small positive number.*

It means that given the number of errors in F_Y in the present state being less than or equal to $\rho_y^{(t+1)} p$, the fraction of errors in F_X in the next state is less than or equal to $\rho'_x n$, denoted as $\rho_x^{(t+1)} n$. Note that $d(X_h, X^{(t+1)}) < \rho'_x n$ implies $d(X_h, X^{(t+1)}) \le \rho'_x n$.

By iteratively solving ρ'_x and ρ'_y, we can construct two sequences of $\rho_x^{(t)}$ and $\rho_y^{(t)}$. These sequences define the statistical dynamics about the confidence interval of the number of errors. Similar to Section 3.4 and 3.5, we can use the above dynamics to estimate the lower bound of the memory capacity, the lower bound of the attraction basin ($\rho_{maxinit}$), and the upper bound of the number of errors in the retrieval pairs (ρ_x^f and ρ_y^f).

By considering (4.23) and (4.28), it is not difficult to see the following relationship between ρ_x^f and ρ_y^f in the second order BAM.

Corollary 4.5 *For the second order BAM, if the sequences $(\rho_x^{(t)}, \rho_y^{(t)})$ converge to small values, then*

$$\rho_y^f \approx r\rho_x^f \, . \tag{4.32}$$

Similarly, from (4.23) and Corollary 4.5, we can easily obtain Corollary 4.6 which can be used to directly estimate ρ_x^f instead of solving the dynamics exactly.

Corollary 4.6 *For the second order BAM, if the sequences $(\rho_x^{(t)}, \rho_y^{(t)})$ converge to small values, then*

$$\rho_x^f \approx \exp\left\{-\frac{r^2}{(1+r^2)6\alpha} + 1 - \frac{r^2}{1+r^2}\log r\right\} , \tag{4.33}$$

$$\rho_y^f \approx \exp\left\{-\frac{r^2}{(1+r^2)6\alpha} + 1 + \frac{1}{1+r^2}\log r\right\} . \tag{4.34}$$

Remark: During solving ρ_x' and ρ_y', we should check whether both (4.24) and (4.29) are satisfied. However, if $\rho_x' << \alpha$ and $\rho_y' << \alpha$, the conditions (4.24) and (4.29) will automatically hold. Especially, when ρ_x' and ρ_y', respectively, are near to ρ_x^f and ρ_y^f, the conditions (4.24) and (4.29) will always hold (see Corollary 4.6).

4.4 Numerical Results

Numerical Example a:

Based on Corollary 4.2 and Corollary 4.4, we will first study the lower bound of the memory capacity of the second order BAM for different values of r. For a given r, let α_r be the largest value of α such that the sequences $(\rho_x^{(t)}, \rho_y^{(t)})$ converge to small values (ρ_x^f, ρ_y^f). Then $\alpha_r n^2$ can be considered as a lower bound of the memory capacity. The result is summarized in Table 4.1. From the table, the lower bound initially increases with r. Up to $r = 5$, the lower bound starts to decrease with r. Also, there is a symmetrical property about α_r. That is

$$\alpha_{\frac{1}{r}} \approx \frac{\alpha_r}{r^2}.$$

The above means that interchanging p and n does not affect the overall estimated lower bound of the memory capacity.

Table 4.1 The lower bound of the memory capacity of the second order BAM at different values of r.

r	α_r
10	0.0211
5	0.0226
2	0.0204
1	0.0128
0.5	0.00510
0.2	0.00090
0.1	0.00021

Numerical Example b:

Based on Corollary 4.2 and Corollary 4.4, we use the statistical dynamics to estimate the lower bound of the attraction basin and the upper bound of the number of errors in the retrieval pairs. For a given α, let $\rho_{maxinit,r}$ be the largest values of $\rho_x^{(0)}$ such that the sequences $(\rho_x^{(t)}, \rho_y^{(t)})$ converge to small (ρ_x^f, ρ_y^f). Figure 4.1 summarizes the lower bound of the attraction basin at $r = 1, 2, 5, 10$. The cases of $r = \frac{1}{2}, \frac{1}{5}, \frac{1}{10}$ are shown in Figure 4.2 to Figure 4.4. Note that we do not show the cases of $r = \frac{1}{2}$, $\frac{1}{5}$, and $\frac{1}{10}$ in the same figure. It is because in such cases the ranges of α's are very different.

From Figure 4.1, for $r = 1, 2, 5, 10$, the lower bound of the attraction basin initially increases as α decreases. But, as α further decreases, the lower bound becomes decreasing. From Figure 4.2 to Figure 4.4 ($r = \frac{1}{2}, \frac{1}{5}, \frac{1}{10}$), as α decreases, the lower bound decreases. This unnatural trend is due to the constraints of the conditions (4.24) and (4.29), which limit our searching range of $(\rho_x^{(t)}, \rho_y^{(t)})$ during numerically solving the dynamics. However, it is rational to accept the claim that for a smaller α a larger attraction basin should be obtained. Hence, we can take the maximum point in the figures as the lower bound of the attraction basin for small α. Table 4.2 summaries the above claim for small α. From the figures and table, the lower bound of the attraction basin is the best when $r = 1$.

Table 4.2 The lower bound of the attraction basin of the second order BAM at different values of r for small α for small α.

r	$\rho_{maxinit,r}$
10	0.00478 ($\alpha < 0.00641$)
5	0.00504 ($\alpha < 0.00701$)
2	0.00546 ($\alpha < 0.00814$)
1	0.00587 ($\alpha < 0.00933$)
$\frac{1}{2}$	0.00542 ($\alpha < 0.00487$)
$\frac{1}{5}$	0.00245 ($\alpha < 0.000866$)
$\frac{1}{10}$	0.00107 ($\alpha < 0.000193$)

Also, (ρ_x^f, ρ_y^f), which reflect the upper bounds of the number of errors in the retrieval pairs, is recorded in Figure 4.5 and Figure 4.6. In the figures, we do not show the cases of $r = \frac{1}{2}$, $\frac{1}{5}$, and $\frac{1}{10}$. It is because such cases (with suitable change in α, ρ_x^f, and ρ_y^f) are very similar to those of $r = 2$, 5, and 10, respectively (see Corollary 4.6). [3] From the two figures, the upper bound exponentially decrease as α decreases. Also, ρ_y^f is approximately equal to to $r\rho_x^f$ at all cases. We easily verify that ρ_x^f and ρ_y^f, estimated from the dynamics (Corollary 4.2 and Corollary 4.4) in Figure 4.5 and Figure 4.6, agree with Corollary 4.5 and Corollary 4.6.

[3]Changing α in Corollary 4.6 to $\frac{\alpha}{r^2}$ and changing r to $\frac{1}{r}$ will get the results of $r = \frac{1}{2}$, $\frac{1}{5}$, and $\frac{1}{10}$.

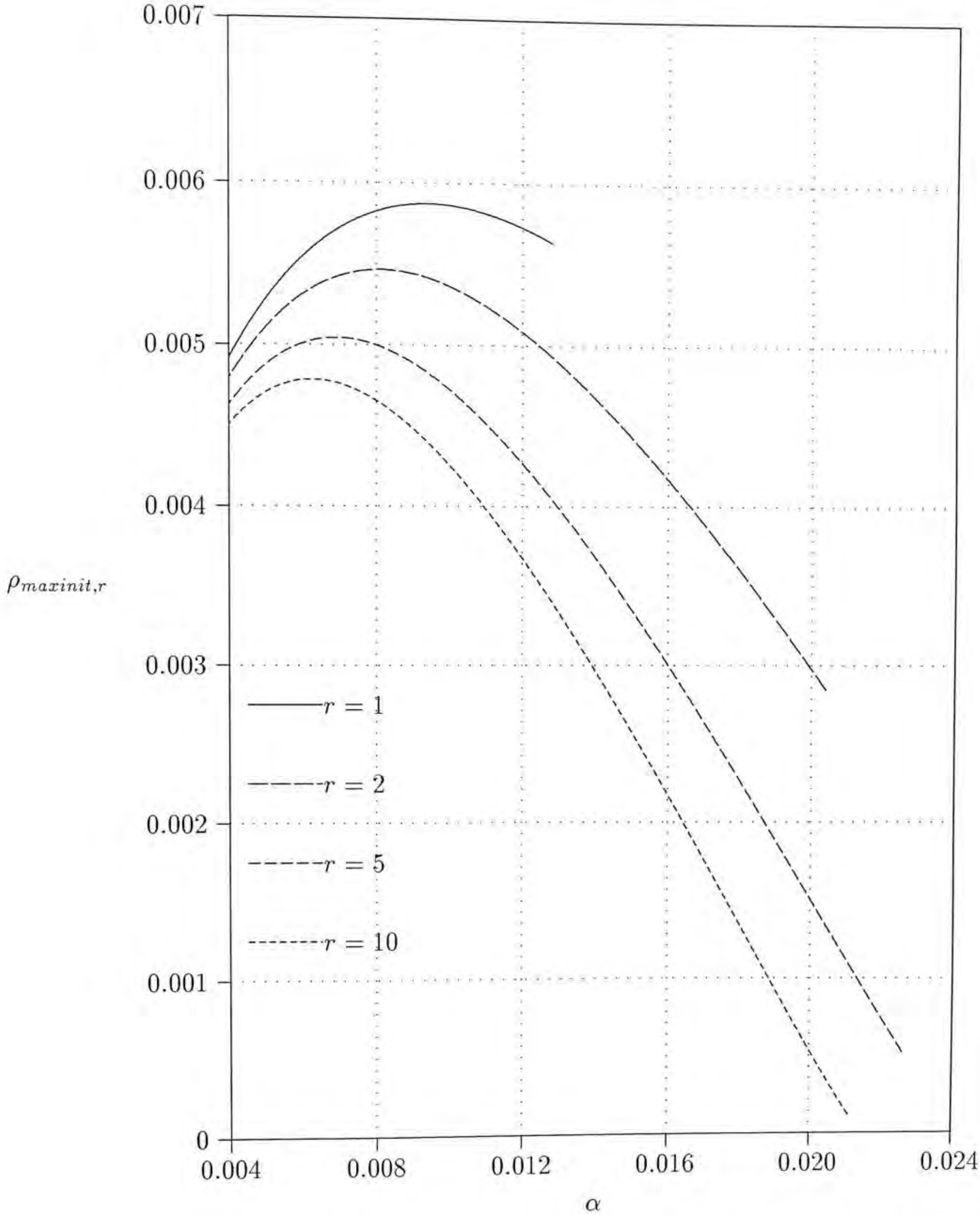


Figure 4.1 The lower bound of the attraction basin for the second order BAM where $r = 1, 2, 5, 10$.

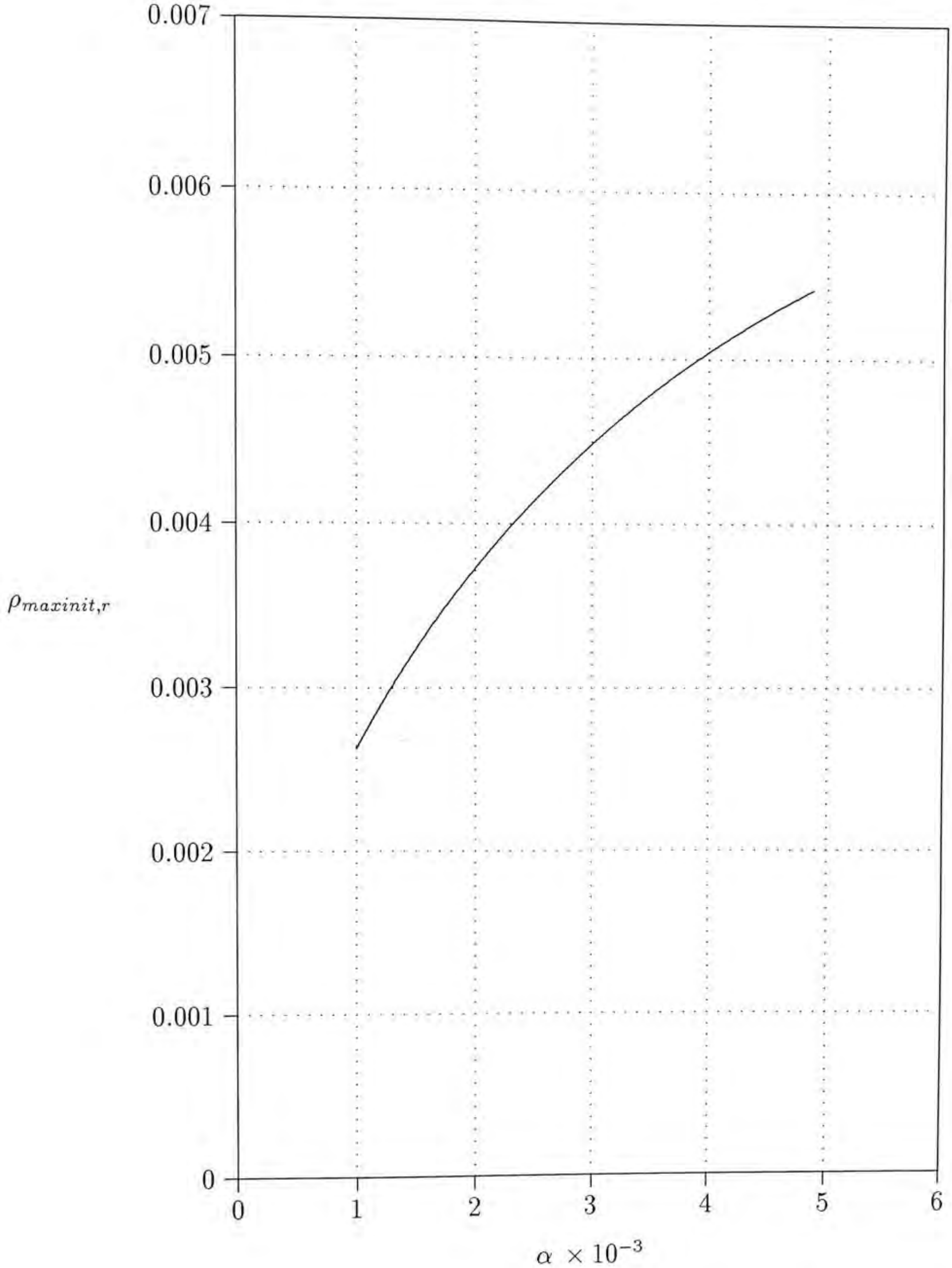


Figure 4.2 The lower bound of the attraction basin for the second order BAM where $r = \frac{1}{2}$.

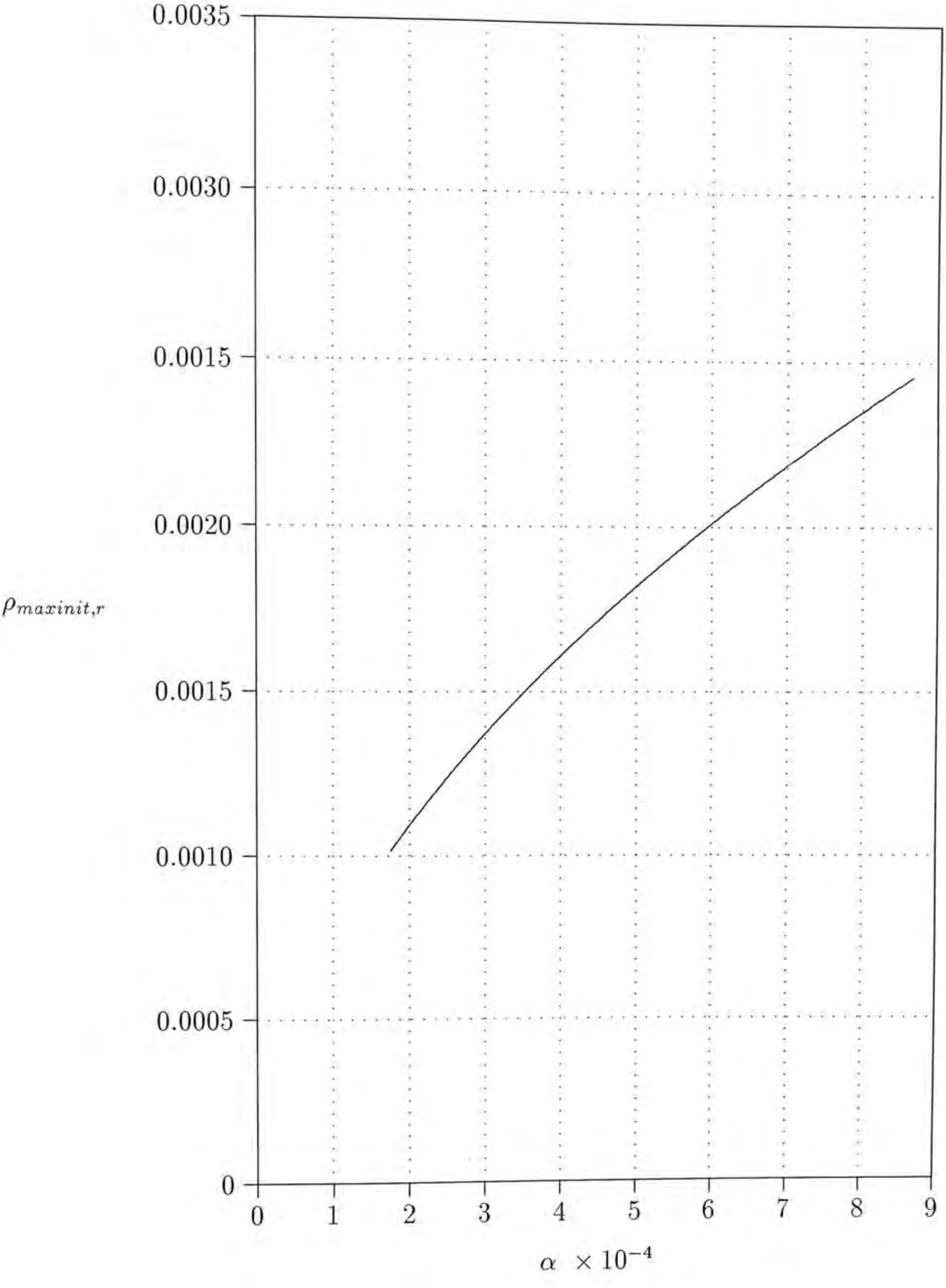


Figure 4.3 The lower bound of the attraction basin for the second order BAM where $r = \frac{1}{5}$.

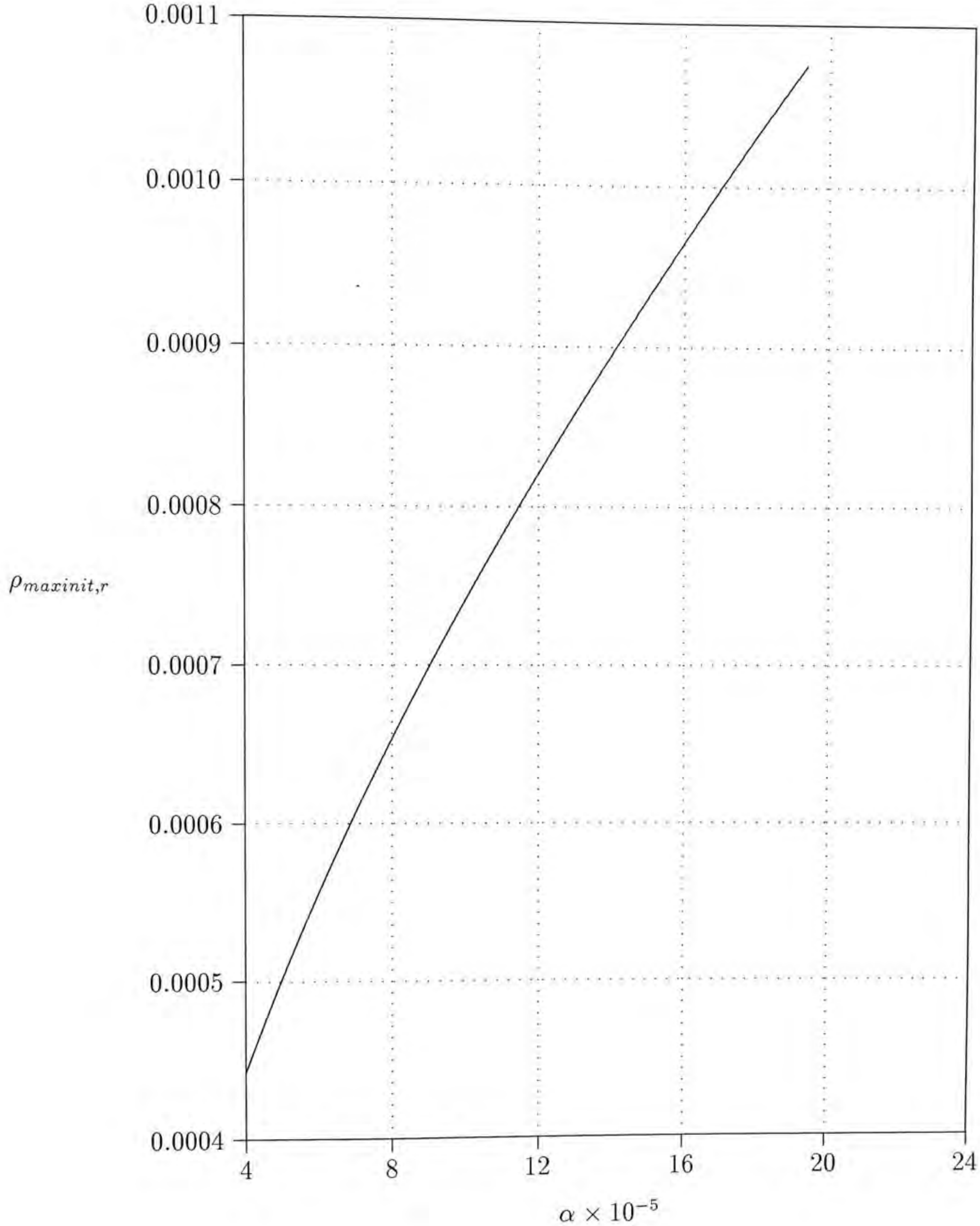


Figure 4.4 The lower bound of the attraction basin for the second order BAM where $r = \frac{1}{10}$.

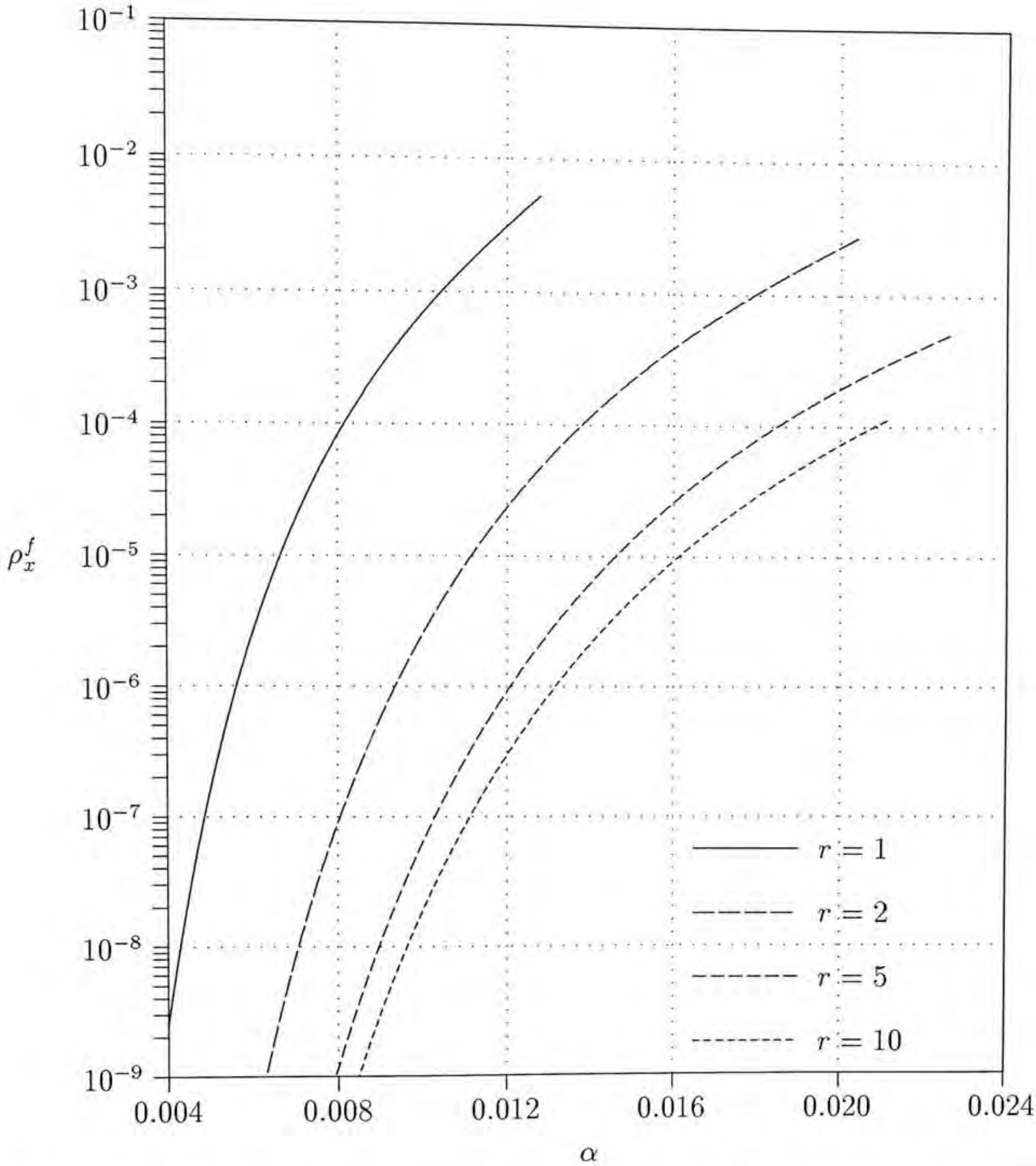


Figure 4.5 The upper bound of the number of errors in the layer F_X in the retrieval pairs for the second order BAM where $r = 1, 2, 5, 10$.

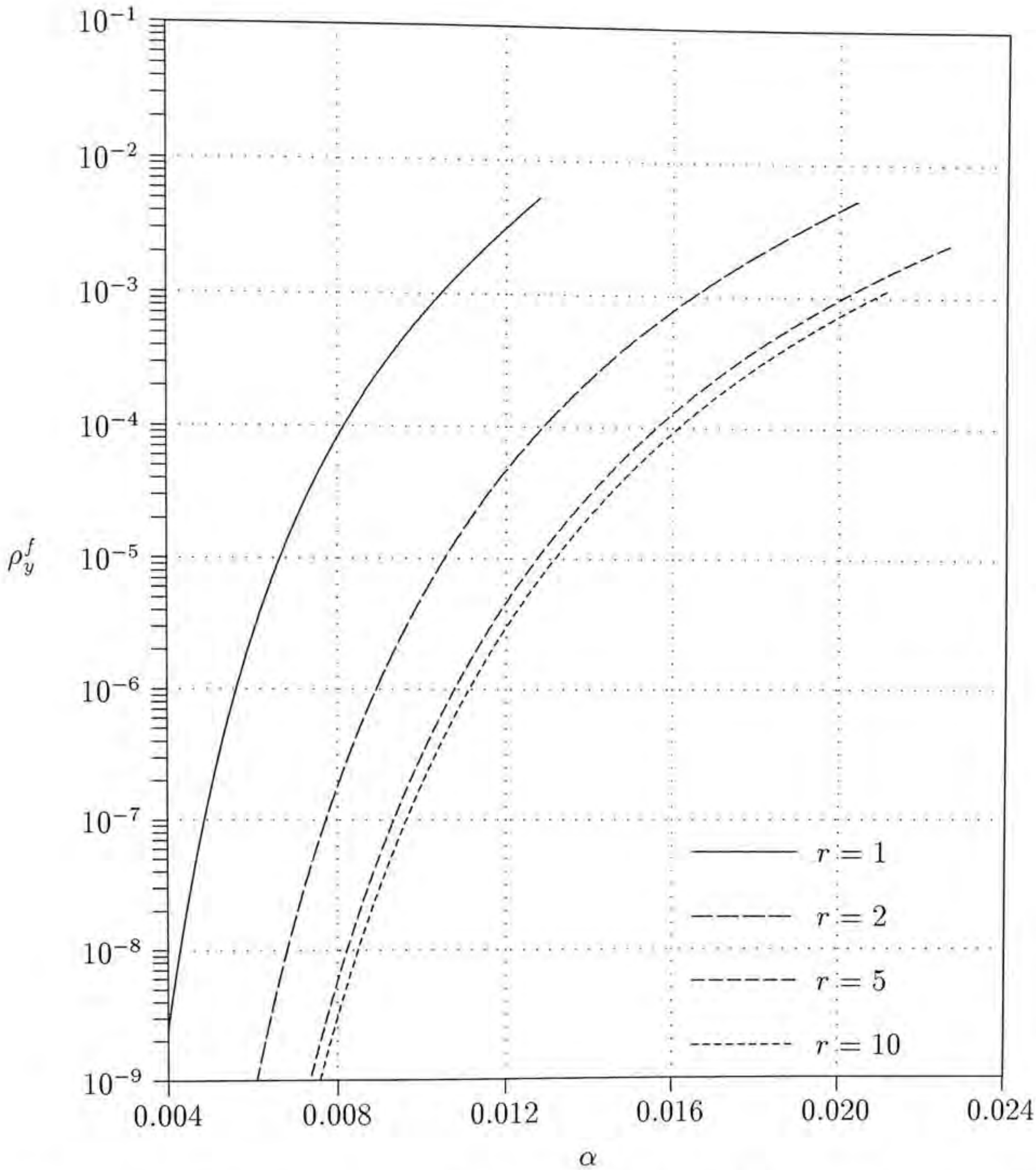


Figure 4.6 The upper bound of the number of errors in the layer F_Y in the retrieval pairs for the second order BAM where $r = 1, 2, 5, 10$.

4.5 Extension to higher order BAM

Although we are mainly concerned with the properties of the second order BAM, we can apply a similar method to analyze the higher order BAM. Here, the only change of assumption is

$$m = \alpha n^q \tag{4.35}$$

where q is a positive integer. In the case of the general q-order BAM, the connections from F_X to F_Y are

$$u_{k,i_1,i_2,\cdots,i_q} = \sum_{h=1}^{m} y_{kh}\, x_{i_1h}\, x_{i_2h} \cdots x_{i_qh} \tag{4.36}$$

where $k = 1, \cdots, p$, $i_1 = 1, \cdots, n$, $i_2 = 1, \cdots, n$, $\cdots$, and $i_q = 1, \cdots, n$. The connections from F_Y to F_X are

$$v_{j,l_1,l_2,\cdots,l_q} = \sum_{h=1}^{m} x_{jh}\, y_{l_1h}\, y_{l_2h} \cdots y_{l_qh} \tag{4.37}$$

where $j = 1, \cdots, n$, $l_1 = 1, \cdots, p$, $l_2 = 1, \cdots, p$, $\cdots$, and $l_q = 1, \cdots, p$. The corresponding recalling rules are:

$$y_k^{(t+1)} = \operatorname{sgn}\left(\sum_{i_q=1,i_{q-1}=1,\cdots,i_1=1}^{n} u_{k,i_1,i_2,\cdots,i_q}\, x_{i_1}^{(t)}\, x_{i_2}^{(t)} \cdots x_{i_q}^{(t)} \right) \tag{4.38}$$

$$x_j^{(t+1)} = \operatorname{sgn}\left(\sum_{l_q=1,l_{q-1}=1,\cdots,l_1=1}^{p} v_{j,l_1,l_2,\cdots,l_q}\, y_{l_1}^{(t+1)}\, y_{l_2}^{(t+1)} \cdots y_{l_q}^{(t+1)} \right) . \tag{4.39}$$

We can obtain the similar results for the general q-order BAM based on Lemma 4.3.

Lemma 4.3 *Let ξ_i's be ± 1 equiprobable independent random variables, $S'_n = \frac{\sum_{i=1}^{n} \xi_i}{\sqrt{n}}$, and n is a positive integer, as $n \to \infty$*

$$E\left[(S'_n)^{2q}\right] \to \frac{(2q)!}{2^q q!} . \tag{4.40}$$

Proof of Lemma 4.3

As $n \to \infty$, S'_n tends to standard normal. Since the $2q$-th moment of a standard normal random variable [69] is

$$1 \cdot 3 \cdots (2q-1) ,$$

the $2q$-th moment of S'_n

$$\begin{aligned} E\left[(S'_n)^{2q}\right] &\to 1 \cdot 3 \cdots (2q-1) \\ &= \frac{(2q)!}{2^q q!} . \end{aligned}$$

Hence, the proof is completed. □

From Lemma 4.3 and the results in the next section, we will obtain the following four corollaries about the general q-order BAM.

Corollary 4.7 *For the q-order BAM as $n \to \infty$ and $p \to \infty$, for every library pair (X_h, Y_h) and every $X^{(t)}$ such that $d(X_h, X^{(t)}) \le \rho_x^{(t)} n$, the probability that $d(Y_h, Y^{(t+1)}) < \rho_y' p$ tends to one, provided that*

$$\left(\frac{\sqrt{\rho_y r}(1 - 2\rho_x^{(t)})^q}{\sqrt{\lambda_q} \alpha^{(q+1)/(2q)}} \right)^{2q} < \frac{\lambda_q}{2},$$

where

$$\lambda_q = \frac{(2q)!}{2^q q!},$$

$\rho_y' = \rho_y^* + \varepsilon$, ρ_y^* is the intersection of L_{1q} and C_{1q}

$$\begin{aligned} L_{1q} &: \quad y = \frac{\rho_y (1 - 2\rho_x^{(t)})^{2q}}{2\lambda_q \alpha} - \frac{h(\rho_x^{(t)})}{r} \\ C_{1q} &: \quad y = h(\rho_y) , \end{aligned}$$

and ε is an arbitrarily small positive number.

Corollary 4.8 *For the q-order BAM, as $n \to \infty$ and $p \to \infty$, for every library pair (X_h, Y_h) and every $Y^{(t+1)}$ such that $d(Y_h, Y^{(t+1)}) \leq \rho_y^{(t+1)} p$, the probability that $d(X_h, X^{(t+1)}) < \rho_x' n$ tends to one, provided that*

$$\left(\frac{\sqrt{\rho_x} r^{q/2} (1 - 2\rho_y^{(t+1)})^q}{\sqrt{\lambda_q} \alpha^{(q+1)/(2q)}} \right)^{2q} < \frac{\lambda_q}{2},$$

where $\rho_x' = \rho_x^* + \varepsilon$, ρ_x^* is the intersection of L_{2q} and C_{2q}

$$\begin{aligned} L_{2q} &: \quad y = \frac{\rho_x r^q (1 - 2\rho_y^{(t+1)})^{2q}}{2\lambda_q \alpha} - r\hbar(\rho_y^{(t+1)}) \\ C_{2q} &: \quad y = \hbar(\rho_x), \end{aligned}$$

and ε is an arbitrarily small positive number.

Corollary 4.9 *For the q-order BAM, if the sequences $(\rho_x^{(t)}, \rho_y^{(t)})$ converge to small values, then*

$$\rho_y^f \approx r^{q-1} \rho_x^f . \tag{4.41}$$

Corollary 4.10 *For the q-order BAM, if the sequences $(\rho_x^{(t)}, \rho_y^{(t)})$ converge to small values, then*

$$\rho_x^f \approx \exp\left\{ -\frac{r^q}{(1+r^q)2\lambda_q \alpha} + 1 - \frac{r^q}{1+r^q} \log r^{q-1} \right\}, \tag{4.42}$$

$$\rho_y^f \approx \exp\left\{ -\frac{r^q}{(1+r^q)2\lambda_q \alpha} + 1 + \frac{1}{1+r^q} \log r^{q-1} \right\} \tag{4.43}$$

We can use Corollary 4.7 and Corollary 4.8 to construct the statistical dynamics of the confidence interval of the number of errors for the q-order BAM. Also, the corresponding memory capacity, attraction basin, and number of errors in the retrieval pairs can be estimated. Moreover, Corollary 4.10 can be used directly to estimate the number of errors in the retrieval pairs without solving the dynamics numerically.

4.6 Verification of the conditions of Newman's Lemma

In this section, we show under what conditions the random variable χ

$$\chi = \frac{(\sum_{k=1}^{\rho_y p} y_k)(\sum_{j=1}^{n} x_j)^q}{\sqrt{\rho_y p \lambda_q n^q}},$$

satisfies the two conditions (4.10) and (4.11) in Newman's Lemma, where y_k's and x_j's are ± 1 equiprobable independent random variables. Clearly, χ is symmetric and

$$E[\chi] = 0. \tag{4.44}$$

Also, from Lemma 4.3

$$\mathrm{Var}(\chi) = E\left[\chi^2\right] = 1\,. \tag{4.45}$$

Hence, (4.10) is satisfied.

To check whether χ satisfies (4.11), we use an existing result about the sum of ± 1 equiprobable independent random variables [68].

Lemma 4.4 *Let ξ_i's be ± 1 equiprobable independent random variables. Then, for $z > 0$ and large n,*

$$E\left[\frac{|\sum_{i=1}^{n} \xi_i|^z}{n^{z/2}}\right] \leq 2^{z/2+1} \pi^{-1/2} \Gamma(\frac{z+1}{2}) \tag{4.46}$$

where $\Gamma(a)$ is the gamma function

$$\Gamma(a) = \int_0^\infty x^{a-1} e^{-x} dx \tag{4.47}$$

The above lemma is part of Lemma A.6 in [68].[4]

[4] In [68], a lot of properties about the sum of ± 1 equiprobable independent random variables have been explored. But the author use these properties to study the memory capacity of the higher order Hopfield network under the condition that errors **are not allowed** in the retrieval patterns. In fact, these properties can be used in many cases. It is because most problems in the field of associative memories can be re-formulated as problems related to the sum of ± 1 equiprobable independent random variables. For example, here we use the above lemma to study the dynamics of the second order BAM.

Based on Lemma 4.4, for $z > 0$, and large n and p,

$$E[|\chi|^z] \leq 2^{z/2+1}\pi^{-1/2}\Gamma(\frac{z+1}{2})\,2^{qz/2+1}\pi^{-1/2}\Gamma(\frac{qz+1}{2})\lambda_q^{-z/2} \tag{4.48}$$

Let $z = \omega k$, where k is a positive integer and $\omega > 0$. Then,

$$\frac{t_o^k}{k!}E[|\chi^z|] \leq C_o\frac{t_o^k 2^{((q+1)\omega k)/2}}{k!}\Gamma(\frac{\omega k+1}{2})\Gamma(\frac{q\omega k+1}{2})\lambda_q^{-q\omega k/2}, \tag{4.49}$$

where C_o is an positive constant. For fixed ω and large k,

$$k! \approx \sqrt{2\pi}e^{-k}k^{k+1/2},$$

$$\Gamma(\frac{\omega k+1}{2}) \approx \sqrt{2\pi}e^{-(\omega k+1)/2}(\frac{\omega k+1}{2})^{(\omega k+1)/2-1/2},$$

and

$$\Gamma(\frac{q\omega k+1}{2}) \approx \sqrt{2\pi}e^{-(q\omega k+1)/2}(\frac{q\omega k+1}{2})^{(q\omega k+1)/2-1/2}.$$

Hence, for fixed ω and large k,

$$\frac{t_o^k}{k!}E\left[|\chi|^{\omega k}\right] \leq C_o e^{-(((q+1)\omega)/2-1)k}(t_o 2^{q\omega/2}\lambda_q^{-\omega/2})^k(\omega k+1)^{(\omega k)/2}(\frac{q\omega k+1}{2})^{(q\omega k)/2}k^{-k-1/2}. \tag{4.50}$$

For large k, the k-th term of the sum

$$S = \sum_{k=0}^{\infty}\frac{t_o^k}{k!}E\left[|\chi|^{\omega k}\right]$$

hence decreases exponentially provided that $\omega = \frac{2}{q+1}$ and $t_o < 2^{-(q)/(q+1)}\lambda_q^{1/(q+1)}$. As S converges to

$$E\left[\exp\left\{t_o\,|\chi|^{2/(q+1)}\right\}\right],$$

it follows that

$$\limsup_{n\to\infty} E\left[\exp\left\{t_o\,|\chi|^{2/(q+1)}\right\}\right] < \infty.$$

4.7 Chapter Summary

In this chapter, we have studied several properties of the second order BAM. The properties are the stability and the statistical properties.

- We have given an an example to show that the second order BAM is not a stable model. That is, its state may converge to a limit cycle.

- When a small number of errors in the retrieval pairs are allowed, we have followed the methodology, presented in Chapter 3, to estimate the statistical dynamics of the number of errors for the second order BAM. Hence, we can estimate the memory capacity, the attraction basin, and the number of errors in the retrieval pairs for the second order BAM.

- The memory capacity can grow as far as $\alpha_r n^2$, which depends on the ratio of the dimensions: $r = \frac{p}{n}$ (see Table 4.1).

- The lower bound of the attraction basin is also a function of α and r (see Figure 4.1 to Figure 4.4, and Table 4.2).

- The number of errors in the retrieval pairs is bounded by $O(\exp\{-\frac{r^2}{6(1+r^2)\alpha} + \frac{r^2}{1+r^2}\log r\}n)$, where the number of library pairs is αn^2 (see Figure 4.5 and Figure 4.6).

- Also, we have briefly explained how to extend the results to the general higher order BAM.

Chapter 5

Enhancement of BAM

In this chapter, we focus on the modified versions of BAM. The task of these modified versions is to enhance the recall performance of BAM. In general, there are two approaches of the modified versions: 'change of encoding method' and 'change of the topology'. We adapt the approach of 'the change of encoding method' to develop four new encoding algorithms, identified as householder encoding algorithm (HCA), enhanced householder encoding algorithm (EHCA), bidirectional learning (BL), and adaptive Ho-Kashyap bidirectional learning (AHKBL) for the BAM. Simulation results show that these four encoding methods can greatly improve the memory capacity of BAM. In particular, the HCA, EHCA, and AHKBL greatly improve the error correction capability. The properties of these four learning rules, such as the memory capacity, the error correction capability, the convergent conditions, the ease of hardware implementation, and the learning speed are also addressed. Additionally, we will also made an empirical comparison on the properties among the four learning rules and some other existing learning rules.

5.1 Background

As mentioned in many articles [32]-[49], the memory capacity and the error correction capability of BAM under the Kosko's encoding is poor even the number of library pairs is very small. Many modifications of BAM, which change either the encoding method or the topology, are available [32]-[49]. The task of the modifications is to improve the recall performance such that the noisy input can correctly recall the desired library pair.

To achieve the task, the first condition is that all the library pairs should be stored as fixed points:

$$\mathrm{sgn}(WX_h) = Y_h, \tag{5.1}$$

$$\mathrm{sgn}(W^TY_h) = X_h \tag{5.2}$$

for all $h = 1, \cdots, m$. Otherwise, the library pairs will never be recalled. Note that (5.1) and (5.2) cannot guarantee that the library pair (X_h, Y_h) is recalled when a perfect library pattern X_h is given. For example, all bipolar pairs are stored as fixed points when the connection matrix is a zero matrix. It is because the sign operator is

$$\mathrm{sgn}(x) = \begin{cases} +1 & x > 0 \\ -1 & x < 0 \\ \text{state unchanged} & x = 0 \end{cases}.$$

The state of a neuron will not change when the weighted sum of its input is zero. Hence, given any initial state, the state is never changed when a zero connection matrix is used. We call this kind of BAM as a "trivial system" in which every possible state is a stable state. Hence, we change the definition of sign operator as:

$$\mathrm{sgn}(x) = \begin{cases} +1 & x > 0 \\ -1 & x < 0 \\ \text{state unchanged (during recalling process)} & x = 0 \\ 0 \text{ (during training)} & x = 0 \end{cases}.$$

With such definition, if the encoding methods can find out a connection matrix W such that

$$y_{jh}(\sum_{i'}^{n} w_{ji'}x_{i'h}) > 0 \tag{5.3}$$

$$x_{ih}(\sum_{j'}^{p} w_{j'i}y_{j'h}) > 0 \tag{5.4}$$

for $i = 1, \cdots, n$ and $j = 1, \cdots, p$, then each perfect library pattern X_h can recall the corresponding pair (X_h, Y_h). From the energy point of view, the new definition leads to the following lemma.

Lemma 5.1 *With a connection matrix W, a library pair (X_h, Y_h) satisfying the following equations:*

$$y_{jh}(\sum_{i'}^{n} w_{ji'}x_{i'h}) \;>\; 0 \tag{5.5}$$

$$x_{ih}(\sum_{j'}^{p} w_{j'i}y_{j'h}) \;>\; 0 \tag{5.6}$$

for $i = 1, \cdots, n$ and $j = 1, \cdots, p$, if and only if the library pair (X_h, Y_h) is an isolated local minimum of the energy function.

Proof Lemma 5.1

Let (X, Y) be one of the neighborhood pairs of (X_h, Y_h). The difference between their energies is

$$-Y_h^T\, W\, X_h \;+ Y^T\, W\, X\,.$$

If the different element is in the i-th position of X_h and X, then the difference of their energies

$$-2x_{ih}(\sum_{j'=1}^{p} w_{j'i}y_{j'h})\;\;.$$

If the different element is in the jth position of Y_h and Y, then the difference of their energies

$$-2y_{jh}(\sum_{i'=1}^{n} w_{ji'}x_{i'h})\;\;.$$

From (5.5) and (5.6), the energy of (X_h, Y_h) is less than that of any one of its neighborhood pair.

Conversely, if the energy of (X_h, Y_h) is less than zero, we can easily get

$$-2x_{ih}(\sum_{j'}^{p} w_{j'i}y_{j'h}) \;<\; 0$$

$$-2y_{jh}(\sum_{i'}^{n} w_{ji'}x_{i'h}) \;<\; 0\,,$$

for all $i = 1, \cdots, n$ and $j = 1, \cdots, p$. Hence, (5.5) and (5.6) can be immediately obtained. The proof is completed. □

If (5.3) and (5.4) is changed to

$$\begin{aligned} y_{jh}(\sum_{i'}^{n} w_{ji'} x_{i'h}) &\geq 0 \\ x_{ih}(\sum_{j'}^{p} w_{j'i} y_{j'h}) &\geq 0 \end{aligned}$$

then the following lemma is obtained.

Lemma 5.2 *With a connection matrix W, a library pair (X_h, Y_h) satisfying the following equations:*

$$y_{jh}(\sum_{i'}^{n} w_{ji'} x_{i'h}) \geq 0 \tag{5.7}$$

$$x_{ih}(\sum_{j'}^{p} w_{j'i} y_{j'h}) \geq 0 \tag{5.8}$$

for $i = 1, \cdots, n$ and $j = 1, \cdots, p$, if and only if the library pair (X_h, Y_h) is a local minimum of the energy function.

It should be emphasized that if we use (5.7) and (5.8) to construct the connection matrix, the 'trivial solution' (i.e. the zero matrix) is one of the solutions. To avoid this ambiguity, we define the task of the encoding methods is to construct the connection matrix such that each library pair is **an isolated local minimum** of the energy function.

We will first give a general review on the modifications of the BAM in Section 5.2. The details of our proposed four encoding methods are presented in Section 5.3 to 5.6. The properties of the four encoding methods will also be discussed in the corresponding section. We also give an empirical comparison on the recall performance between our methods and some other existing methods in Section 5.7. Finally, a conclusion is given in Section 5.8.

5.2 Review on Modifications of BAM

Up to now, many modifications are available regarding the BAM [34]-[53]. In general, these modifications apply one or both approaches: 'change of the encoding method' and 'change of the topology'. They often make improvements on the recall performance with extra cost, e.g., increased computation or implementation complexity.

The efficiency of a modification can be in terms of the information ratio[45]:

$$\text{information ratio} = \frac{\text{number of library pairs correctly stored}}{\text{number of connections}}. \tag{5.9}$$

Upon simplification, if we set $n = p$ in the BAM, then the information ratio of Kosko's encoding is $\frac{1}{4n \log n}$.

5.2.1 Change of the encoding method

In the case of 'change of the encoding method', other algorithms (instead of the Kosko's encoding scheme) are employed to construct the connection matrix such that the recall performance is improved. Note that the recall equations are unchanged. Hence, the stabilization can almost be maintained during recall.

One technique [41] uses a multiple training concept which improves the memory capacity, but does not guarantee the recall of all library pairs. In the multiple training, the initial connection matrix is formed by Kosko's encoding scheme. The library pairs are then repeatedly and sequentially presented to the BAM. If the presented library pair is a fixed point, then the connection matrix is not changed. Otherwise, the connection matrix is updated as:

$$W^{new} = W^{old} + Y_{h'} X_{h'}^T. \tag{5.10}$$

where $(X_{h'}, Y_{h'})$ is the currently presented library pair. Recently, Wang [42] *et al.* have formulated the multiple training to solve a set of inequalities and suggested the use of a linear programming technique. The multiple training guarantees that a particular library pair is stored as a fixed point. However, the memory capacity with it is still very small (see Section 5.7 or [42]). The hardware implementation can be accomplished in an easy way since only the integer-valued weights are involved and the connection matrix can only be changed by 1 or -1.

Spurious states, which are the undesired local minima in the energy function (or said as the undesired fixed point), are traps to use of BAM. Therefore, ensuring the library pairs to be local minima is not enough. Eliminating spurious states is also important. Srinivasan and China [40] suggested adding the unlearning concept to the multiple training. The connection matrix W has the form

$$W = \sum_{h}^{m} q_h Y_h X_h^T - \sum_{j} q_{js} Y_{js} X_{js}^T \tag{5.11}$$

where (X_h, Y_h) are the library pairs and (X_{js}, Y_{js}) are the spurious states. However, the values of q_h and q_{js} should be determined experimentally. There is still no systematic way to find q_h and q_{js}. Through computer simulations, Srinivasan and China [40] have shown that their unlearning technique can increase the memory capacity and the error correction capability.

Haines and H. Nielson [24] extended the BAM to include thresholds and then they improve the memory capacity of BAM through the use of the sparse coding.

Suppose that it is permitted to use two different matrices W_f and W_b in different directions. Then (5.1) and (5.2) can be rewritten as

$$\operatorname{sgn}(W_f X_h) = Y_h \tag{5.12}$$

$$\operatorname{sgn}(W_b Y_h) = X_h. \tag{5.13}$$

Moreover, we can further change (5.12) and (5.13) to

$$W_f F_A = F_B \tag{5.14}$$

$$W_b F_B = F_A \tag{5.15}$$

where $F_A = [X_1 \vdots \cdots \vdots X_m]$ and $F_B = [Y_1 \vdots \cdots \vdots Y_m]$. Note that (5.14) and (5.15) imply (5.12) and (5.13). Hassoun [54], Leung [32], and Hu [51] individually proposed three algorithms to solve (5.12) and (5.13). These three algorithms lead to the concept of the generalized inverse. That is,

$$W_f = F_B F_A^+ \tag{5.16}$$

$$W_b = F_A F_B^+ \tag{5.17}$$

where F_A^+ and F_B^+ are the generalized inverses of F_A and F_B respectively. In [54], Hassoun proposed to use the Ho-Kashyap encoding method with the generalized inverses, called HKDAM, for the constructions of W_f and W_b. Note that the HKDAM usually terminates within one learning cycle and leads to (5.16) and (5.17), which was not realized in [54]. It is because the probability that the rank of F_A (and F_B) is m is very high [62] and then the generalized inverses of F_A (and F_B) are uniquely defined. In [32] Leung proposed the householder encoding algorithm (HCA) to construct W_f and W_b. There are two models in the HCA: batch model and iterative model. In the batch model, the construction of the connection matrices is based on the householder transformation [70]. The iterative model HCA is an incremental learning rule in which the new library pairs can be encoded into the connection matrices based on the current connection matrices only. Also, Leung proved that the memory capacity of HCA tends to $\min(n, p)$. Recently, Hu [51] proposed the unilateral orthogonalization based BAM (UOBAM), which is similar to the iterative model HCA, to construct the connection matrices. The major difference between them is that there are two additional projection matrices which must be stored during encoding in the UOBAM (see Section 5.3.2). It means that if the UOBAM is used under an adaptive environment, we should also keep the two projection matrices. When compared with the multiple training, the approach of the generalized inverse greatly improve the error correction capability. Since there are two different connection matrices in this approach, the stabilization of BAM with this approach is unknown during recall. Since the HCA, UOBAM, and HKDAM are all related to the concept of the generalized inverse, their memory capacity and error correction capability are the same. The hardware implementation of them is relatively difficult since real-valued weights are involved. The details of HCA is presented in Section 5.3.

The disadvantages of the approach of generalized inverse is that the stabilization of BAM is unknown during recall and the number of connections is double. To overcome this problem, enhanced householder encoding algorithm (EHCA) was later introduced [34, 33]. The EHCA reduces the two connection matrices found from the generalized inverse into one. Hence, the stabilization can surely be maintained. The recall performance of EHCA is comparable to that of HCA. However, the drawbacks of EHCA are that it is a batch mode learning rule and a computation intensive learning rule. Also, the hardware implementation of EHCA is relatively difficult since real-valued weights are involved. The details of EHCA will be presented in Section 5.4.

Borrowing the idea from the perceptron rule, Leung [35, 36, 37] proposed bidirectional learning (BL) which is an iterative mode learning rule. At the same time, two similar algorithms, called optimal learning algorithm (OLA) and optimal learning scheme (OLS), are individually discovered [43, 44]. The development of OLA and OLS is based on the global minimization of a cost function. On the other hand, the BL views the encoding problem as the training of a single-layer perceptron in a bidirectional sense. The memory capacity of BL is proved to be the greatest among all encoding methods (with one connection matrix only). As only integer-valued weights are involved in the BL, the hardware implementation of BL is as simple as the case of the multiple training. Since the perceptron rule cannot locate a good decision surface [74], the BL also has weak error correction nature. Leung later [38] introduced a robust learning rule, named adaptive Ho-Kashyap bidirectional learning (AHKBL), to enhance the error correction capability. However, the AHKBL loses the feature of integer-valued weights. Note that there is only connection matrix in the BL and AHKBL. Hence, the stable property of BAM can be maintained. The details of BL and AHKBL will be presented in Section 5.5 and 5.6.

The pseudo-relaxation learning algorithm for the BAM (PRLAB) [52], which is similar to the BL and AHKBL, uses a different mathematical approach adapted from the relaxation method originally proposed in [75]. The aim of PRLAB is to accelerate the learning process. From Section 5.7, the memory capacity of PRLAB is the same as that of BL and AHKBL. The error correction capability of PRLAB is only similar to that of BL. Similar to the case of AHKBL, the PRLAB loses the feature of integer-valued weights but the learning speed of PRLAB is very fast.

Apart from the multiple training, the encoding methods mentioned here greatly improve the information ratio. Although there are two matrices in the approach of generalized inverse, the information ratio of this approach is still greater than that of

Kosko's encoding scheme. It is because the memory capacity of the approach is much greater than that of Kosko's encoding scheme.

5.2.2 Change of the topology

In the approach of 'change of the topology', the structure and recall process of BAM are changed. However, the concept of the bidirectional feedback between the two layers F_X and F_Y is maintained. Since the topology is changed in this approach, we may interpret the modifications as the new heterassociative memory models.

One technique [45, 56], the higher order BAM, is based on the higher order connection. Its statistical properties have been presented in Chapter 4. Empirically, it greatly improves the memory capacity and the error correction capability. However, the number of connections of the higher order BAM is much greater than that of the first order BAM. From the simulation results in [45], the information ratio of the higher order BAM is much poorer than that of the original BAM. As mentioned in Chapter 4, the stabilization of the higher order BAM is not guaranteed.

The concept of the layered extension is another valuable strategy since this approach is able to improve the BAM's storage capacity with a relatively small increase in complexity [48, 49]. The backpropagation learning [5] is employed to find the connections [48, 49]. However, the stable property of BAM cannot be maintained. Note that if the hidden layer in [48, 49] is removed, then the model becomes a feedback heterassociative memory with two different connection matrices. The two connection matrices are the solutions of (5.12) and (5.13). There are many methods to find them. One of the methods is the approach of the generalized inverse mentioned above. Another method is to individually use the perceptron rule to construct the two matrices [58]. However, in the above two methods, the two matrices are not the same (i.e. $W_f \neq W_b^T$) and then the stable property of BAM cannot be maintained.

The dummy augmentation encoding (DAE), as proposed by Wang [41], improves the recall performance by introducing additional neurons in the two layers. Each library pair is attached with an orthogonal pair and then a new set of library pairs is obtained. The new connection matrix can be obtained from the outer product rule based on this new set library pairs. As there is one connection matrix in the DAE, the stabilization of DAE is guaranteed during recall. In the DAE, given a set of library pairs, each library pair can be stored as a fixed point if the number of additional neurons is sufficiently large (i.e. The number of additional neurons is data-dependent.). However, it is difficult for us to comment on the recall performance of DAE if the resources (number of additional neurons) are not limited. In [39], Leung investigated the statistical memory capacity of DAE for a given number of additional neurons and then evaluate the efficiency of DAE in terms of information ratio. The statistical memory capacity of DAE is

$$\min(\frac{n(1+r_d)^2}{4\log n}, \frac{p(1+r_d)^2}{4\log p}),$$

where there are $r_d n$ and $r_d p$ additional neurons in the two layers of BAM. For simplification, we set $n = p$ in the BAM. In the DAE, the number of connections is $n^2(1+r_d)^2$. In the original BAM, the information ratio is $\frac{1}{4n\log n}$. The information ratio of DAE is also $\frac{1}{4n\log n}$. The simulation results in [45] show that the information ratio of the higher order BAM is much less than that of the original BAM. Hence, from our theoretical results in [39] and the empirical results in [45], we can expect that the information ratio of DAE is better than that of the higher order BAM. The advantage of DAE is that its recall performance depends on the number of the additional neurons.

The modified bidirectional decoding strategy (MBDS) [47] uses two additional cascade networks to improve the memory capacity. The two cascade networks force the library pairs, which are not stored as fixed points in the original BAM, to become fixed points. Since the sizes and the weights of the two cascade networks depend on the library pairs in the MBDS, it is very difficult for us to comment its efficiency. Also, the stabilization of MBDS is unknown.

Another technique [53], namely the exponential bidirectional associative memory (EBAM), uses an exponential scheme of information flow to enhance the recall performance. Its recall process is

$$Y^{(t+1)} = \operatorname{sgn}\left(\sum_{h=1}^{m} Y_h \exp(X_h^T X^{(t)})\right)$$

$$X^{(t+1)} = \operatorname{sgn}\left(\sum_{h=1}^{m} X_h \exp(Y_h^T Y^{(t+1)})\right). \tag{5.18}$$

The EBAM was later individually discovered as the modified bidirectional associative memory (MBAM) in [50]. In the EBAM, there is no distributed connection matrix and the library pairs are directly encoded into the model. In my opinion, once we use the approach of direct encoding in the associative memories, we should use the Hamming net with MAXNET [17] (which can optimally recall the library pairs under noisy input) instead of using an exponential scheme of information flow. Same as the higher order BAM, its stability is unknown. [1]

Remark: Except the approach of the layer-extension, all the modification above-mentioned only involve integer-valued weights. Hence, their hardware implementation can be accomplished in an easy way.

5.3 Householder Encoding Algorithm

5.3.1 Construction from Householder Transforms

Here, we present the details of HCA [32]. To ensure each library pair (X_h, Y_h) be stored as a fixed point, it is equivalent to find a matrix W^*, such that

$$F_B = \operatorname{sgn}(W^* F_A) \tag{5.19}$$

$$F_A = \operatorname{sgn}\left(W^{*T} F_B\right) \tag{5.20}$$

where

$$F_A = \left[X_1 \vdots X_2 \vdots \cdots X_m\right]$$

$$F_B = \left[Y_1 \vdots Y_2 \vdots \cdots Y_m\right].$$

If we use two different connection matrices, W_f^* (from layer F_X to layer F_Y) and W_b^* (from layer F_Y to layer F_X), the requirement that all library pairs are fixed points

[1] Although the stabilization of the EBAM was proved in [50] and [53], the proof has a mistake which is similar to the case of the second order BAM mentioned in Section 4.2.

then becomes

$$
\begin{aligned}
F_B &= \operatorname{sgn}\left(W_f^* F_A\right) \\
F_A &= \operatorname{sgn}\left(W_b^* F_B\right) . \qquad (5.21)
\end{aligned}
$$

A sufficient condition for establishing the above relationship is

$$F_B = W_f^* F_A \qquad (5.22)$$

$$F_A = W_b^* F_B . \qquad (5.23)$$

In the following, we will demonstrate that the Householder transforms [70, 71] (also named as the Householder reflection) is a handy tool to find W_f^* and W_b^* such that (5.22) and (5.23) can be established. As the technique involves the Householder transforms, we refer it as the *householder encoding algorithm.*

If all X_h's, as well as all Y_h's, are linearly independent, we can define two rotation matrices R_A and R_B such that

$$Y_A = R_A \, F_A = \begin{bmatrix} Y_{A,\emptyset} \\ \emptyset_{n-m,m} \end{bmatrix} \qquad (5.24)$$

and

$$Y_B = R_B \, F_B = \begin{bmatrix} Y_{B,\emptyset} \\ \emptyset_{p-m,m} \end{bmatrix} . \qquad (5.25)$$

where $Y_{A,\emptyset}$ and $Y_{B,\emptyset}$ are upper triangular matrices with dimension $m \times m$, and $\emptyset_{n-m,m}$ and $\emptyset_{p-m,m}$ are zero matrices. Since both R_A and R_B are orthonormal,

$$F_A = {R_A}^T Y_A \qquad (5.26)$$

$$F_B = {R_B}^T Y_B . \qquad (5.27)$$

Due to sparsity of Y_A and Y_B, the rotation matrix R_A in (5.26) can be reduced in dimension by eliminating the $n-m$ rows after the $(m+1)$th row. Likewise, we also eliminate the $p-m$ rows after the $(m+1)$th row of the rotation matrix R_B. After the eliminations, (5.26) and (5.27) becomes

$$F_A = \hat{R_A}^T Y_{A,\emptyset} \qquad (5.28)$$

$$F_B = \hat{R_B}^T Y_{B,\emptyset} \qquad (5.29)$$

where $\hat{R_A}$ and $\hat{R_B}$ are the first m rows of R_A and R_B, respectively. By substituting (5.28) and (5.29) into (5.22) and (5.23), we obtain

$$F_B = W_f^* \hat{R_A}^T Y_{A,\emptyset} \tag{5.30}$$

$$F_A = W_b^* \hat{R_B}^T Y_{B,\emptyset} \,. \tag{5.31}$$

Note that $\hat{R_A}\,\hat{R_A}^T$ and $\hat{R_B}\,\hat{R_B}^T$ are identity matrices with dimension $m \times m$, we therefore obtain two connection matrices, W_f^* and W_b^*, as follows:

$$W_f^* = F_B \, Y_{A,\emptyset}{}^{-1} \hat{R_A} \tag{5.32}$$

$$W_b^* = F_A \, Y_{B,\emptyset}{}^{-1} \hat{R_B} \,, \tag{5.33}$$

where W_f^* and W_b^* are referred as the forward connection matrix and backward connection matrix, respectively. Let $F_A^+ = Y_{A,\emptyset}{}^{-1}\hat{R_A}$ and $F_B^+ = Y_{B,\emptyset}{}^{-1}\hat{R_B}$. Since $F_A F_A^+$ and $F_B F_B^+$ are the identity matrices, F_A^+ and F_B^+ can be considered as the generalized inverses of F_A and F_B, respectively.

5.3.2 Construction from iterative method

In the above, we use the Householder transforms to find the generalized inverses and then create the two connection matrices such that all the library pairs are stored as fixed points. In fact, one can use Widrow Hoff algorithm [1] to separately find the connection matrices W_f^* and W_b^* by introducing two projection matrices. If all X_h's, as well as all Y_h's, are linearly independent, we first define two initial connection matrices $W_f^{(1)}$ and $W_b^{(1)}$,

$$W_f^{(1)} = \frac{Y_1 X_1{}^T}{\| X_1 \|^2} \tag{5.34}$$

$$W_b^{(1)} = \frac{X_1 Y_1{}^T}{\| Y_1 \|^2} \,, \tag{5.35}$$

where

$$\| X \| = \sqrt{X^T X} \,.$$

The two connection matrices can be obtained in the following way:

$$W_f^{(h+1)} = W_f^{(h)} + \frac{(Y_{h+1} - W_f^{(h)} X_{h+1}) \hat{X}_{h+1}^T}{\| \hat{X}_{h+1} \|^2} \tag{5.36}$$

$$
\begin{aligned}
W_b^{(h+1)} &= W_b^{(h)} + \frac{(X_{h+1} - W_b^{(h)} Y_{h+1}) \hat{Y}_{h+1}^T}{\| \hat{Y}_{h+1} \|^2}, && (5.37) \\
\hat{X}_{h+1} &= X_{h+1} - P_X^{(h)} X_{h+1} && (5.38) \\
\hat{Y}_{h+1} &= Y_{h+1} - P_Y^{(h)} Y_{h+1} \,,. && (5.39)
\end{aligned}
$$

where $P_X^{(h)}$ and $P_Y^{(h)}$ are the linear projection matrices whose spaces are formed by $X_1, X_2, \cdots, X_h$ and $Y_1, Y_2, \cdots, Y_h$, respectively. We can apply the mathematical induction to prove that (5.36) and (5.37) can find the two connection matrices W_f^* and W_b^*. **In the above equations we separately update the two connection matrices and should keep the two projection matrices during learning.**

In fact, it is not necessary to memorize the two projection matrices if we thoroughly consider the relationship between $W_f^{(h)}$ and $W_b^{(h)}$:

$$
\begin{aligned}
W_b^{(h)} W_f^{(h)} &= P_X^{(h)} && (5.40) \\
W_f^{(h)} W_b^{(h)} &= P_Y^{(h)} . && (5.41)
\end{aligned}
$$

The above equations can be proved by mathematical induction. In our proposed iterative mode HCA, the update equations are

$$
\begin{aligned}
W_f^{(h+1)} &= W_f^{(h)} + \frac{(Y_{h+1} - W_f^{(h)} X_{h+1}) \hat{X}_{h+1}^T}{\| \hat{X}_{h+1} \|^2} && (5.42) \\
W_b^{(h+1)} &= W_b^{(h)} + \frac{(X_{h+1} - W_b^{(h)} Y_{h+1}) \hat{Y}_{h+1}^T}{\| \hat{Y}_{h+1} \|^2}, && (5.43) \\
\hat{X}_{h+1} &= X_{h+1} - W_b^{(h)} W_f^{(h)} X_{h+1} && (5.44) \\
\hat{Y}_{h+1} &= Y_{h+1} - W_f^{(h)} W_b^{(h)} Y_{h+1} \,. && (5.45)
\end{aligned}
$$

Clearly, it is not necessary to memorize the two projection matrices in the iterative mode HCA during learning. Hence, if a new library is encoded, we need to update the connection matrices based on the current connection matrices only. Moreover, $\hat{X}_{h+1}$ and $\hat{Y}_{h+1}$ can be regarded as the feedback error vectors which are obtained from the bidirectional information flow between the two layers.

Remark: Recently, similar iterative equations are independently discovered in [51]. But, in [51] the two projection matrices $P_X^{(h)}$ and $P_Y^{(h)}$ should be memorized during encoding. As shown in the above, it is not necessary to memorize the two projection matrices by well considering the relationship between $W_f^{(h)}$ and $W_b^{(h)}$.

5.3.3 Remarks on HCA

So far, we have presented two different methods to construct the two connection matrices W_f^* and W_b^*. One is a batch mode method which finds the generalized inverses of F_A and F_B first. Another one is an iterative mode method which directly finds W_f^* and W_b^*. Their properties are summarized below.

- From (5.22) and (5.23), we can regard HCA as a variation of the outer product rule. The connection matrices are the outer products of the library pairs and their generalized inverses.
- If each component of the library pairs is a ± 1 equiprobable independent random variable, then the memory capacity of HCA tends to $\min(n,p)$. It is because the probability that m random bipolar vectors with dimension d (d is greater than m) are linearly independent tends to one when d is sufficiently large [63]. From Chapter 2, the memory capacity of Kosko's encoding scheme is $\frac{\min(n,p)}{4\log\min(n,p)}$. Hence, from the statistical point of view, the memory capacity of HCA is greater than that of Kosko's encoding scheme. Also, the difference of the memory capacities between them is dramatic when the dimensions of the library pairs are large.
- From the simulation shown in Section 5.7, the error correction capability of HCA is better than that of the three other proposed encoding methods.
- Since the HCA has two different connection matrices, the hardware resource of HCA is twice to that of Kosko's encoding scheme. Since the real-valued weights are involved in the HCA, the hardware implementation of HCA is more difficult than that of Kosko's encoding scheme.
- Since there are two connection matrices in the HCA, we cannot make any conclusion about the stability of BAM under the HCA. Note that the proof of

stability in Section 2.3 is only suitable for the case of one connection matrix.

- In the iterative mode HCA, when a new library pair is encoded, we update the two connection matrices based on (5.42) and (5.43). Also, we need to keep the two connection matrices only and do not need to memorize the previous library pairs. On the other hand, the batch mode HCA produces the generalized inverses first. They are necessary for the EHCA presented in the next section. The iterative mode directly solves the two connection matrices and we cannot use the result of the iterative mode in the EHCA.

5.4 Enhanced Householder Encoding Algorithm

5.4.1 Construction of EHCA

Under the HCA, there are two different connection matrices and hence the stabilization of BAM is unknown. The EHCA [34] presented here is developed on the basis of HCA and projection on convex sets (POCS) [72, 73]. In the EHCA, the two matrices found by the HCA are reduced into one matrix by POCS. Hence, the stable property of BAM can surely be maintained.

If all X_h's and all Y_h's are linearly independent, the two connections W_f^* and W_b^* can be found by using HCA

$$W_f^* = F_B\ F_A^+ \text{ and } W_b^* = F_A\ F_B^+ \tag{5.46}$$

In fact, W_f^* and W_b^* defined by (5.46) are the particular solutions of (5.19) and (5.20). The general solution, $W_{f\alpha}$ and $W_{b\beta}$, is

$$W_{f\alpha} = F_{B\alpha}\ F_A^+ \text{ and } W_{b\beta} = F_{A\beta}\ F_B^+\ . \tag{5.47}$$

The matrices $F_{B\alpha}$ and $F_{A\beta}$ are defined as

$$F_{B\alpha} = \begin{pmatrix} \alpha_{11}\, y_{11} & \alpha_{12}\, y_{12} & \cdots & \alpha_{1m}\, y_{1m} \\ \alpha_{21}\, y_{21} & \alpha_{22}\, y_{22} & \cdots & \alpha_{2m}\, y_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_{p1}\, y_{p1} & \alpha_{p2}\, y_{p2} & \cdots & \alpha_{pm}\, y_{pm} \end{pmatrix}$$

$$F_{A\beta} = \begin{pmatrix} \beta_{11}\, x_{11} & \beta_{12}\, x_{12} & \cdots & \beta_{1m}\, x_{1m} \\ \beta_{21}\, x_{21} & \beta_{22}\, x_{22} & \cdots & \beta_{2m}\, x_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ \beta_{n1}\, x_{n1} & \beta_{n2}\, x_{n2} & \cdots & \beta_{nm}\, x_{nm} \end{pmatrix}$$

where $\alpha_{jh} > 0 \,\forall\, (j,h)$, and $\beta_{ih} > 0 \,\forall\, (i,h)$. If there exist

$$\alpha_{jh} > 0 \text{ and } \beta_{ih} > 0 \;\; \forall (i,j,h) \tag{5.48}$$

such that

$$W_{f\alpha} = {W_{b\beta}}^T\,, \tag{5.49}$$

then the solution of (5.21) becomes

$$\underline{W}^* = W_{f\alpha} = {W_{b\beta}}^T \tag{5.50}$$

Inequality (5.48) defines a convex set C_0 of α_{jh} and β_{ih}. Also, Equation (5.49) defines np convex sets, C_1 to C_{np}, of α_{jh} and β_{ih}. It is because there are np linear homogeneous equations of α_{jh} and β_{ih} in (5.49). Assume that these convex sets have common intersection

$$C_0 \cap C_1 \cdots \cap C_{n \cdot p} = C \neq \emptyset\,. \tag{5.51}$$

As mentioned in [72, 73], the fundamental result of POCS is that repeated sequential projection onto these sets asymptotically approaches a point in C. Hence, the technique of POCS can be applied to find α_{jh} and β_{ih} such that (5.48) and (5.49) can be established. Finally, the solution connection matrix W^* can be constructed.

The EHCA can be summarized as follows:

- Using the HCA, the particular connection matrices, W_f^* and W_b^*, can be found.

- Applying the POCS, the solution of (5.20), W^*, can be obtained. It means that the two connection matrices W_f and W_b can be reduced into a matrix W^* by POCS.

5.4.2 Remarks on EHCA

- From (5.47), the EHCA can also be considered as a variation of the outer production. The connection matrix is the outer product of the weighted library pairs and their generalized inverses.

- In the EHCA, if C is an empty set, then the two matrices W_f^* and W_b^* cannot be reduced into one. Therefore, the memory capacity of HCA is greater than that of EHCA. Fortunately, the simulation results in Section 5.7 show that the probability of C being not empty is very high.

- From the simulation in Section 5.7, the error correction capability of EHCA is only a little poorer than that of HCA. Also, the error correction capability of EHCA is better than that of BL or AHKBL. Hence, the EHCA is also an efficient algorithm for BAM in terms of the error correction capability and the memory capacity.

- As there is only one connection matrix in the EHCA, the stable property of BAM can surely be maintained.

- The EHCA is a fully batch mode learning rule. Also, there are np convex sets, we need to do np projections within an iteration. From the simulations, we will get the asymptotical point after hundreds of iterations (for $n = p = 32$, $m = 16$). Hence, the computation complexity of EHCA is higher than that of HCA, BL, and AHKBL.

- Since the real-valued weights are involved in the EHCA, the hardware implementation of EHCA is more difficult than that of Kosko's encoding method.

5.5 Bidirectional Learning

5.5.1 Construction of BL

The BL [35]–[37] is developed on the basis of the perceptron learning algorithm [4]. Recall that the task of an encoding method is to find one of the solution connection matrices (Suppose they exist) W^* such that (5.19) and (5.20) are established, i.e. each library pair is an isolated local minimum of the energy function. The equations (5.19) and (5.20) can be viewed as a decision function of a single-layer perceptron in a bidirectional sense. In a forward manner, X_h and Y_h are input and output patterns, respectively. In a backward manner , Y_h and X_h are input and output patterns, respectively. In the BL, the library pairs are repeatedly and sequentially presented to the BAM to update the connection matrix according to the perceptron rule until all are correctly classified by the connection matrix. The changes of connection matrix are computed by the delta-rule, in both the forward and backward directions. They are updated by a sum of changes from both directions at each iteration. Thus, it is named *bidirectional learning.*

Suppose that the library pair at the u-th iteration is (X_u, Y_u). In the forward manner, the correction in the j-th row and the i-th column of the connection matrix is

$$\Delta_1 w_{ji}^{(u)} = \left(y_{ju} - \operatorname{sgn}\left(\sum_{i'=1}^{n} w_{ji'}^{(u)} x_{i'u}\right)\right) x_{iu}. \tag{5.52}$$

Similarly, in the backward manner the correction in the j-th row and the i-th column of the connection matrix is

$$\Delta_2 w_{ji}^{(u)} = \left(x_{iu} - \operatorname{sgn}\left(\sum_{j'=1}^{p} w_{j'i}^{(u)} y_{j'u}\right)\right) y_{ju}. \tag{5.53}$$

A total correction at the u-th iteration is

$$\Delta w_{ji}^{(u)} = \Delta_1 w_{ji}^{u} + \Delta_2 w_{ji}^{u}, \tag{5.54}$$

and the updating rule is

$$w_{ji}^{(u+1)} = w_{ji}^{(u)} + \Delta w_{ji}^{(u)}. \tag{5.55}$$

The updating rule is expressed in a matrix form as follows

$$W^{(u+1)} = W^{(u)} + \Delta W^{(u)}, \tag{5.56}$$

where

$$\begin{aligned}
\Delta W^{(u)} &= \Delta_1 W^{(u)} + \Delta_2 W^{(u)}, \\
\Delta_1 W^{(u)} &= \left(Y_u - \operatorname{sgn}\left(W^{(u)} X_u\right)\right) X_u^T, \\
\Delta_2 W^{(u)} &= Y_u \left(X_u - \operatorname{sgn}\left(W^{(u)T} Y_u\right)\right)^T.
\end{aligned}$$

Note that we can use any random matrix as the initial guess $W^{(1)}$ but we suggest using $F_B\, {F_A}^T$ (Kosko's method). From the simulation, we observe that the learning speed of using this initial guess is higher than that of using random matrix when m is small. In the case of large m, the learning speeds of the two initial conditions are similar. When m is small, $F_B\, F_A^T$ is close to a solution. Hence, the BL with $F_B {F_A}^T$ as the initial guess can find the solution with a few learning cycles.

5.5.2 The Convergence of BL and the memory capacity of BL

We first prove two lemmas. Based on the two lemmas, we prove that the BL yields a solution connection matrix within a finite number of iterations if there exists a set of solution connection matrices. Then, we can conclude that the memory capacity of BL is greater than or equal to that of any other learning rule. Note that a stacking vector form rather than a matrix form is used throughout the proof.

$$\begin{aligned}
W_S^* &= \begin{pmatrix} w_{11}^* & w_{21}^* & \dots & w_{p1}^* & \dots & w_{ji}^* & \dots & w_{pn}^* \end{pmatrix}^T, \\
W_S^{(u)} &= \begin{pmatrix} w_{11}^{(u)} & w_{21}^{(u)} & \dots & w_{p1}^{(u)} & \dots & w_{ji}^{(u)} & \dots & w_{pn}^{(u)} \end{pmatrix}^T, \\
\Delta_1 W_S^{(u)} &= \begin{pmatrix} \Delta_1 w_{11}^{(u)} & \Delta_1 w_{21}^{(u)} & \dots & \Delta_1 w_{p1}^{(u)} & \dots & \Delta_1 w_{ji}^{(u)} & \dots & \Delta_1 w_{pn}^{(u)} \end{pmatrix}^T, \\
\Delta_2 W_S^{(u)} &= \begin{pmatrix} \Delta_2 w_{11}^{(u)} & \Delta_2 w_{21}^{(u)} & \dots & \Delta_2 w_{p1}^{(u)} & \dots & \Delta_2 w_{ji}^{u} & \dots & \Delta_2 w_{pn}^{(u)} \end{pmatrix}^T, \\
\Delta W_S^{(u)} &= \Delta_1 W_S^{(u)} + \Delta_2 W_S^{(u)}
\end{aligned}$$

Then the learning rule becomes

$$W_S^{(u+1)} = W_S^{(u)} + \Delta W_S^{(u)} . \tag{5.57}$$

Lemma 5.3 *If there exists a solution vector W_S^* (the stacking vector form of a solution connection matrix W^*) and $\Delta W_S^{(u)}$ is not a zero vector $\emptyset$ (meaning that corrections in the connection matrix take place at the u-th iteration), then $\Delta W_S^{(u)T} W_S^* > 0$,*

Proof of Lemma 5.3

$$\Delta W_S^{(u)T} W_S^* = \sum_{j=1}^{p} A'_j + \sum_{i=1}^{n} B'_i,$$

where

$$\begin{aligned} A'_j &= \left(y_{ju} - \operatorname{sgn}\left(\sum_{i'=1}^{n} x_{i'u} w_{ji'}^{(u)}\right)\right)\left(\sum_{i'=1}^{n} x_{i'u} w_{ji'}^{*}\right), \\ B'_i &= \left(x_{iu} - \operatorname{sgn}\left(\sum_{j'=1}^{p} y_{j'u} w_{j'i}^{(u)}\right)\right)\left(\sum_{j'=1}^{p} y_{j'u} w_{j'i}^{*}\right). \end{aligned}$$

Since W_S^* is a solution, the sign of $\sum_{i'=1}^{n} x_{iu} w_{ji'}^*$ equals that of y_{ju}. As a result, $A'_j \geq 0$ $\forall j$. Also, the sign of $\sum_{j'=1}^{p} y_{j'u} w_{j'i}^*$ equals that of x_{iu}, then $B'_i \geq 0$ $\forall i$. But $\Delta W_S^{(u)} \neq \emptyset$ and it is not possible that all A'_j and B'_i are zero for all i, j. Hence, $\Delta W_S^{(u)T} W_S^* > 0$ is true. The proof is completed. Note that the only assumption of Lemma 5.3 is that there exists a set of solution connection matrices. □

Lemma 5.4 *A non-solution vector $W_S^{(u)}$ yields $\Delta W_S^{(u)T} W_S^{(u)} \leq 0$.*

Proof of Lemma 5.4

$$\Delta W_S^{(u)T} W_S^{(u)} = \sum_{j=1}^{p} A_j + \sum_{i=1}^{p} B_i,$$

where

$$\begin{aligned} A_j &= \left(y_{ju} - \operatorname{sgn}\left(\sum_{i'=1}^{n} x_{i'u} w_{ji'}^{(u)}\right)\right)\left(\sum_{i'=1}^{n} x_{i'u} w_{ji'}^{(u)}\right) \\ B_i &= \left(x_{iu} - \operatorname{sgn}\left(\sum_{j'=1}^{p} y_{j'u} w_{j'i}^{(u)}\right)\right)\left(\sum_{j'=1}^{p} y_{iu} w_{j'i}^{(u)}\right) \end{aligned}$$

It can easily be shown that $A_j \leq 0$ for all j and $B_i \leq 0$ for all i. Hence, $\Delta W_S^{(u)^T} W_S^{(u)} \leq 0$ is true. Note that Lemma 5.4 is valid for all cases (either a solution exists or not). □

With Lemma 5.3 and 5.4, we can obtain the following theorem which can be proved by following the proof of the perceptron convergence in [4].

Theorem 5.1 *Given any set of library pairs, if there exists a set of solution connection matrices, the BL yields one of these within a finite number of iterations.*

Proof of Theorem 5.1

The theorem means that if the solution connection matrices exist, then after some finite index value u',

$$W_S^{(u')} = W_S^{(u'+1)} = W_S^{(u'+2)} = \ldots.$$

Note that if u' is finite, $W_S^{(u')}$ (stacking vector form of $W^{(u')}$) is also a solution. It is because the correction in the connection matrix $W^{(u')}$ is zero for each library pair (X_h, Y_h). That is, each library pair is stored as a fixed point. Therefore, if u' is finite, $W_S^{(u')}$ is also a solution.

Now, the proof of the theorem is equivalent to proving that u' is finite if there exists a set of solutions. Without loss of generality, the proof is facilitated by considering only the indices u's for which corrections in the connection matrix take place during training.

From (5.57),

$$W_S^{(u+1)} = W_S^{(1)} + \Delta W_S^{(1)} + \Delta W_S^{(2)} + \ldots + \Delta W_S^{(u)}. \tag{5.58}$$

Taking the inner product of one of the solutions W_S^* on both sides of equation (5.58) yields,

$$W_S^{(u+1)^T} W_S^* = W_S^{(1)^T} W_S^* + \Delta W_S^{(1)^T} W_S^* + \Delta W_S^{(2)^T} W_S^* + \ldots + \Delta W_S^{(u)^T} W_S^*. \tag{5.59}$$

From Lemma 5.3, each term $\Delta W_S^{(i)^T} W_S^*, i = 1, \ldots, u$, is greater than zero, then

$$W_S^{(u+1)^T} W_S^* \geq W_S^{(1)^T} W_S^* + ug, \tag{5.60}$$

where

$$g = \min_i \left(\Delta W_S^{(i)^T} W_S^* \right).$$

Using the Cauchy-Schwartz inequality,

$$\| W_S^{(u+1)} \|^2 \geq \frac{(W_S^{(u+1)^T} W_S^*)^2}{\| W_S^* \|^2}. \tag{5.61}$$

Substituting (5.60) into (5.61) yields

$$\| W_S^{(u+1)} \|^2 \geq \frac{(W_S^{(1)^T} W_S^* + ug)^2}{\| W_S^* \|^2}. \tag{5.62}$$

From equation (5.57),

$$\| W_S^{(u+1)} \|^2 = \| W_S^{(u)} \|^2 + 2 W_S^{(u)^T} \Delta W_S^{(u)} + \| \Delta W_S^{(u)} \|^2 . \tag{5.63}$$

On the basis of Lemma 5.4 and letting $g' = \max_i \| \Delta W_S^{(i)} \|^2$ results in

$$\| W_S^{(i+1)} \|^2 - \| W_S^{(i)} \|^2 \leq g'. \tag{5.64}$$

Adding these inequalities for $i = 1, \ldots, u$ yields

$$\| W_S^{(u+1)} \|^2 \leq \| W_S^{(1)} \|^2 + ug'. \tag{5.65}$$

Expressions (5.62) and (5.65) establish conflicting bounds on $\| W_S^{(u+1)} \|^2$ for sufficiently large u. In fact, u cannot be larger than u', where u' is the solution of the equation

$$\frac{(W_S^{(1)^T} W_S^* + u'g)^2}{\| W_S^* \|^2} = \| W_S^{(1)} \|^2 + u'g' . \tag{5.66}$$

According to equation (5.66), u' is finite, implying that the BL yields a solution connection matrix if there exists a set of solution connection matrices. The proof is completed. □

Theorem 5.1 means that given any set of the library pairs, if the solution connection matrices exist, the BL will find one. From the theorem, if the BL is not able to find a solution (which indeed does not exist), then other learning rules also cannot find a solution. Hence, the following corollary can be obtained.

Corollary 5.1 *The memory capacity of BL is larger than or equal to that of other learning rules.*

Note that other learning rules, such as the Kosko's method or EHCA, may not be able to find a solution connection matrix even though the solutions exist. Since Corollary 5.1 implies that the BL pushes the memory capacity to the maximal, the BL can be considered as an optimum learning rule for the BAM in terms of memory capacity. Also, Theorem 5.1 is independent of the statistical distribution of the library pairs. That means, Theorem 5.1 and Corollary 5.1 hold for any statistical distribution. Hence, *the BL is an optimum learning rule for any statistical distribution in terms of memory capacity*. However, it should be emphasized that the BL cannot be used to determine whether the solution exists or not. It is because we cannot use the infinite number of learning cycles to train the BAM.

5.5.3 Remarks on BL

- The BL is an iterative encoding algorithm. The library pairs are repeatedly and sequentially presented to the BAM to update the connection matrix. It yields one of the solution connection matrices within a finite number of iterations (if solutions exist).

- Here, we only present the synchronous BL:

$$W^{new} = W^{old} + \Delta W_f + \Delta W_b \,,$$

where ΔW_f and ΔW_b (see (5.56)), respectively, are the forward error and backward error on the basis of the old connection matrix W^{old}. We can also use the sequential BL :

$$W^{temp} = W^{old} + \Delta W_{f'} \tag{5.67}$$

$$W^{new} = W^{temp} + \Delta W_{b'} \tag{5.68}$$

where $\Delta W_{f'}$ is the forward error based on the old connection matrix W^{old}, and $\Delta W_{b'}$ is the backward error based on the temporary connection matrix W^{temp}. We can easily verify that the sequential BL also yields one of the solution connection matrices within a finite number of iterations (if solutions exist).

- Under the condition that there is only one connection matrix in the BAM, the memory capacity of BL is greater than or equal to that of the other learning rules.

- As there is only one connection matrix in the BL, the stable property of BAM is guaranteed.

- Since the BL only involves integer-valued weights, the hardware implementation of BL is easier than that of HCA and EHCA. Also, the connection matrix is only changed by $-4, -3, -2, -1, 1, 2, 3$ or 4 during each presentation. It makes its hardware implementation easier.

- Since the perceptron rule cannot locate a good decision surface [74], the BL has also weak error correction nature. From the simulations in Section 5.7, the error correction capability of BL is poorer than that of HCA, EHCA, and AHKBL.

5.6 Adaptive Ho-Kashyap Bidirectional Learning

5.6.1 Construction of AHKBL

As mentioned previously, the BL is weak in error correction. Since the concept of adaptive Ho-Kashyap rule (AHK) [74] can generate a robust decision surface in single layer perceptron, we extend the BL to the AHKBL [38] to improve the error correction capability.

Prior to going through the detail of AHKBL, the AHK rule is first reviewed. In a single-layer perceptron, the output $\vec{Y}$ is defined as:

$$\vec{Y} = \mathrm{sgn}(\underline{W}\vec{X}),$$

where $\underline{W}$ is the connection matrix of the perceptron and $\vec{X}$ is the input. Consider there is a set of training input vectors $\vec{X}_k$ with dimension L_X, where $k = 1, \cdots, N$. Each training vector associates with a bipolar output vector $\vec{Y}_k \in \{-1, 1\}^{L_Y}$. Training the perceptron is equivalent to finding out a $\underline{W}$ such that :

$$\vec{Y}_k \otimes (\underline{W}\ \vec{X}_k) = \vec{D}_k > \vec{0} \text{ for } k = 1, \cdots, N \tag{5.69}$$

where $\otimes$ is an elementwise multiplication operator ($\vec{C} = \vec{A} \otimes \vec{B}$ means that $c_i = a_i b_i\ \forall\, i$, and $\vec{A} > \vec{B}$ means that $a_i > b_i\ \forall\, i$). In the AHK, a positive valued margin vector $\vec{D}_k$ and an error vector $\vec{\varepsilon}_k$ are introduced with each input-output pair ($\vec{X}_k$, $\vec{Y}_k$). The training procedure is that the input-output pair $(\vec{X}_k, \vec{Y}_k)$ is repeatedly and sequentially presented to the perceptron. The connection matrix is updated according to

$$\underline{W}^{new} = \underline{W}^{old} + \frac{\rho_1 \rho_2}{2} \left[\vec{Y}_k \otimes \mid \vec{\varepsilon}_k^{old} \mid + (1 - \frac{2}{\rho_1}) \vec{Y}_k \otimes \vec{\varepsilon}_k^{old} \right] \vec{X}_k^T \tag{5.70}$$

where

$$\vec{\varepsilon}_k^{\,old} = \vec{Y}_k \otimes (\underline{W}^{old} \vec{X}_k) - \vec{D}_k^{old} . \tag{5.71}$$

Also, the margin vector $\vec{D}_k$ is updated as:

$$\vec{D}_k^{new} = \vec{D}_k^{old} + \frac{\rho_1}{2} (\mid \vec{\varepsilon}_k^{old} \mid + \vec{\varepsilon}_k^{old}) . \tag{5.72}$$

The superscripts "old" and "new" in the above represent current and updated values, respectively. The notation $\mid \cdot \mid$ denotes the absolute value of the components of the argument vector (i.e. $\mid \vec{A} \mid = (\mid a_1 \mid, \mid a_2 \mid, \ldots, \mid a_{L_A} \mid)^T$). Note that within a learning cycle, the connection matrix is updated N times but each margin vector $\vec{D}_k$ is updated once only.

After a presentation, a new estimate of weights is obtained and each element of the margin vector may be increased. If an element of the error vector is negative, the corresponding element of the margin vector is unchanged and only the weights are updated to overcome the current margin. Otherwise, the value of the corresponding element in the margin vector is increased and the weights are updated. This iterative process creates a perceptron with a large margin. However, large margin does not always imply a robust boundary. For example, if the magnitude of the weights is very large, then we also get large margins and this cannot imply a robust boundary. On the other hand, if the magnitude of errors decreases [2] and the value of margins increases during training, then the boundary becomes robust. The sufficient conditions for convergence of the AHK rule [74] have been found as

$$0 < \rho_1 < 2 \text{ and } 0 < \rho_2 < \frac{2}{L^2_{\max}} . \tag{5.73}$$

where $L_{\max}$ is the length of the longest input vector.

Combining the concepts of BL and AHK, we propose the AHKBL here. For each library pair (X_h, Y_h), we introduce two positive margin vectors $D_{f,h}$ (in forward sense) with dimension p and $D_{b,h}$ (in backward sense) with dimension n. Also, we introduce two error vectors $\varepsilon_{f,h}$ (in forward sense) with dimension p and $\varepsilon_{b,k}$ (in backward sense) with dimension n for each library pair (X_h, Y_h). During training, each library pair is repeatedly and sequentially presented to the BAM. Suppose that in the current presentation (X_h, Y_h) is presented, the correction of the connection matrix takes place as follows:

1. In forward manner, X_h and Y_h are input and output respectively. The forward correction of connection matrix is

$$\Delta W_f = \frac{\rho_1 \rho_2}{2} \left[Y_h \otimes \mid \varepsilon^{old}_{f,h} \mid + (1 - \frac{2}{\rho_1}) Y_h \otimes \varepsilon^{old}_{f,h} \right] X^T_h \ , \tag{5.74}$$

   where $\varepsilon^{old}_{f,h} = Y_h \otimes (W^{old} X_h) - D^{old}_{f,h}$. Then the forward margin vector is updated as

$$D^{new}_{f,h} = D^{old}_{f,h} + \frac{\rho_1}{2} (\mid \varepsilon^{old}_{f,h} \mid + \varepsilon^{old}_{f,h}) . \tag{5.75}$$

2. According to this forward correction, a new connection matrix W^{fnew} is obtained

$$W^{fnew} = W^{old} + \Delta W_f . \tag{5.76}$$

3. Now, in backward manner, Y_h and X_h are input and output respectively. According to the new connection matrix W^{fnew} obtained in step 2, the backward

[2]From (5.70), decreasing of the magnitude of error vectors means that the magnitude of the weights is limited.

correction of the connection matrix is

$$\Delta W_b = \frac{\rho_1\rho_2}{2} Y_h \left[X_h \otimes \mid \varepsilon_{b,h}^{old} \mid + (1 - \frac{2}{\rho_1}) X_h \otimes \varepsilon_{b,h}^{old} \right]^T , \tag{5.77}$$

where $\varepsilon_{b,h}^{old} = X_h \otimes (W^{fnew^T} Y_h) - D_{b,h}^{old}$. Then the new backward margin vector is

$$D_{b,h}^{new} = D_{b,h}^{old} + \frac{\rho_1}{2}(\mid \varepsilon_{b,h}^{old} \mid + \varepsilon_{b,h}^{old}) . \tag{5.78}$$

4. According to the backward correction, the connection matrix is updated again

$$W^{new} = W^{fnew} + \Delta W_b . \tag{5.79}$$

Note that there is only one connection matrix in the AHKBL and the connection matrix is updated twice in each presentation. Hence, the BAM with AHKBL is stable during recall.

In general, there is no terminating condition on the AHKBL and more training implies a more robust BAM obtained. However, we can impose some terminating conditions on the AHKBL. For example, the AHKBL terminates until all the library pairs are stored as fixed points. Or, we can pre-define the number of learning cycles for the AHKBL.

5.6.2 Convergent Conditions for AHKBL

In the AHKBL, there are twice updates in each presentation. Each updating follows the AHK rule either in the forward sense or in the backward sense. We can extend the analysis in [74] to obtain the sufficient conditions for convergence of AHKBL. In the $t+1$-th learning cycle the h-th library pair is presented. After the forward updating (Step 2 of AHKBL has taken place), the new error vector $\varepsilon_{f,h}^{new}(t+1)$ is

$$\varepsilon_{f,h}^{new}(t+1) = Y_h \otimes (W^{fnew} X_h) - D_{f,h}^{new}(t+1) . \tag{5.80}$$

Note that $\varepsilon_{f,h}^{new}(t+1)$ is expressed in terms of W^{fnew} instead of W^{new}. It is because the current updated matrix is W^{fnew} after step 2. Substituting (5.74), (5.75), and (5.76) into (5.80),

$$\varepsilon_{f,h}^{new}(t+1) = \left[(1 - \frac{\rho_1}{2}) + \frac{\rho_1\rho_2 n}{2}(1 - \frac{2}{\rho_1}) \right] \varepsilon_{f,h}^{old}(t+1) + (\frac{\rho_1\rho_2 n}{2} - \frac{\rho_1}{2}) \mid \varepsilon_{f,h}^{old}(t+1) \mid . \tag{5.81}$$

Taking the absolute value yields

$$\left|\varepsilon_{f,h}^{new}(t+1)\right| = \left|\left[(1-\frac{\rho_1}{2}) + \frac{\rho_1\rho_2 n}{2}(1-\frac{2}{\rho_1})\right]\varepsilon_{f,h}^{old}(t+1) + (\frac{\rho_1\rho_2 n}{2} - \frac{\rho_1}{2}) \mid \varepsilon_{f,h}^{old}(t+1) \mid \right| . \tag{5.82}$$

Assuming that the minimal disturbance [74] is in effect over one learning cycle $\varepsilon_{f,h}^{old}(t+1) \simeq \varepsilon_{f,h}^{new}(t)$, (5.82) may be approximated as

$$\left|\varepsilon_{f,h}^{new}(t+1)\right| = \left|\left[(1-\frac{\rho_1}{2}) + \frac{\rho_1\rho_2 n}{2}(1-\frac{2}{\rho_1})\right]\varepsilon_{f,h}^{new}(t) + (\frac{\rho_1\rho_2 n}{2} - \frac{\rho_1}{2}) \mid \varepsilon_{f,h}^{new}(t) \mid \right| . \tag{5.83}$$

Hence, in the forward sense the sufficient condition for convergence (i.e. $\mid \varepsilon_{f,h}^{new}(t+1) \mid \leq \mid \varepsilon_{f,h}^{new}(t) \mid$) is

$$0 < \rho_2 < \frac{2}{n} \quad \text{and} \quad 0 < \rho_1 < 2 . \tag{5.84}$$

Similarly, in the backward sense the sufficient condition for convergence (i.e. $\mid \varepsilon_{b,h}^{new}(t+1) \mid \leq \mid \varepsilon_{b,h}^{new}(t) \mid$) is

$$0 < \rho_2 < \frac{2}{p} \quad \text{and} \quad 0 < \rho_1 < 2 . \tag{5.85}$$

Then the overall sufficient convergent conditions are

$$0 \ < \ \rho_1 < 2 \tag{5.86}$$

$$0 \ < \ \rho_2 < \frac{2}{\max(n,p)} . \tag{5.87}$$

5.6.3 Remarks on AHKBL

- The AHKBL is an iterative encoding algorithm. The library pairs are repeatedly and sequentially presented to the BAM. The connection matrix is sequentially updated twice at each presentation. [3] The sufficiently convergent conditions are

$$0 < \rho_1 < 2$$

$$0 < \rho_2 < \frac{2}{\max(n,p)} .$$

[3]We also try to develop a parallel AHKBL in which only one time of updating takes place at each presentation. However, we cannot derive the convergent condition for it. As the computation complexity of the parallel model AHKBL is the same as that of our proposed AHKBL, we prefer to use the proposed AHKBL instead of the parallel model AHKBL.

- As there is only one connection matrix in the AHKBL, the stabilization of the BAM is guaranteed.
- Since the AHKBL involves real-valued weights, the hardware implementation of AHKBL is more difficult than that of BL.
- From the simulations shown in Section 5.7, the memory capacity of AHKBL is the same as that of BL.
- From the simulations shown in Section 5.7, the error correction capability of AHKBL is better than that of BL. Usually, using small learning parameters in the AHKBL will give the better error correction capability.
- There is a trade-off between the learning speed and the error correction capability in the AHKBL (suppose that we terminate the learning when all library pairs are stored as fixed points). Small learning parameters will lead to better performance but a longer training time is required. Conversely, with large learning parameters, we get a relatively poor performance but the training time is shorter. We have done other several simulations for different values of learning parameters and dimensions. Based on the simulations, our recommendation of the learning parameters is ρ_1=0.5 and $\rho_2 = \frac{1}{\max(n,p)}$ which compromises the trade-off.

5.7 Computer Simulations

5.7.1 Memory Capacity

In this section, we will empirically study the memory capacity of different modifications of BAM. The modifications being studied are the DAE, second order BAM, multiple training, PRLAB, HCA, EHCA, BL, and AHKBL. The dimensions are $n = p = 10$, $n = p = 20$, $n = p = 32$, and $n = p = 64$. We vary the number of library pairs m. For the cases of $n = p = 10$ and $n = p = 20$, each value of m has 1000 sets of library pairs and each set contains m library pairs. For the case of

$n = p = 32$ and $n = p = 64$, each value of m has 100 sets of library pairs and each set contains m library pairs. For each element in the library pairs, the probabilities of being 1 and −1 are equal. A set of library pairs is called an event.

In the BL, AHKBL, PRLAB, and multiple training, we terminate the training when all library pairs are stored as fixed points, or the learning cycles reach 1000. The initial connection matrix for the BL, AHKBL, PRLAB, and multiple training is the matrix obtained from Kosko's method. We have tested the AHKBL with four sets of learning parameters, $\rho_1 = 1$ or 0.5 and $\rho_2 = \frac{1}{n}$ or $\frac{2}{n}$. The elements of the margin vectors in the AHKBL are initially set to one. In the DAE, the numbers of additional neurons in each field are 16 and 32 for $n = p = 10$. For the cases of $n = p = 20$, $n = p = 32$, and $n = p = 64$, the numbers of additional neurons in each field are 32 and 64. After training, if every library pair in an event is stored as a fixed point, the event is called a successful event. Table 5.1, 5.2, 5.3 and 5.4 summarize the results of this simulation. Empirically, the tables reflect the memory capacity of different modifications. Since, in the AHKBL the number of successful events is the same for different sets of learning parameters, we use one column to summarize the results.

From the tables, all the modifications investigated here (except the multiple training) greatly improve the memory capacity. In the cases of BL, AHKBL, HCA, EHCA, and PRLAB, the memory capacities are very similar. For the DAE, the memory capacity increases with the number of additional neurons. In terms of information ratio, the HCA, EHCA, BL, AHKBL, and PRLAB are superior to the DAE and second order BAM (if we set 90 % as the threshold of the memory capacity). But there is no significant difference in the information ratio among the EHCA, BL, AHKBL, and PRLAB.

Table 5.1 Comparison of the memory capacity among different modifications of BAM where $n = p = 10$.

No. of successful events ($n = p = 10$)										
m No. of library pairs	Kosko's method	multiple training	HCA	EHCA	BL	AHKBL	PRLAB	DAE with 16 additional neurons in each layer	DAE with 32 additional neurons in each layer	second-order BAM
1	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
2	997	997	997	997	997	997	997	1000	1000	1000
3	905	911	981	981	981	981	981	1000	1000	1000
4	502	539	978	976	977	977	977	1000	1000	984
5	119	122	961	945	950	950	950	1000	1000	938
6	17	17	921	883	890	890	890	991	1000	822
7	1	1	878	787	810	810	810	963	1000	694
8	0	0	799	632	693	693	693	848	1000	526
9	0	0	655	455	550	550	550	676	999	350
10	0	0	364	209	382	382	382	451	997	176

Table 5.2 Comparison of the memory capacity among different modifications of BAM where $n = p = 20$.

No. of successful events ($n = p = 20$)							
m No. of library pairs	Kosko's method	multiple training	Inverse	EHCA	BL	AHKBL	PRLAB
2	1000	1000	1000	1000	1000	1000	1000
4	866	901	1000	1000	1000	1000	1000
6	153	186	1000	1000	1000	1000	1000
8	2	2	999	999	999	999	999
10	0	0	1000	1000	1000	1000	1000
12	0	0	999	991	999	999	999
14	0	0	998	976	998	998	998
16	0	0	998	961	997	997	997
18	0	0	998	933	995	995	995
20	0	0	983	911	993	993	993

Table 5.3 Comparison of the memory capacity among different modifications of BAM where $n = p = 32$.

m No. of library pairs	No. of successful events ($n = p = 32$) Kosko's method	multiple training	HCA	EHCA	BL	AHKBL	PRLAB	DAE with 32 additional neurons in each layer	DAE with 64 additional neurons in each layer	second-order BAM
2	100	100	100	100	100	100	100	100	100	100
4	99	100	100	100	100	100	100	100	100	100
6	41	45	100	100	100	100	100	100	100	100
8	4	5	100	100	100	100	100	100	100	100
10	0	0	100	100	100	100	100	100	100	100
12	0	0	100	100	100	100	100	90	100	100
14	0	0	100	100	100	100	100	66	100	100
16	0	0	100	100	100	100	100	40	99	99
18	0	0	100	100	100	100	100	5	98	97
20	0	0	100	100	100	100	100	1	93	96
	0	0	100	100	100	100	100	0	80	94

Table 5.4 Comparison of the memory capacity among different modifications of BAM where $n = p = 64$.

m No. of library pairs	No. of successful events ($n = p = 64$) Kosko's method	multiple training	HCA	EHCA	BL	AHKBL	PRLAB	DAE with 32 additional neurons in each layer	DAE with 64 additional neurons in each layer	second-order BAM
2	100	100	100	100	100	100	100	100	100	100
4	100	100	100	100	100	100	100	100	100	100
6	98	100	100	100	100	100	100	100	100	100
8	59	70	100	100	100	100	100	98	100	100
10	9	18	100	100	100	100	100	97	100	100
12	0	0	100	100	100	100	100	91	100	100
14	0	0	100	100	100	100	100	42	99	100
16	0	0	100	100	100	100	100	17	94	100
18	0	0	100	100	100	100	100	1	83	100
20	0	0	100	100	100	100	100	0	56	100
22	0	0	100	100	100	100	100	0	33	100
24	0	0	100	100	100	100	100	0	17	100
26	0	0	100	100	100	100	100	0	1	100
28	0	0	100	100	100	100	100	0	1	100
30	0	0	100	100	100	100	100	0	0	100
32	0	0	100	100	100	100	100	0	0	100

5.7.2 Error Correction Capability

Memory capacity is not meaningful without considering the error correction capability. The error correction capabilities of the modifications of BAM are investigated here. The simulation conditions are similar to the above. The dimensions being considered are $n = p = 32$ and $n = p = 64$. After learning, the modifications are tested with the noisy versions of vectors X_h. The fraction of correct recall is recorded in Figure 5.1 to Figure 5.26. Correct recall means that noisy version of X_h recalls (X_h, Y_h). In the AHKBL, we use "until all library pairs as fixed points" as the terminating condition.[4] Note that we can also observe the memory capacity when the noisy level is zero.

From the figures, all the modifications (except the multiple training) greatly improve the error correction capability and the second order BAM has the better error correction capability. The error correction capability of DAE is improved when more additional neurons are used.

Among the approaches of 'change of encoding method' investigated here, if we take 90 % as the threshold, the HCA is the best[5] and the EHCA is the second best[6] in terms of error correction capability. With a suitable choice of learning parameters ($\rho_1 = 0.5$ and $\rho_2 = \frac{1}{\min(n,p)}$) in the AHKBL, the result of AHKBL is comparable to that of EHCA (see Figure 5.5 and Figure 5.10). The result of PRLAB is similar to that of BL. Compared with the PRLAB and BL, the AHKBL (for the four sets of learning parameter) has a better error correction capability. In the AHKBL, using a smaller ρ_1 can obtain a better result but the value of ρ_2 does not much affect the error correction capability.

---

[4]In the AHKBL, we have also used "pre-defined number of learning cycle as 400 " as the terminal conditions but there is only a little improvement.

[5]For example, in the case of $n = p = 32$, BAM under HCA can store 16 pairs when the number of errors is two (see Figure 5.4).

[6]In the case of $n = p = 32$, BAM under EHCA can store 12 pairs when the number of errors is two (see Figure 5.5).

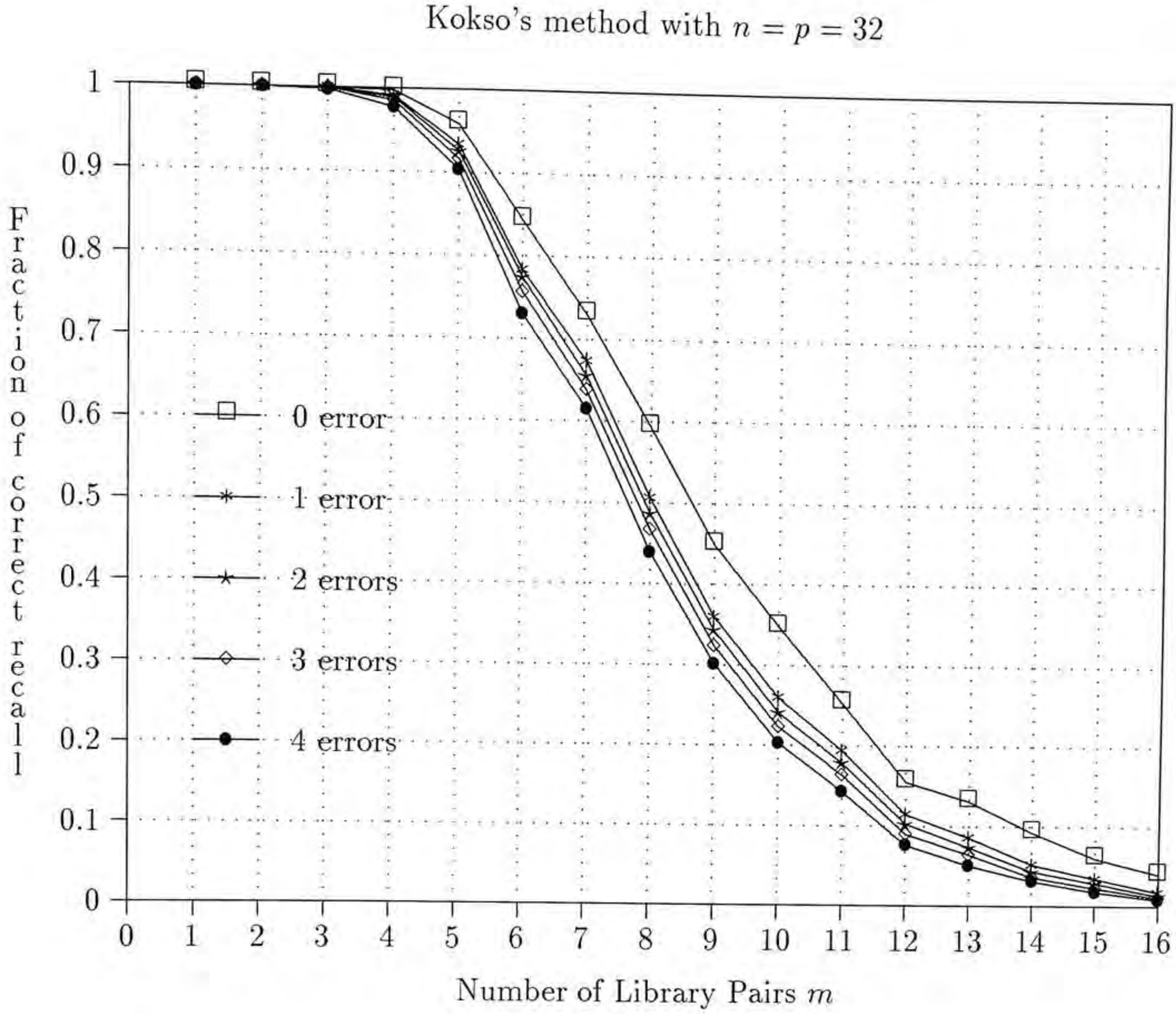


Figure 5.1 The error correction capability of Kosko's encoding method. The dimension is $n = p = 32$.

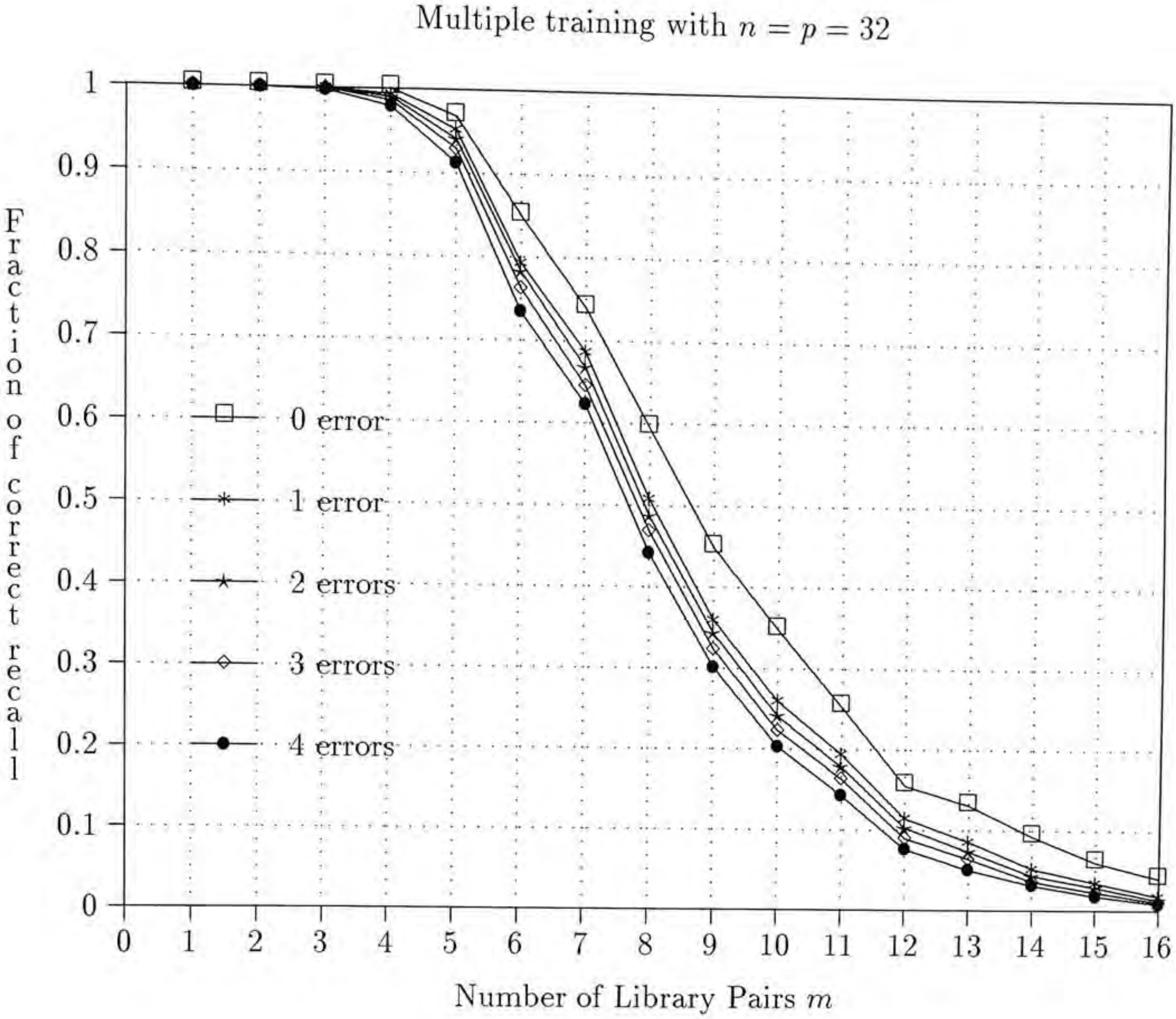


Figure 5.2 The error correction capability of multiple training. The dimension is $n = p = 32$.

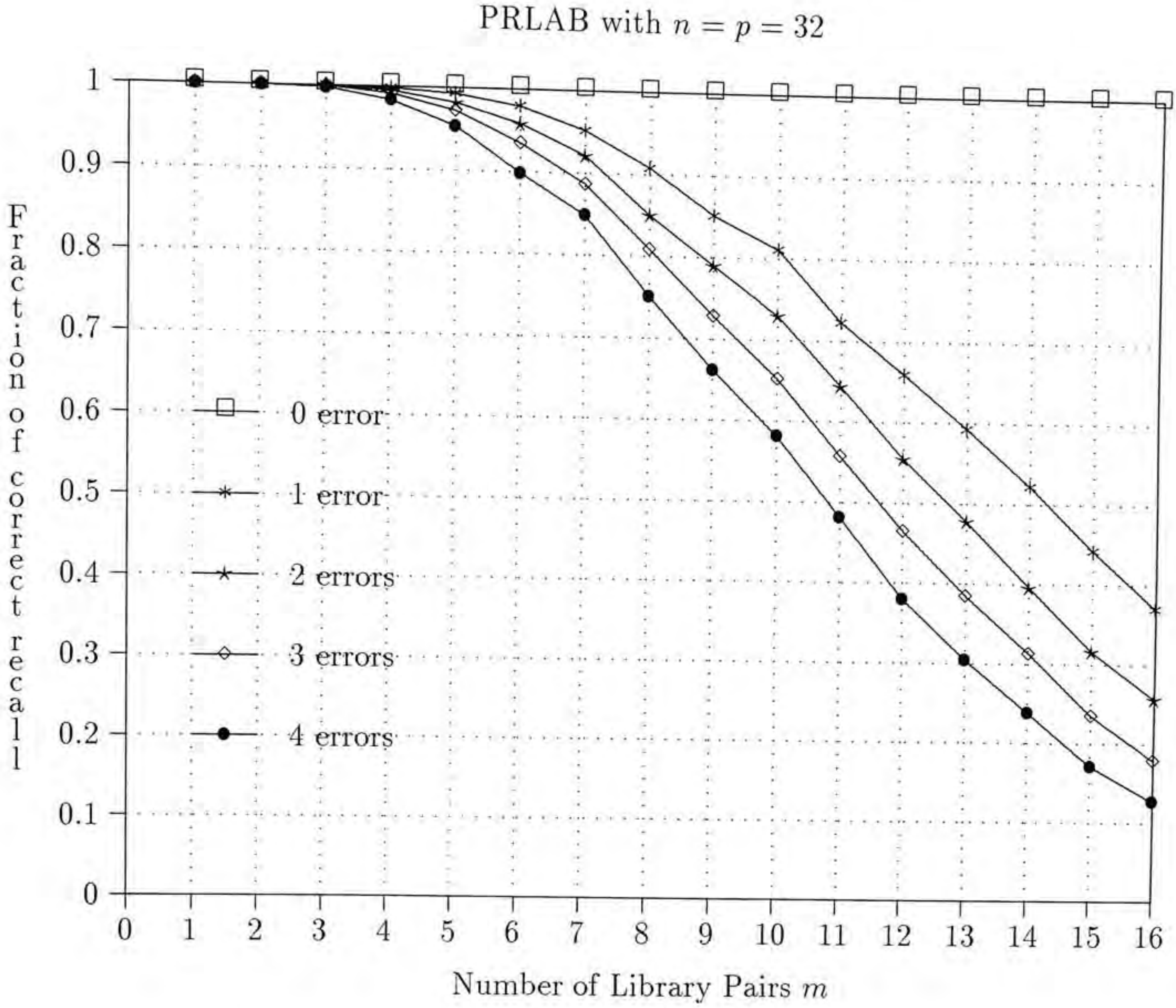


Figure 5.3 The error correction capability of PRLAB. The dimension is $n = p = 32$.

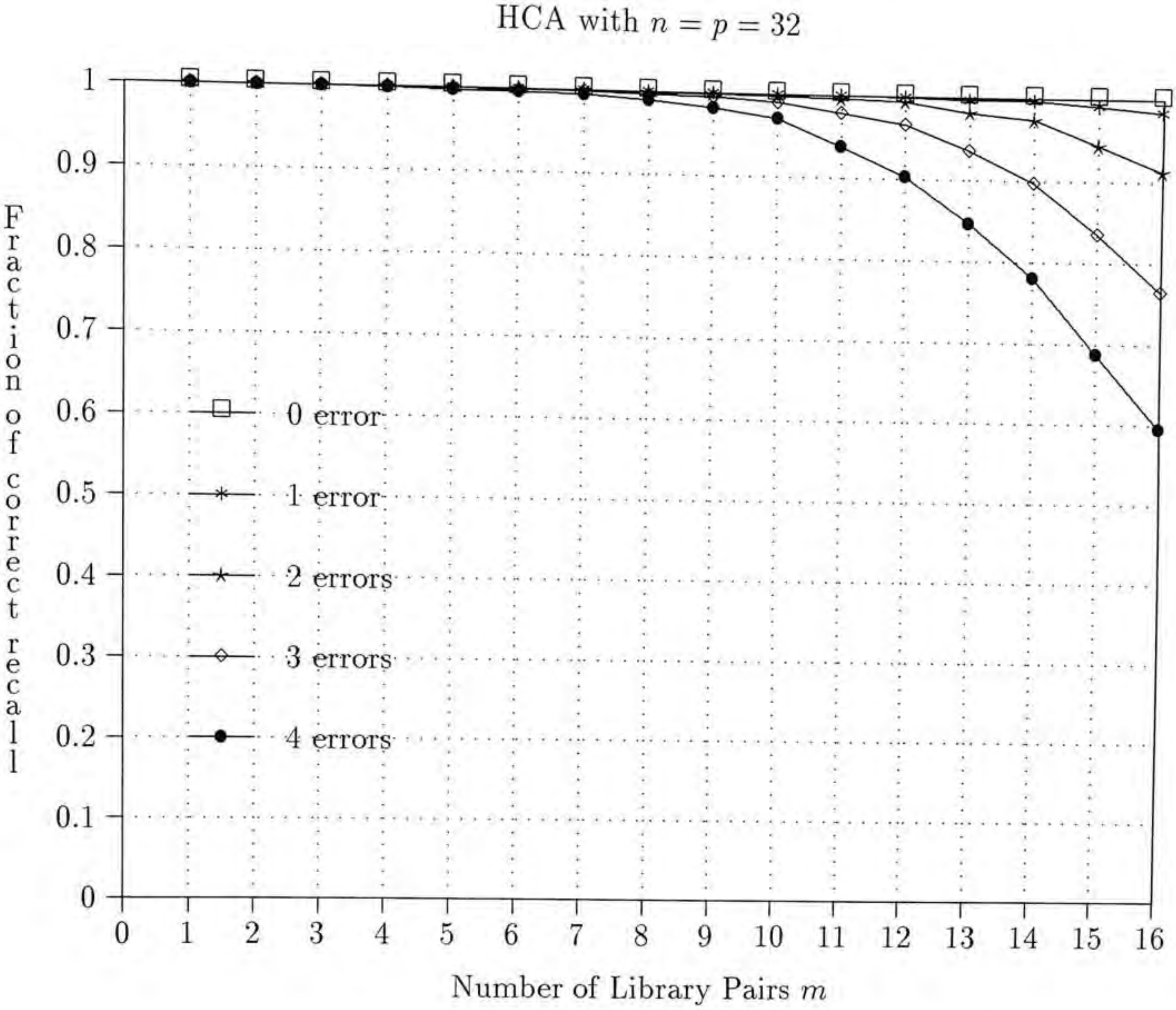


Figure 5.4 The error correction capability of HCA. The dimension is $n = p = 32$.

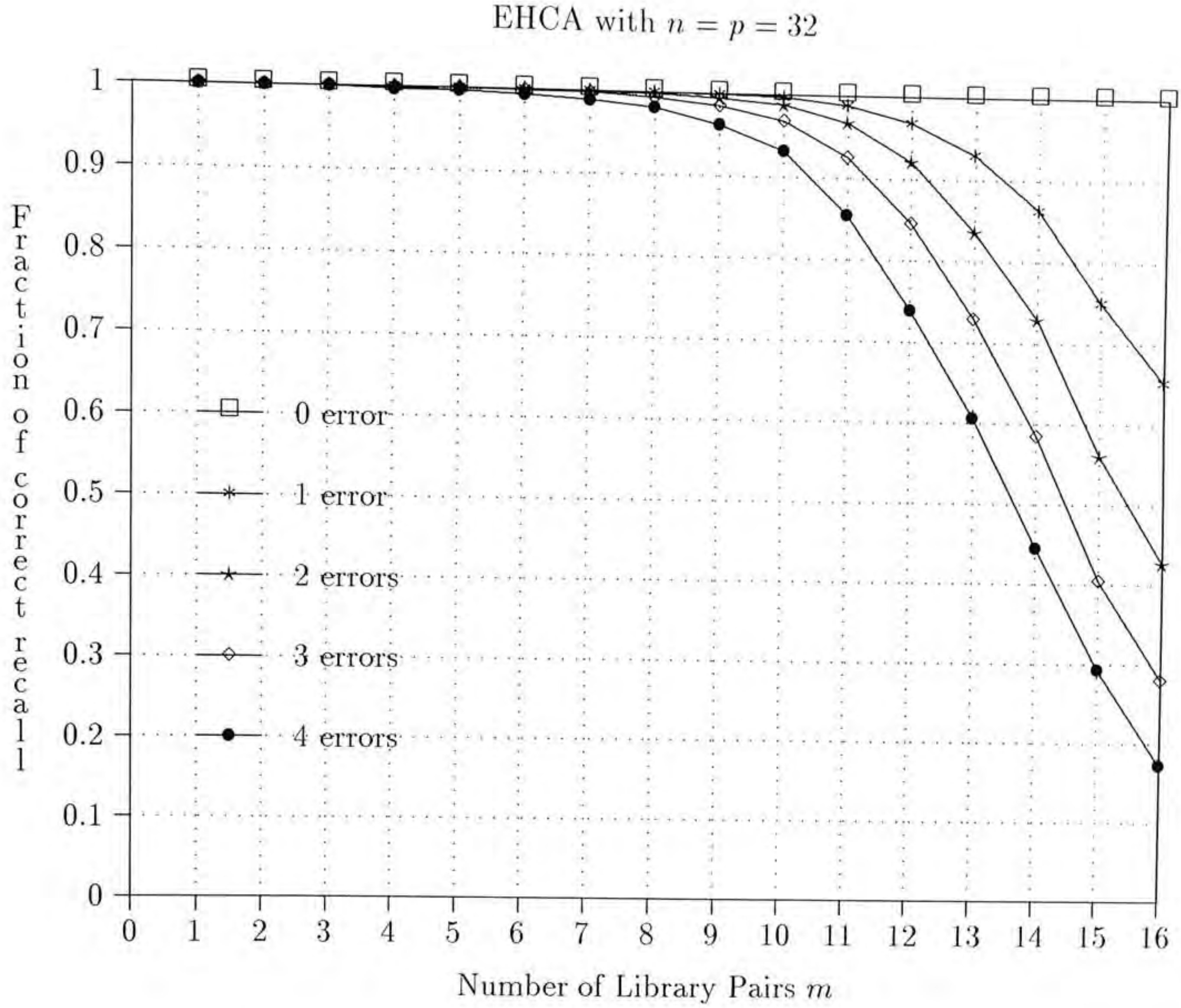


Figure 5.5 The error correction capability of EHCA. The dimension is $n = p = 32$.

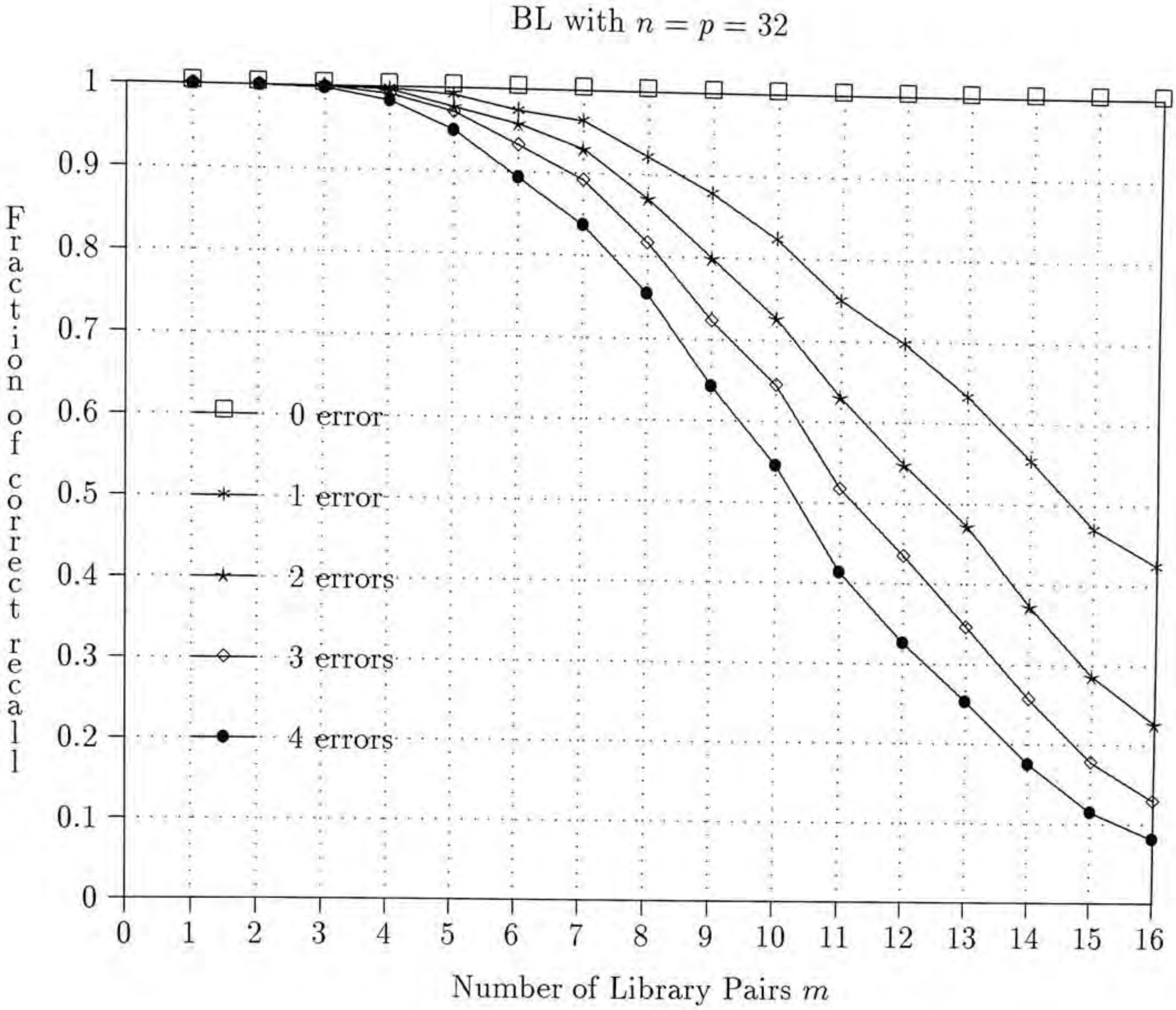


Figure 5.6 The error correction capability of BL. The dimension is

$n = p = 32.$

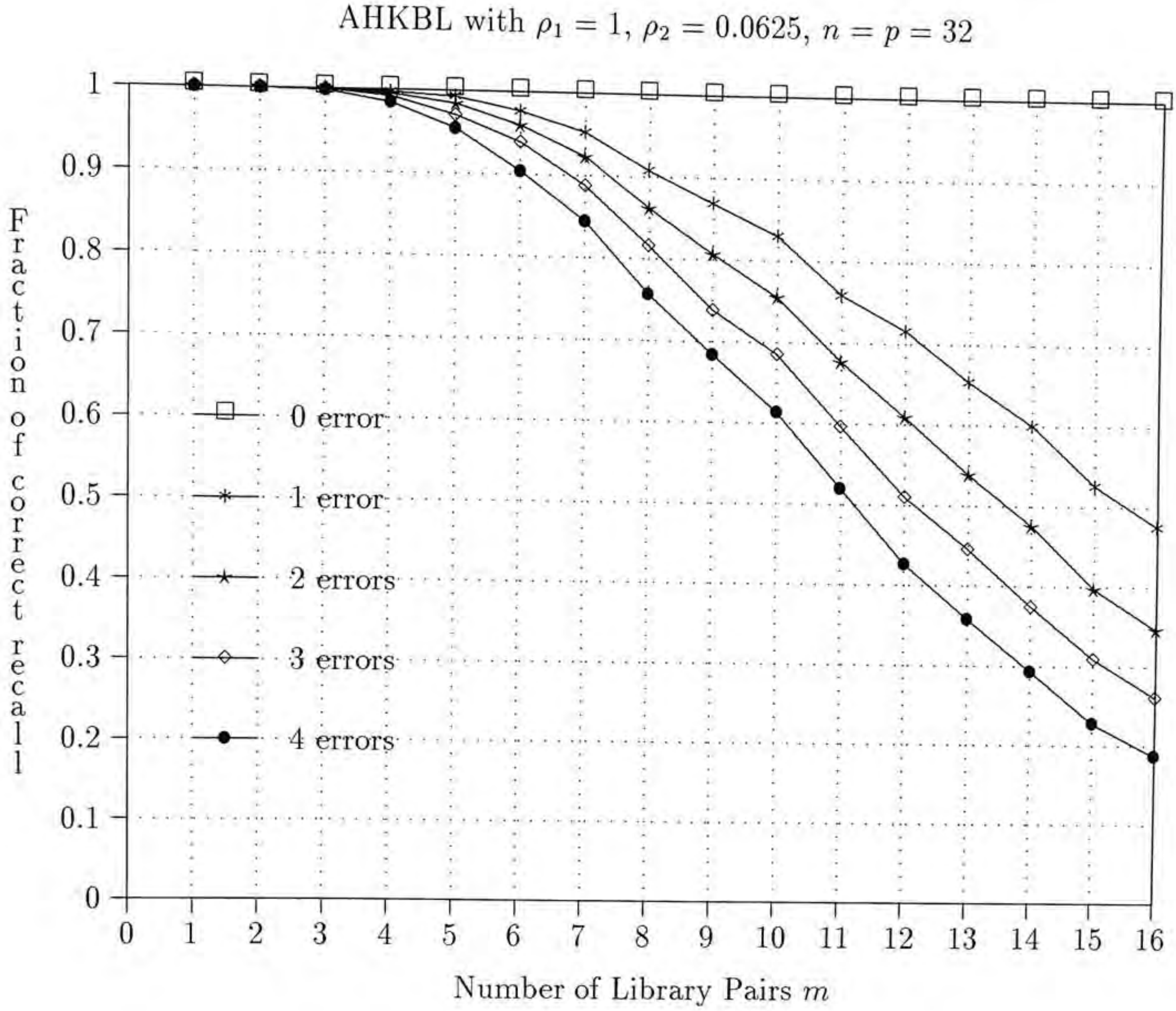


Figure 5.7 The error correction capability of AHKBL, where $\rho_1 = 1$, $\rho_2 = 0.0625$. The dimension is $n = p = 32$.

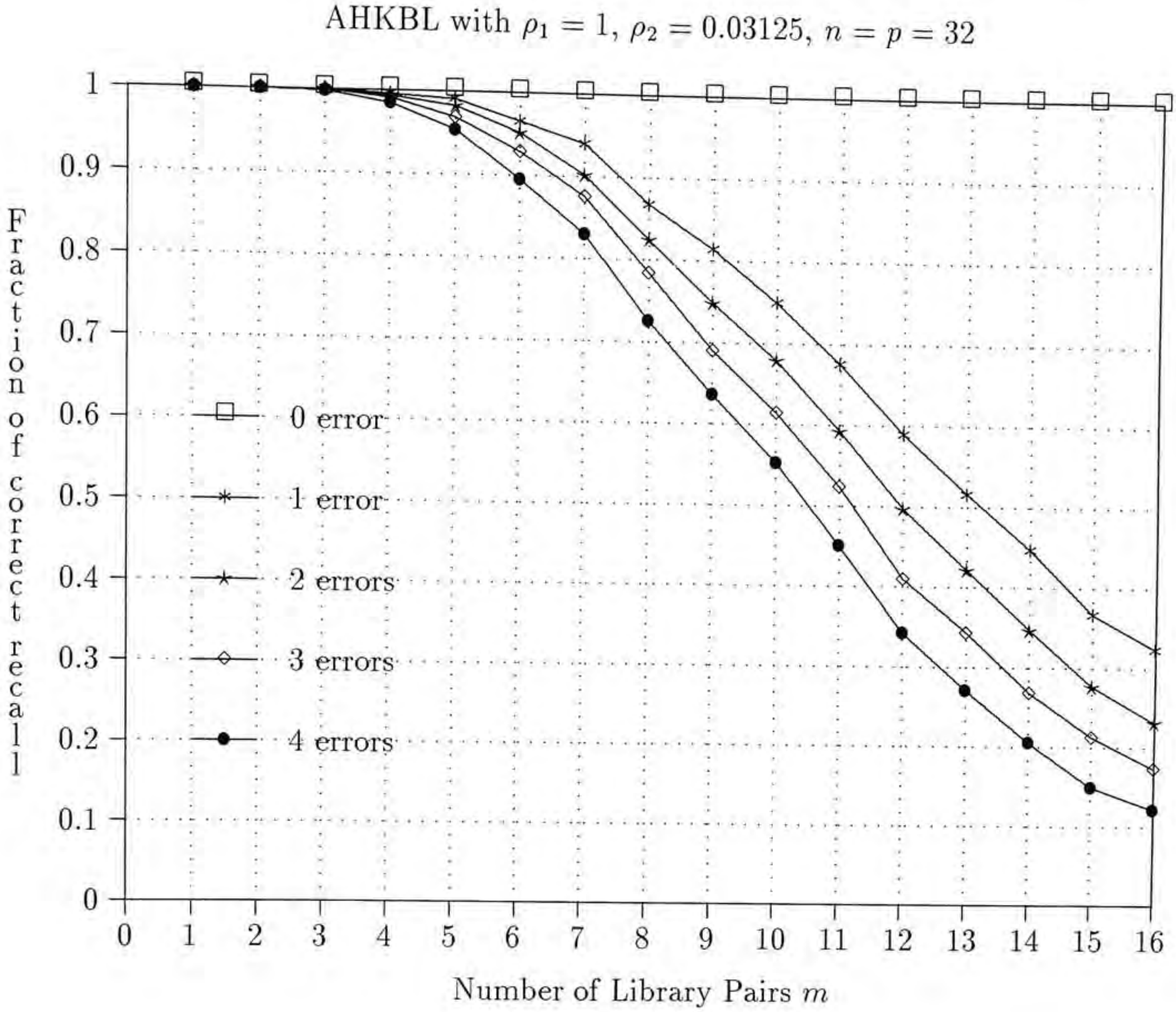


Figure 5.8 The error correction capability of AHKBL, where $\rho_1 = 1$, $\rho_2 = 0.03125$. The dimension is $n = p = 32$.

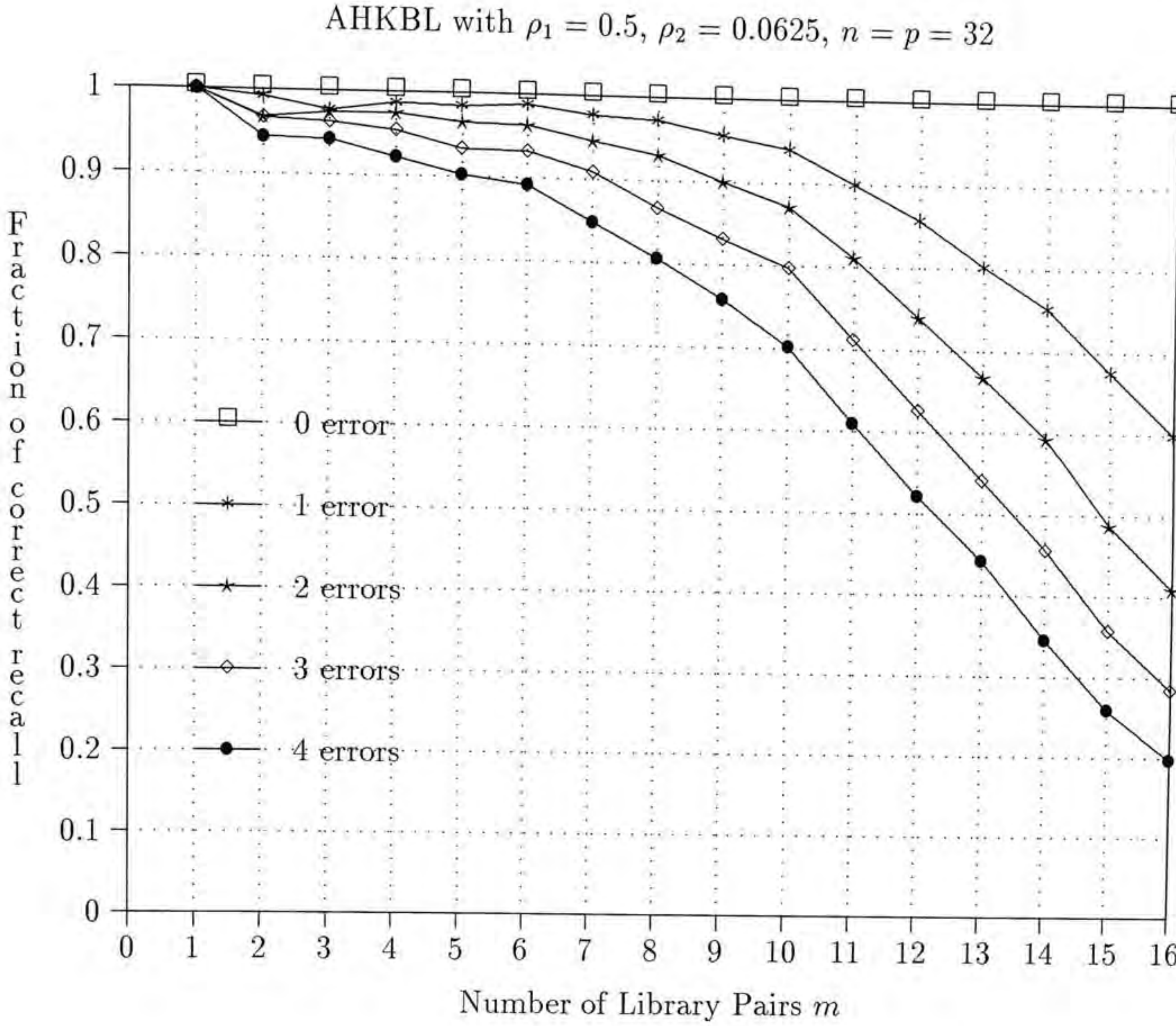


Figure 5.9 The error correction capability of AHKBL, where $\rho_1 = 0.5$, $\rho_2 = 0.0625$. The dimension is $n = p = 32$.

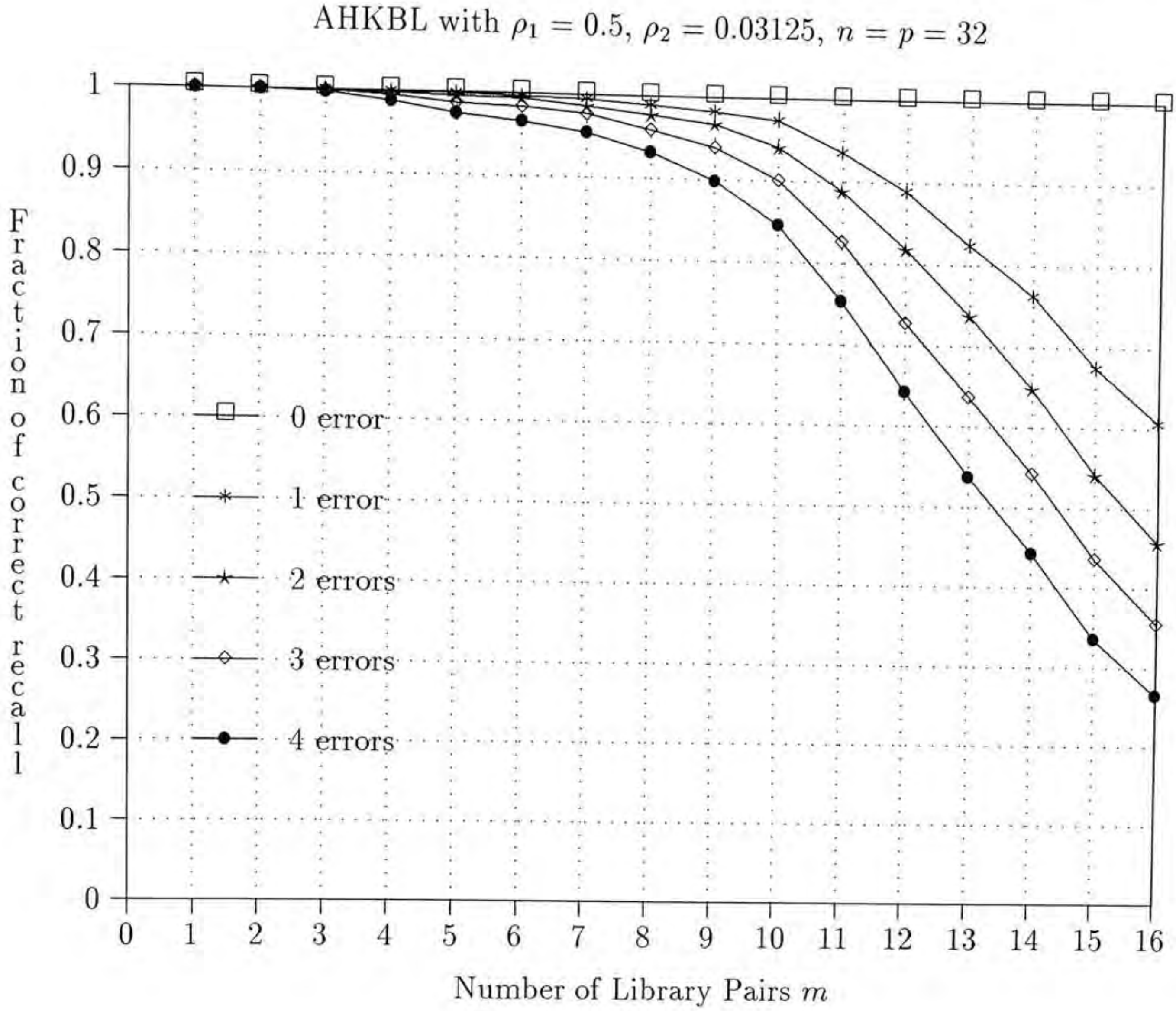


Figure 5.10 The error correction capability of AHKBL, where $\rho_1 = 0.5$, $\rho_2 = 0.03125$. The dimension is $n = p = 32$.

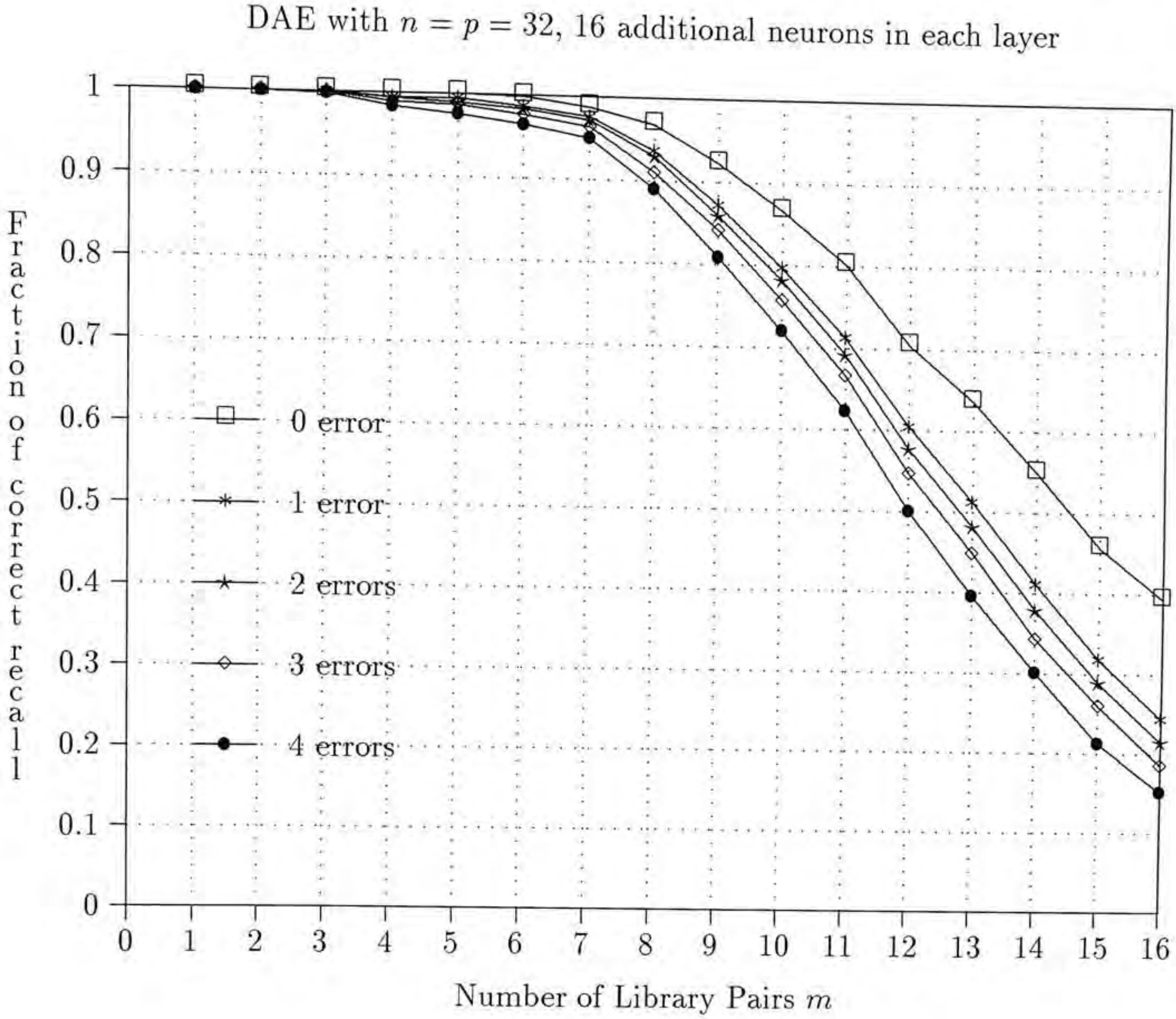


Figure 5.11 The error correction capability of DAE, where the number of additional neurons is 16. The dimension is $n = p = 32$.

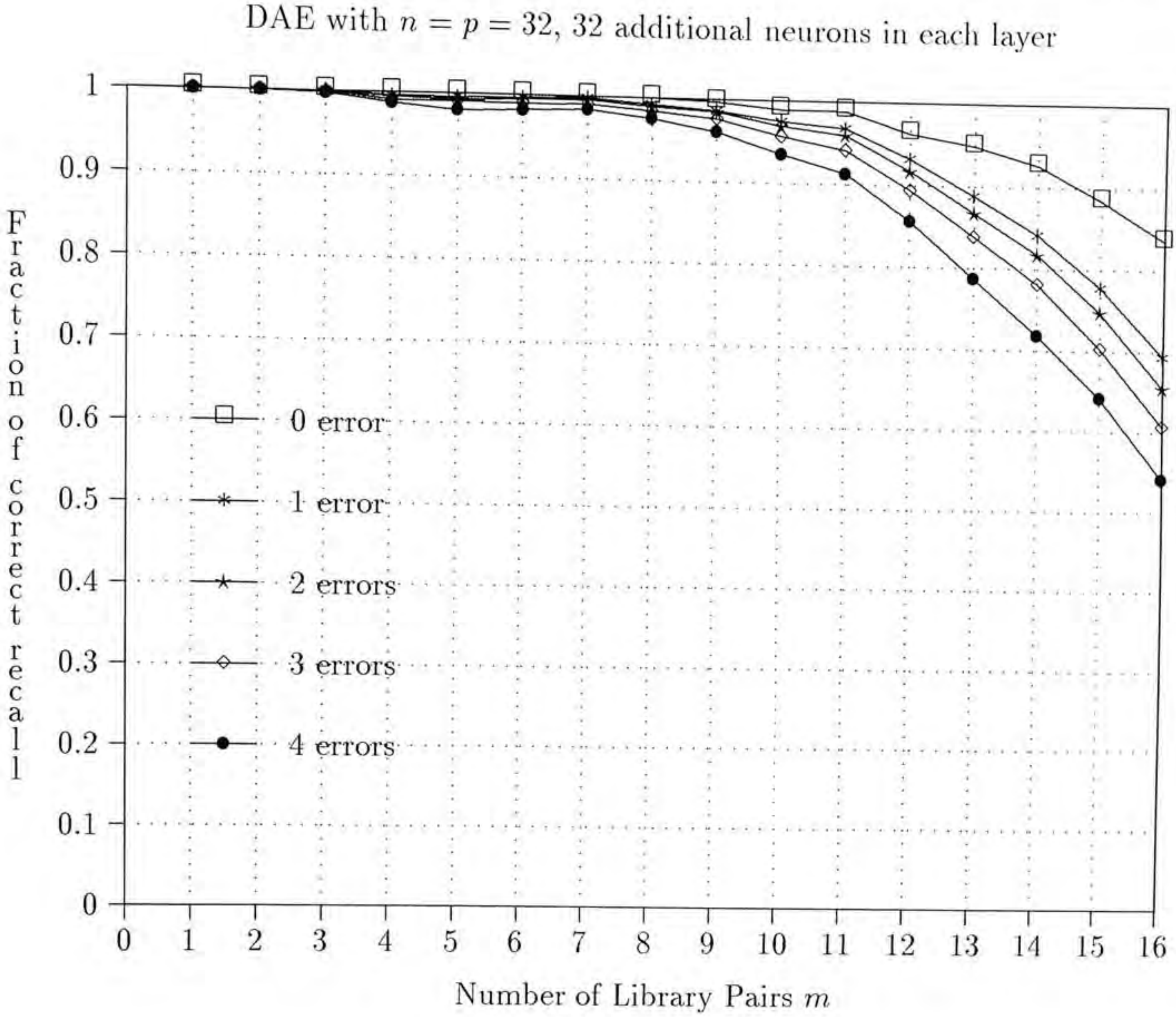


Figure 5.12 The error correction capability of DAE, where the number of additional neurons is 32. The dimension is $n = p = 32$.

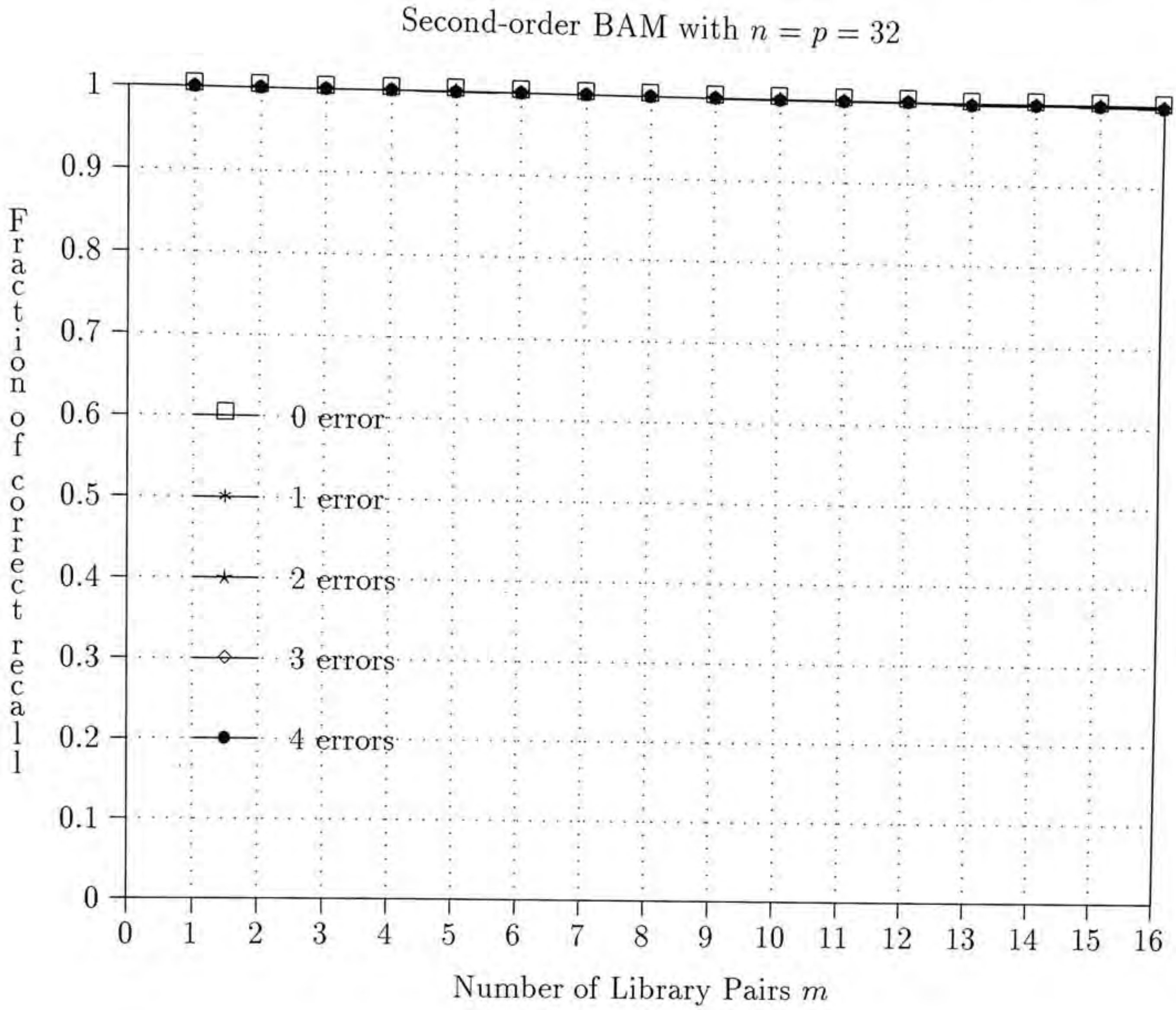


Figure 5.13 The error correction capability of second order BAM. The dimension is $n = p = 32$.

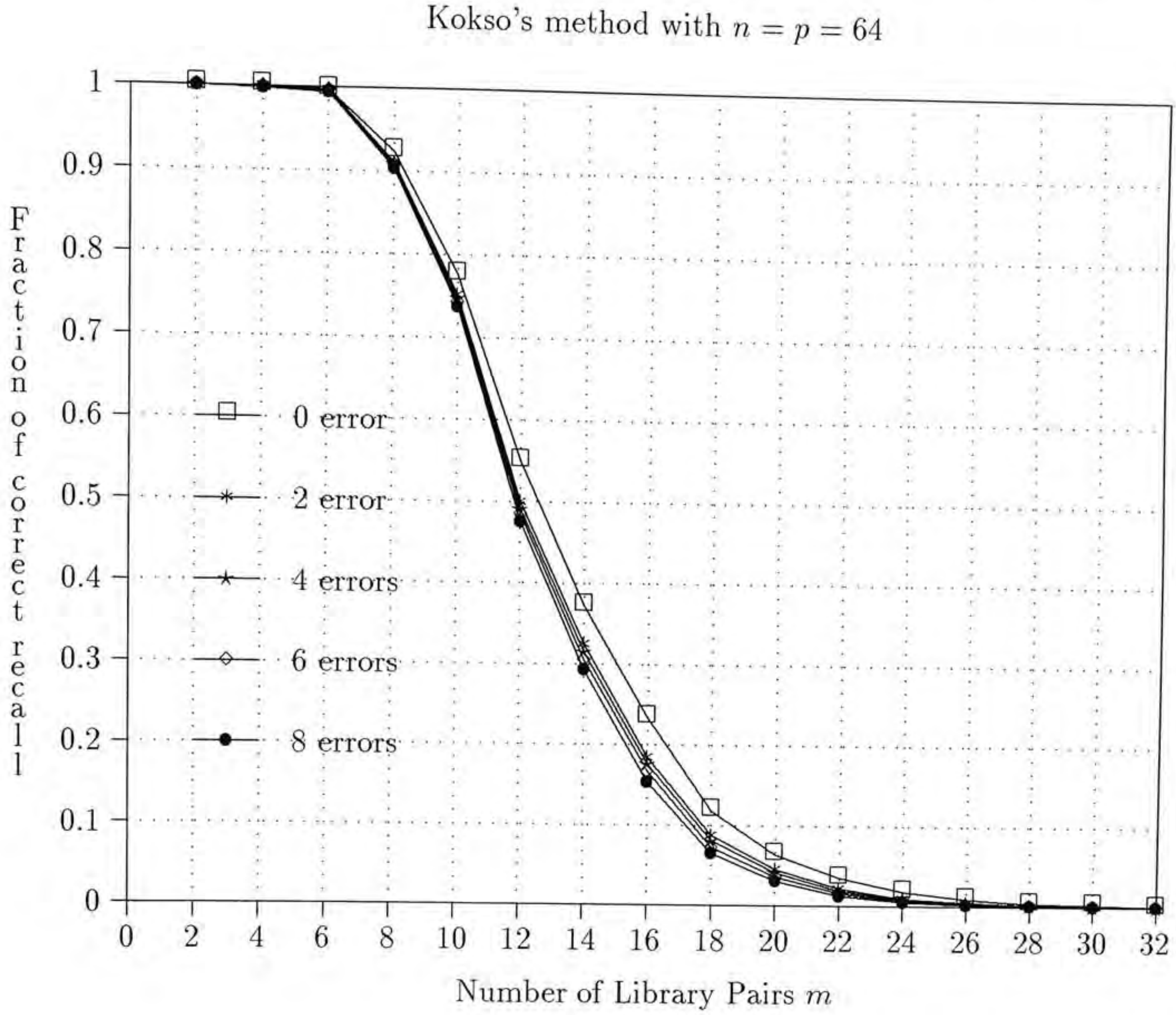


Figure 5.14 The error correction capability of Kosko's encoding method. The dimension is $n = p = 64$.

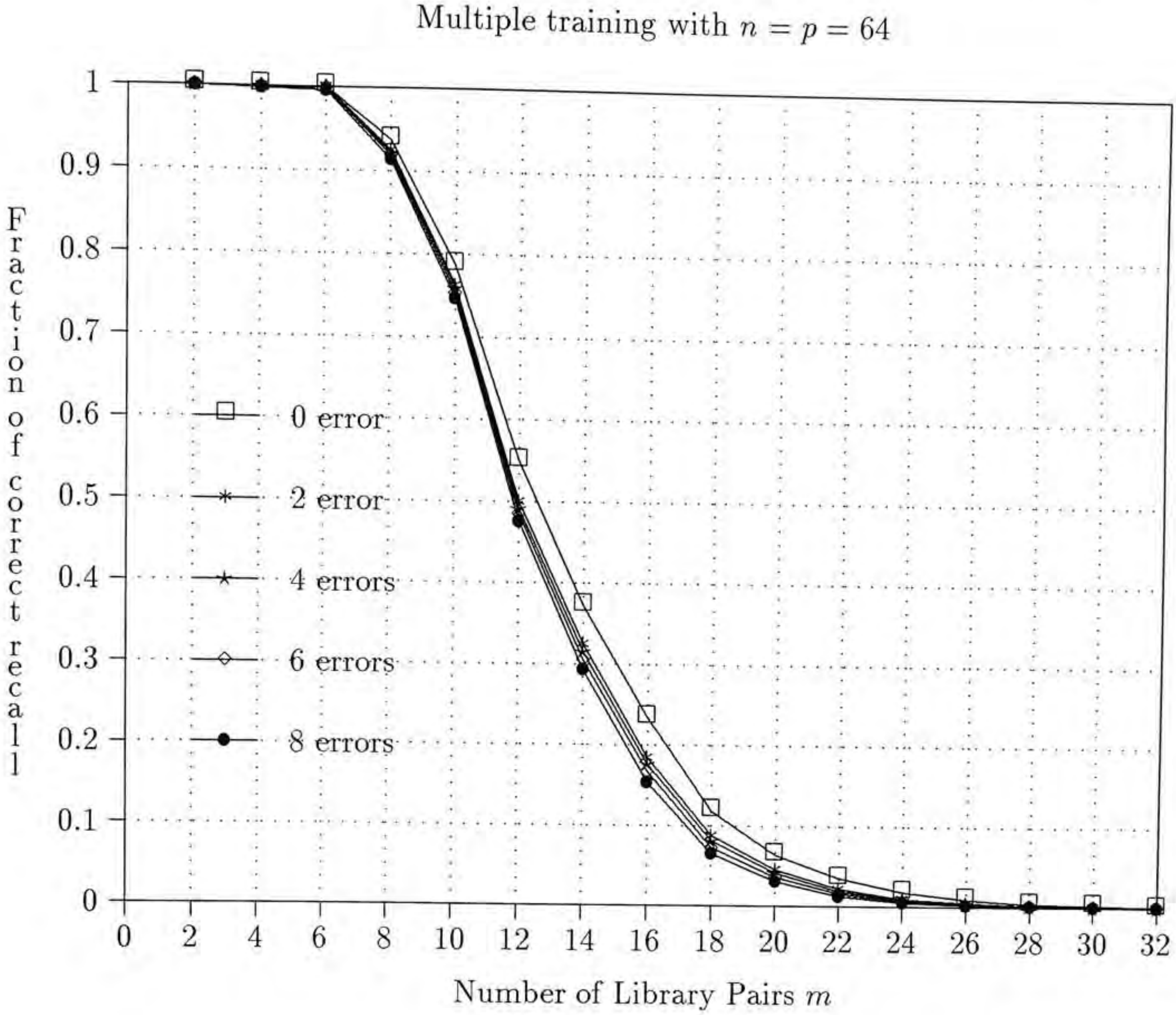


Figure 5.15 The error correction capability of multiple training. The dimension is $n = p = 64$.

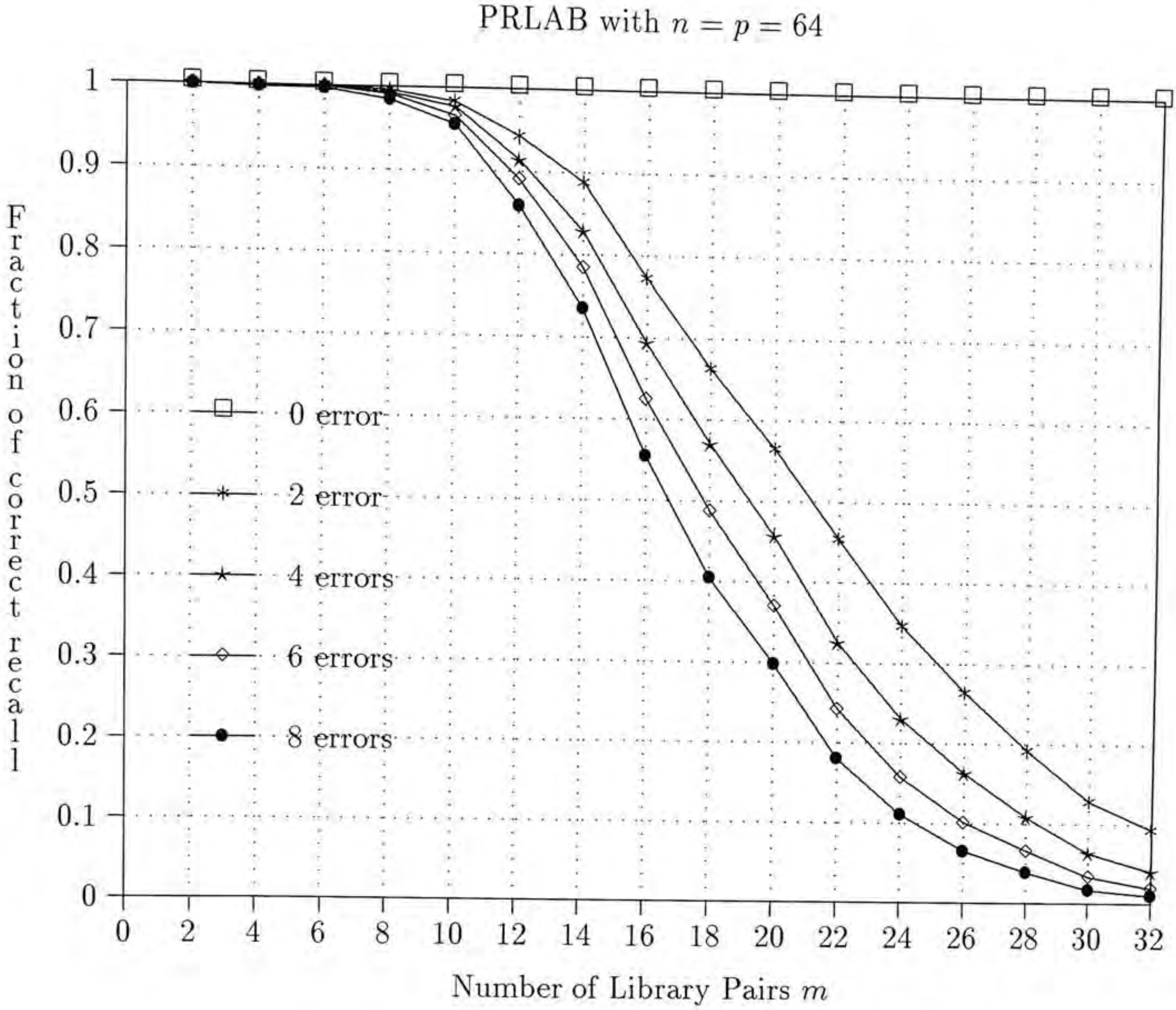


Figure 5.16 The error correction capability of PRLAB. The dimension is $n = p = 64$.

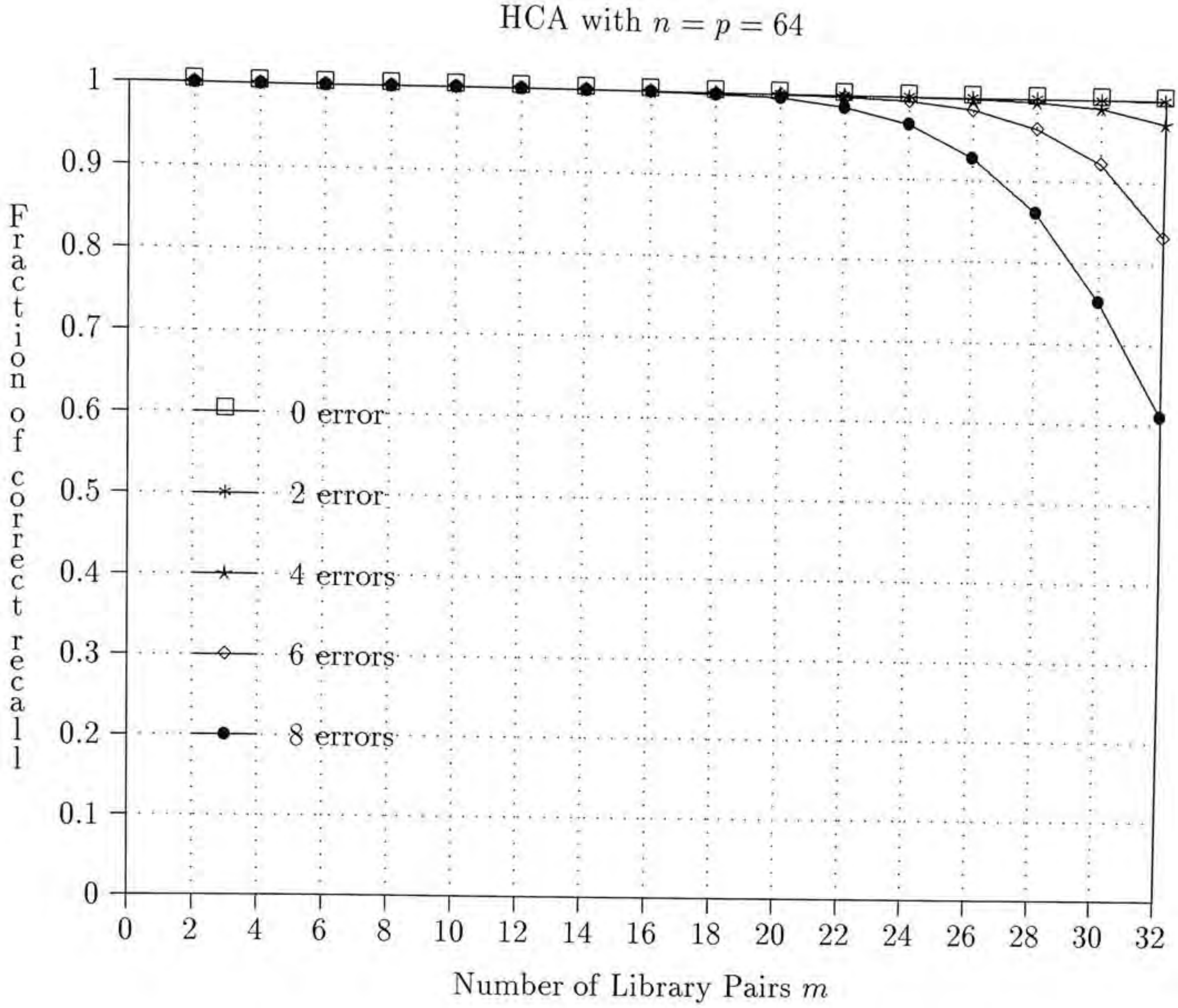


Figure 5.17 The error correction capability of HCA. The dimension is $n = p = 64$.

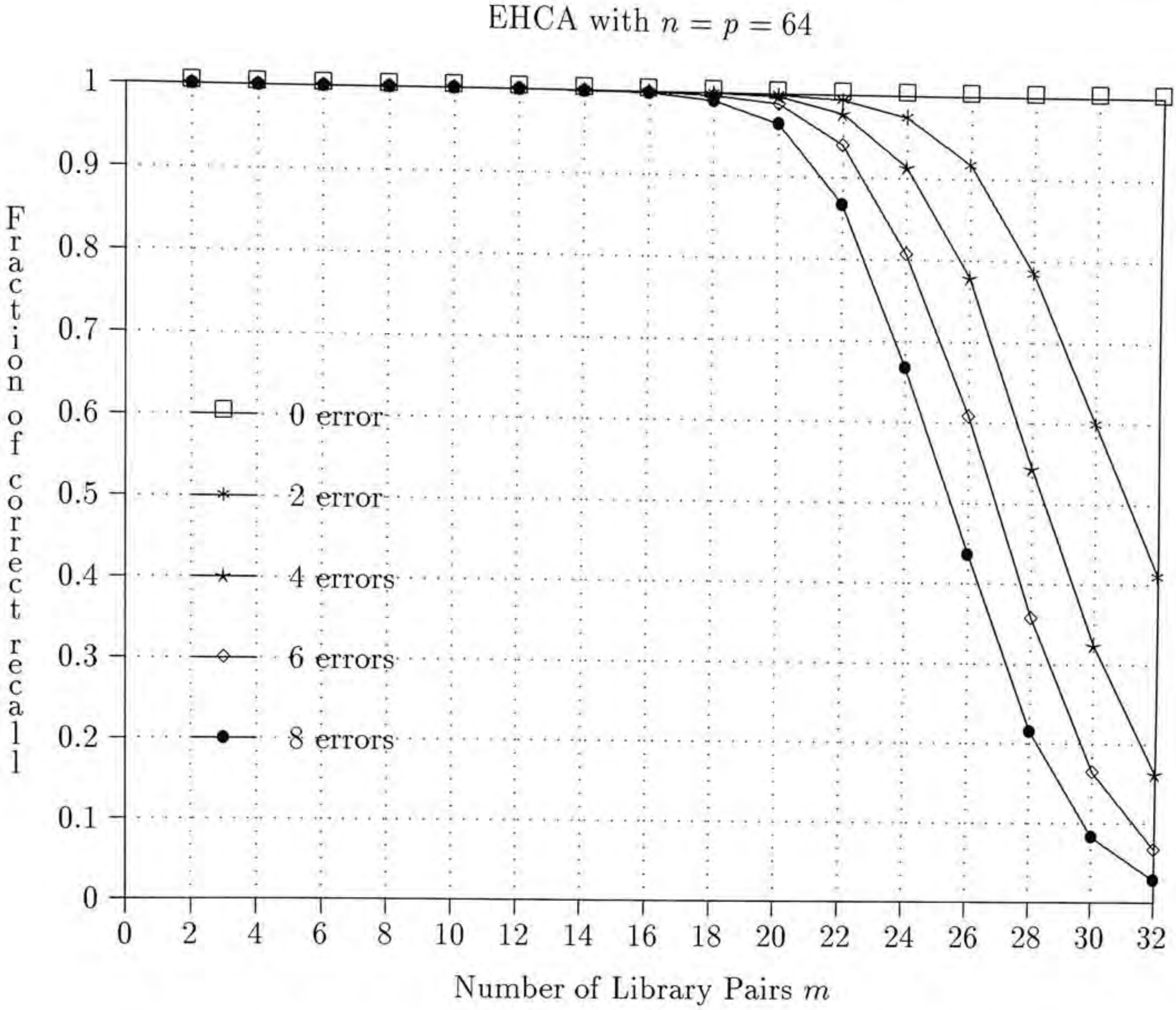


Figure 5.18 The error correction capability of EHCA. The dimension is $n = p = 64$.

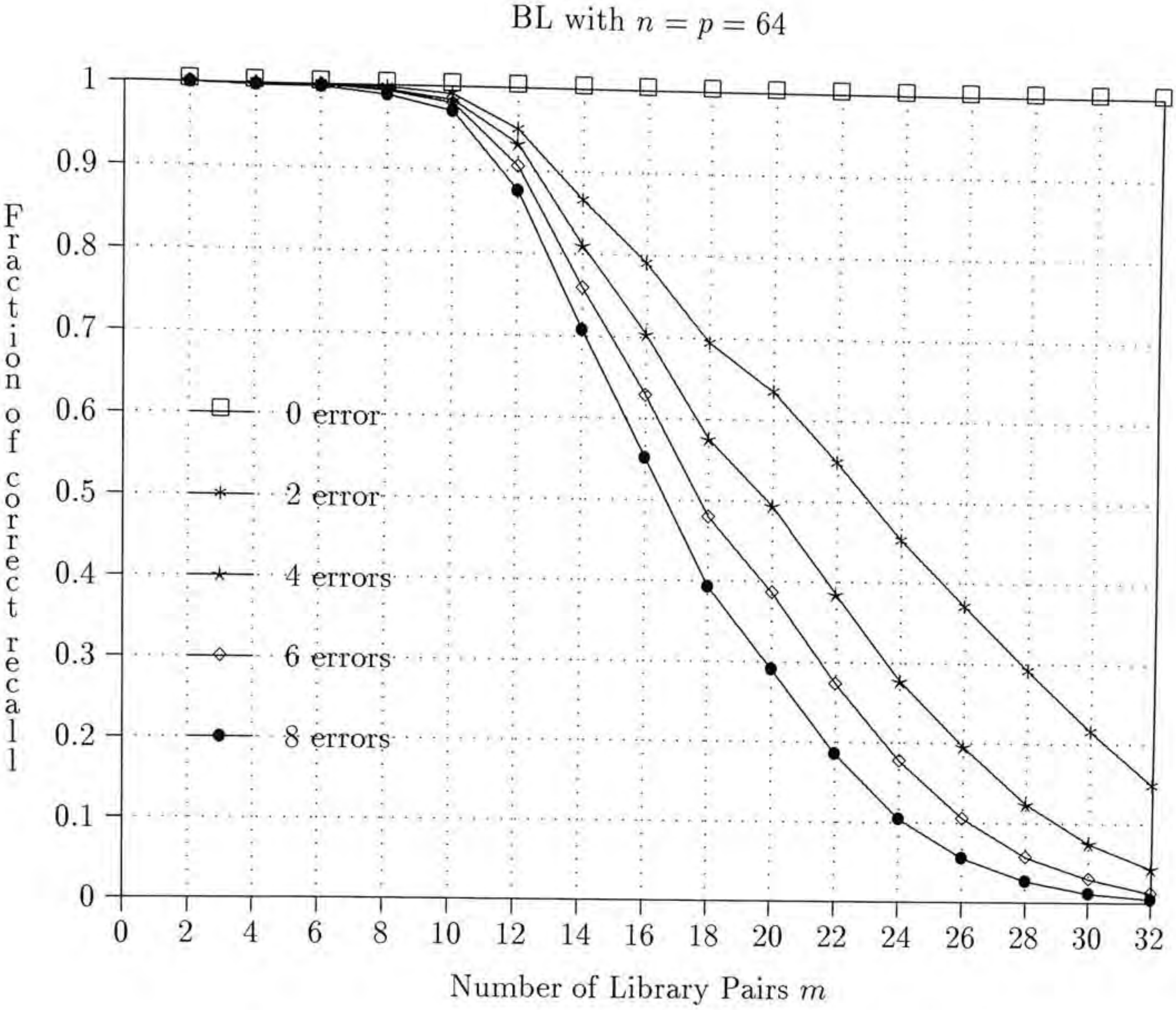


Figure 5.19 The error correction capability of BL. The dimension is $n = p = 64$.

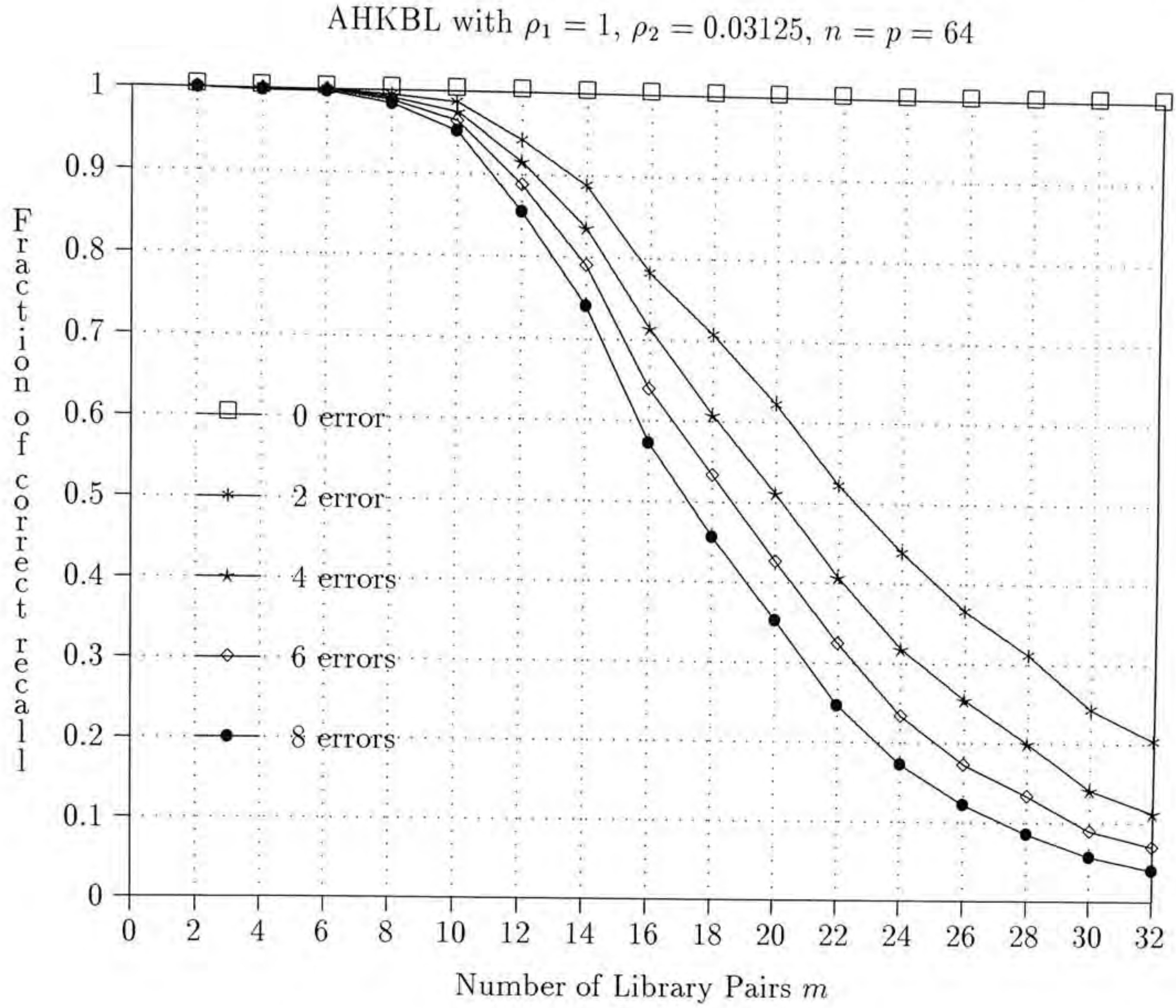


Figure 5.20 The error correction capability of AHKBL, where $\rho_1 = 1$, $\rho_2 = 0.03125$. The dimension is $n = p = 64$.

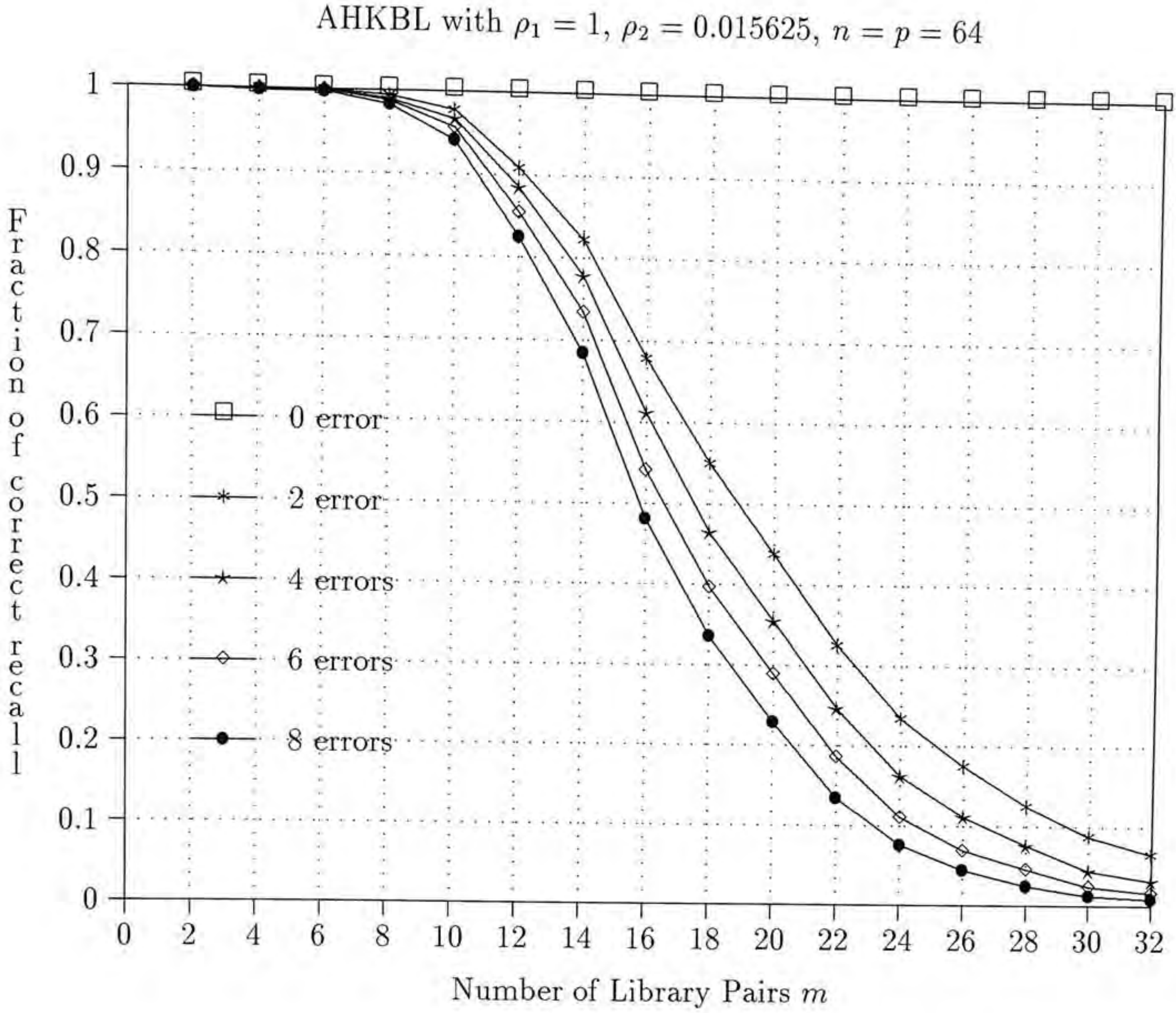


Figure 5.21 The error correction capability of AHKBL, where $\rho_1 = 1$, $\rho_2 = 0.015625$. The dimension is $n = p = 64$.

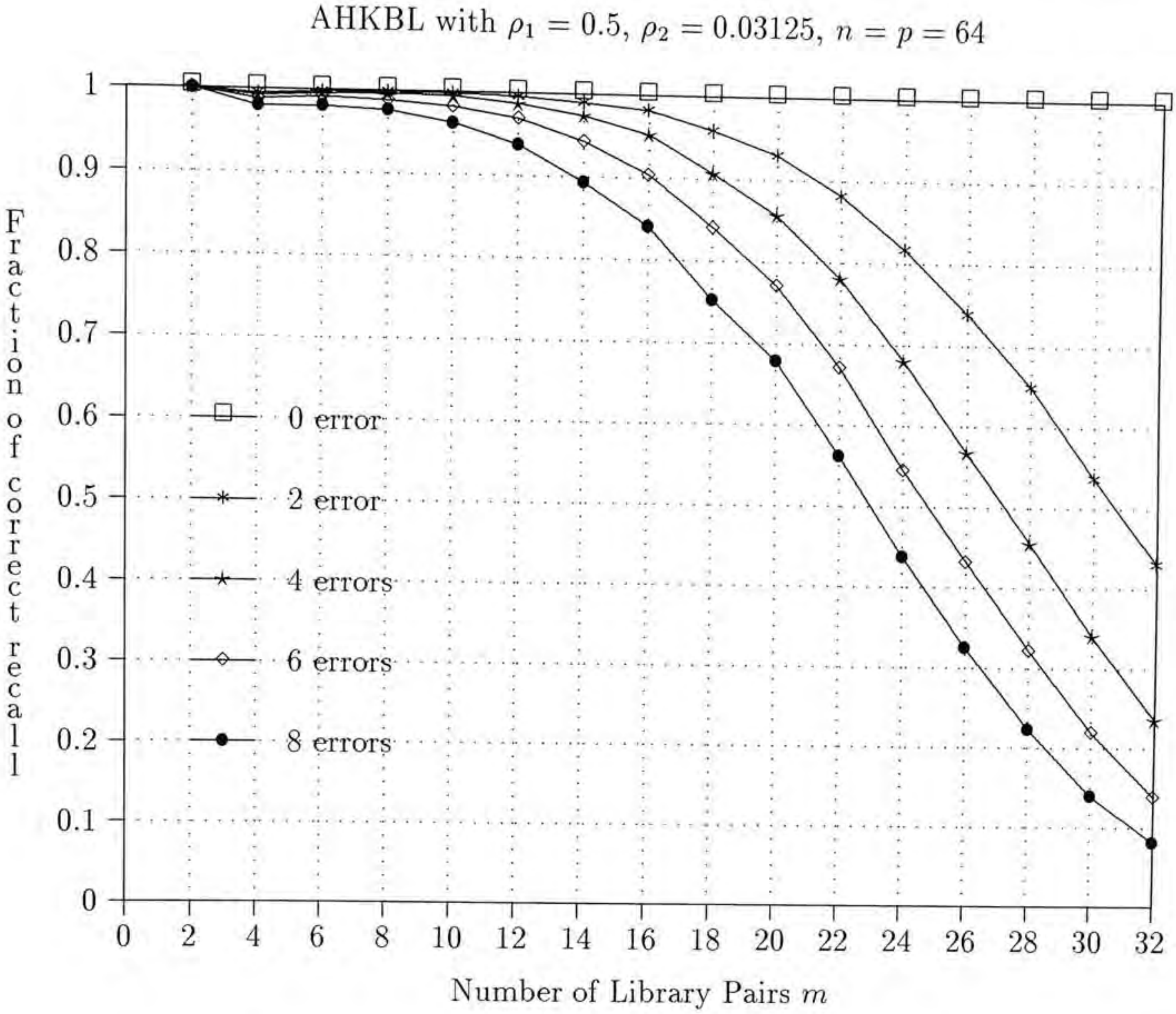


Figure 5.22 The error correction capability of AHKBL, where $\rho_1 = 0.5$, $\rho_2 = 0.03125$. The dimension is $n = p = 64$.

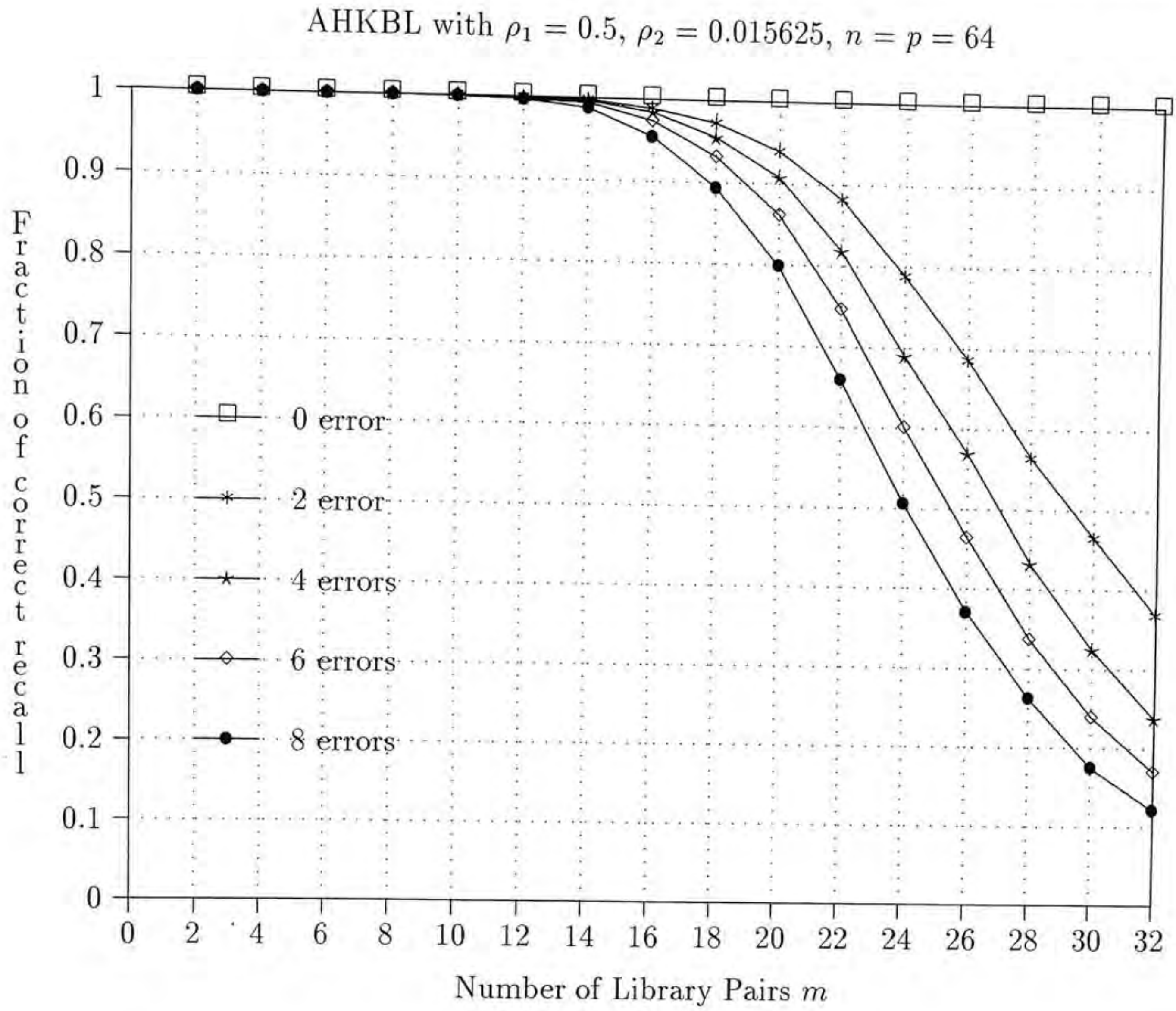


Figure 5.23 The error correction capability of AHKBL, where $\rho_1 = 0.5$, $\rho_2 = 0.015625$. The dimension is $n = p = 64$.

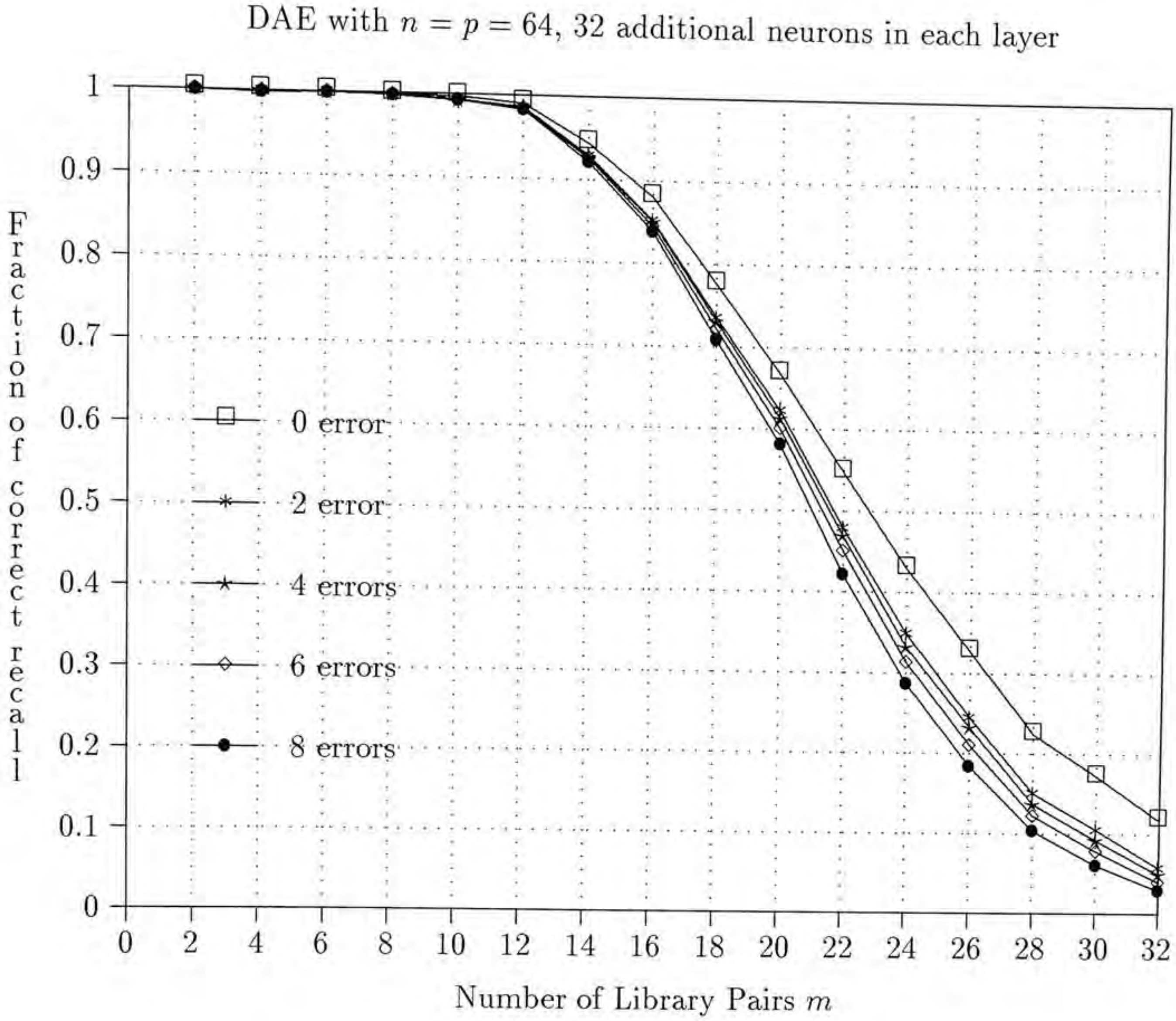


Figure 5.24 The error correction capability of DAE, where the number of additional neurons is 32. The dimension is $n = p = 64$.

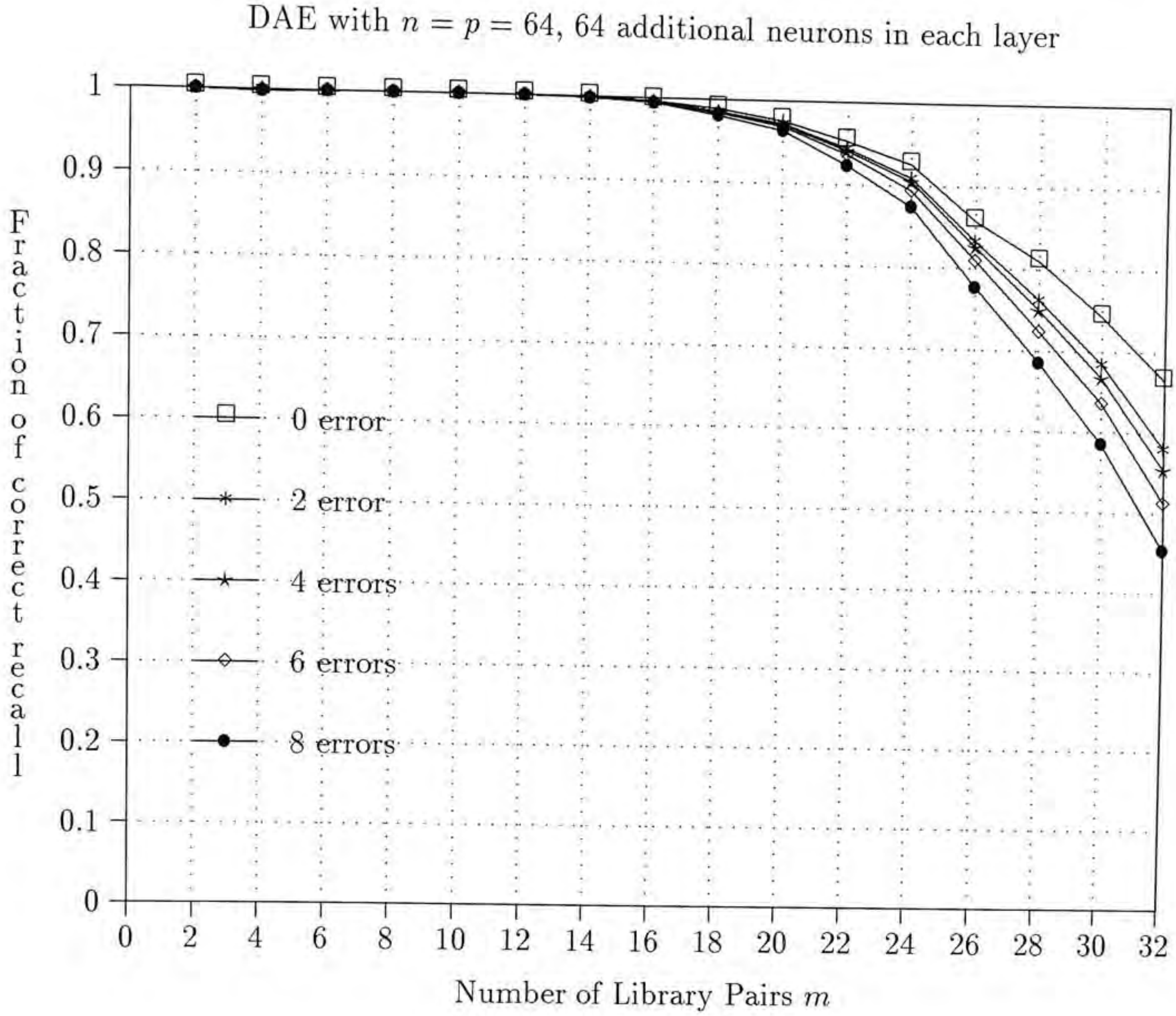


Figure 5.25 The error correction capability of DAE, where the number of additional neurons is 64. The dimension is $n = p = 64$.

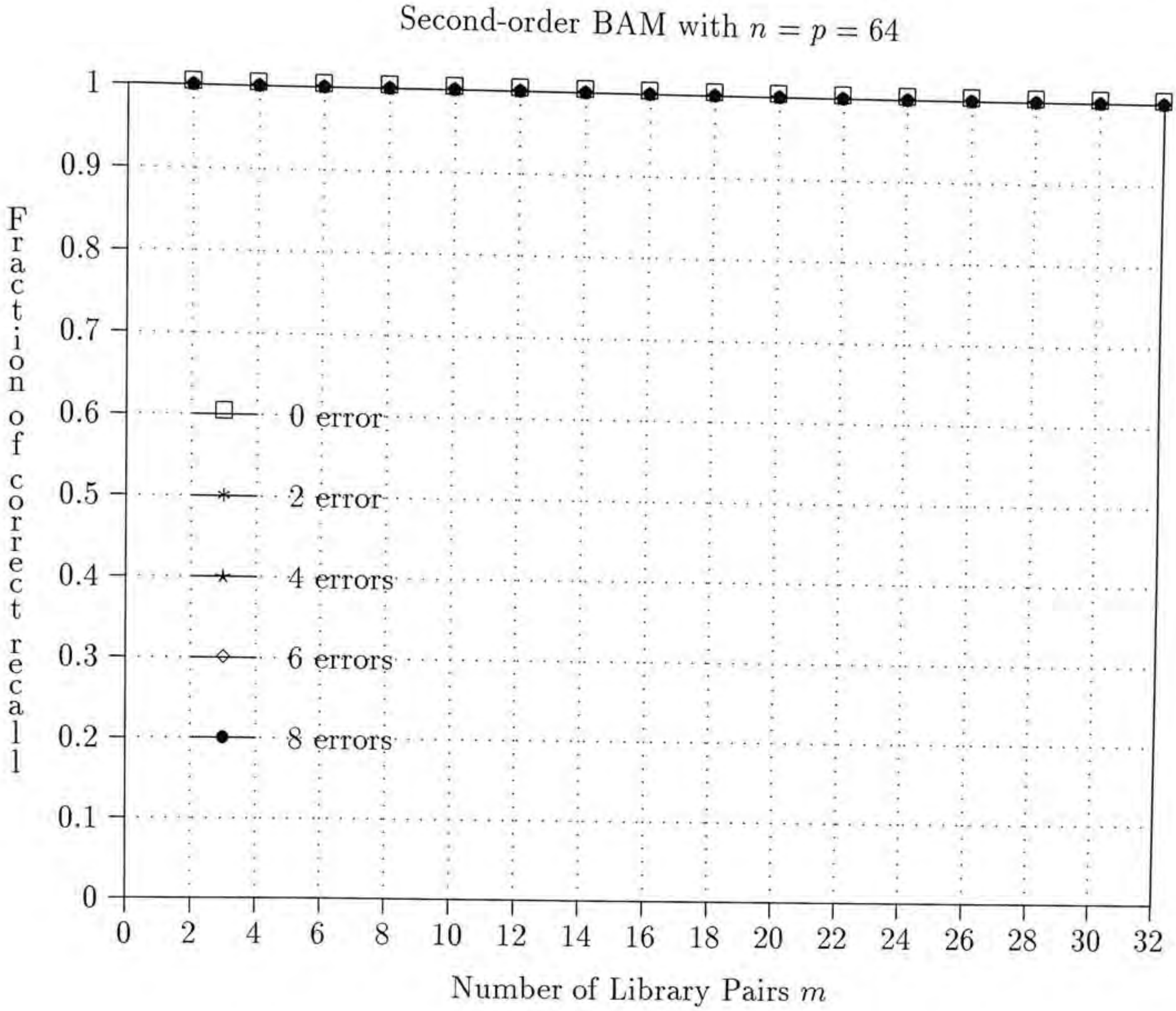


Figure 5.26 The error correction capability of second order BAM. The dimension is $n = p = 64$.

5.7.3 Learning Speed

In the above simulations, we also record the number of learning cycles required by PRLAB, BL, and AHKBL. These results are shown in Table 5.5 and Table 5.6. The average number of learning cycles increases when the number of library pairs increases. The PRLAB has the faster learning speed. In the AHKBL, the average number of learning cycles increases as the learning parameter ρ_1 decreases. For example, in the case of $n = p = 32$, ρ_1=0.5, $\rho_2 = \frac{2}{n} = 0.0625$, the mean value is 36.58. When large learning parameter ρ_1=1, $\rho_2 = \frac{2}{n} = 0.0625$ is used in the AHKBL, the mean value decreases to 9.87.

From the two tables, there does not exist a general conclusion about the effect of ρ_2 on the average number of learning cycles. In the case of $\rho_1 = 1$, the mean value decreases as a larger ρ_2 is used. However, when $\rho_1 = 0.5$, we cannot make a general conclusion about the effect of ρ_2 on the learning speed from the two tables. From the tables, the average number of learning cycles of AHKBL is smaller than that of BL when large learning parameters ρ_1=1 and $\rho_2 = \frac{2}{n}$ in the AHKBL are used.

Table 5.5 Learning speed of PRLAB BL, and AHKBL where dimensions $n = p = 32$

Average No. of learning cycle						
m No. of library pairs	PRLAB	BL	AHKBL $\rho_1 = 1$ $\rho_2 = 0.0625$	AHKBL $\rho_1 = 1$ $\rho_2 = 0.03125$	AHKBL $\rho_1 = 0.5$ $\rho_2 = 0.0625$	AHKBL $\rho_1 = 0.5$ $\rho_2 = 0.03125$
1	1	1	1	1	1	1
2	1	1	1	1	2.04	1
3	1	1	1	1	3.33	1
4	1	1	1	1	4.51	1
5	1.01	1.22	1.06	1.09	5.66	1.09
6	1.06	2.11	1.12	1.3	7.39	1.82
7	1.15	3.37	1.29	2.19	9.07	3.36
8	1.45	4.76	2.12	4.29	11.1	6.11
9	1.73	5.93	3.10	6.39	12.63	9.40
10	1.92	7.04	4.08	8.24	14.86	14.10
11	2.30	7.65	4.80	10.33	17.69	19.98
12	2.45	7.85	5.80	12.44	20.75	25.82
13	2.74	8.49	6.72	14.52	22.28	30.09
14	2.86	9.79	7.65	17.03	27.87	36.97
15	3.12	11.35	8.60	20.25	32.90	43.03
16	3.27	12.66	9.87	22.73	36.58	47.57

Table 5.6 Learning speed of PRLAB BL ,and AHKBL where dimensions $n = p = 64$

Average No. of learning cycle						
m No. of library pairs	PRLAB	BL	AHKBL $\rho_1 = 1$ $\rho_2 = 0.03125$	AHKBL $\rho_1 = 1$ $\rho_2 = 0.015625$	AHKBL $\rho_1 = 0.5$ $\rho_2 = 0.03125$	AHKBL $\rho_1 = 0.5$ $\rho_2 = 0.015625$
2	1	1	1	1	2.06	1
4	1	1	1	1	3.85	1
6	1	1	1	1	5.33	1
8	1.01	1.19	1.02	1.04	6.90	1.21
10	1.13	2.26	1.24	1.79	8.95	2.68
12	1.5	5.84	2.05	4.31	10.74	6.33
14	1.85	9.43	3.15	6.23	12.69	10.91
16	2.08	10.11	4.11	7.81	15.40	14.78
18	2.30	9.78	5.12	9.68	17.24	19.52
20	2.58	10.32	5.91	11.06	22.09	26.76
22	2.88	11.76	6.71	13.07	25.18	31.02
24	3.02	12.27	7.94	15.51	30.24	37.07
26	3.23	13.45	9.25	17.19	36.73	41.09
28	3.39	16.00	10.87	19.95	46.19	45.86
30	3.69	18.46	12.39	22.37	53.60	53.39
32	3.90	20.93	14.26	25.57	59.97	59.43

Remark: There is a trade-off between the learning speed and the error correction capability in the AHKBL. Using small learning parameters, we get a better performance but a longer training time is required. Conversely, with large learning parameters, we get a relatively poor performance but the training time is shorter. We have done other several simulations for different values of learning parameters. Based on the simulations, our recommendation of learning parameters for the AHKBL is ρ_1=0.5, and $\rho_2 = \frac{1}{\max(n,p)}$ which compromises the trade-off.

5.8 Chapter Summary

There are two approaches of modifications on the BAM: 'change of encoding method' and 'change of topology'. From the approach of 'change of encoding method', we have proposed four learning rules for the BAM. The concepts of the four learning rules for the BAM have been described and shown to have a significant improvement over the Kosko's encoding method and the multiple training on the memory capacity and the error correction capability. The properties of each learning rules have also been presented.

Besides, we have also compared our four encoding methods with other existing approaches in different aspects: stability during recalling, ease of hardware implementation, information ratio, memory capacity, error correction capability and learning speed. Table 5.7 is the summary of their properties. From Table 5.7, we can choose the model which better meets our preference.

Table 5.7 Summary of the properties of modifications of BAM.

	Kosko's method	multiple training	PRLAB	HCA	EHCA	BL	AHKBL $\rho_1 = 0.5$ $\rho_2 = \frac{1}{\max(n,p)}$	DAE	second-order BAM
stability during recalling	Yes	Yes	Yes	unknown	Yes	Yes	Yes	Yes	No
hardware implement-ation	integer weight	integer weight	real weight	real weight	real weight	integer weight	real weight	integer weight	integer weight
information ratio	low	low	high	high	high	high	high	low	very low
memory capacity	low	low	high	high	high	high	high	depends on No. of additional neurons	very high
error correction capability	low	low	fair	high	high	fair	high	depends on No. of additional neurons	very high
learning speed	NA	NA	fast	NA	NA	fair	low	NA	NA

Chapter 6

BAM under Forgetting Learning

Forgetting learning is one kind of incremental learning in associative memory. With it, the recent learning patterns can be recalled and the old learning patterns will be forgotten. The storage behavior of BAM under the forgetting learning is studied here. That is, "Can the last k-th previous library pair be stored as a fixed point ?". Also, we discuss the way to choose the forgetting constant in the forgetting learning, such that the number of most recent library pairs being stored as fixed points is nearly maximal.

6.1 Introduction

So far, we have discussed several design techniques for the BAM in Chapter 5. These techniques, except Kosko's method and HCA, cannot handle the case of encoding a new library pair based on the current connection matrix. In the Kosko's method and HCA, the new library pair can be encoded based on the current connection matrix only. Hence, the Kosko's method and HCA has the ability of incremental learning (i.e. the ability of adding new library pairs into the model). However, the Kosko's method and HCA can store effectively only up to $\frac{\min(n,p)}{2\log\min(n,p)}$ and $\min(n,p)$, respectively. When the number of library pairs exceeds these numbers, the library pairs, including the old library pairs and the recent library pairs, may not be stored as fixed points. In addition to incremental learning (as defined above), another desirable feature of an associative memory is 'forgetting' (i.e. the ability of deleting old library pairs which have been encoded long time ago).

Forgetting learning can encode the recent library pairs and delete the old library pairs. In the simple form of forgetting learning, when a new library pair (X_t, Y_t) (the t-th library pair) is encoded, the connection matrix is updated as

$$W^{(t)} = \alpha_f W^{(t-1)} + Y_t X_t^T \tag{6.1}$$

where $W^{(0)}$ is a zero matrix and $\alpha_f \in (0,1)$ is called the forgetting constant. One may view the above equation (6.1) as a discrete version of adaptive BAM (ABAM) [13]. There is another approach about forgetting learning [76]. This is to forget a particular library pair. However, in such approach we need an additional device to store all the previous library pairs otherwise we do not know what the particular library pair is.

In (6.1), the correction of the connection matrix is based on the current connection and the current library pair only. Hence, an additional device is not necessary. With (6.1), the BAM can be regard as an open-end-pipe. The new library pair is put into the pipe at the entrance and the old library pair is taken away from the end of the pipe. The forgetting constant determines the length of the pipe. Up to now, it is not clearly known theoretically how many most recent library pairs can be stored as fixed points and how the value of forgetting constant is chosen. Under some assumptions made in the next section, we will prove the following theorem.

Theorem 6.1 *Under the forgetting learning, if the BAM is trained with t library pairs and k is less than*

$$\min\left(\frac{\log\frac{(1-\alpha_f^2)n}{2\log n}}{2\log\frac{1}{\alpha_f}}, \frac{\log\frac{(1-\alpha_f^2)p}{2\log p}}{2\log\frac{1}{\alpha_f}}\right), \tag{6.2}$$

then the probability that the $(t-k)$-th library pair (X_{t-k}, Y_{t-k}) is stored as a fixed point tends to 1, as $n \to \infty$ and $p \to \infty$.

From Theorem 6.1, one can determine the number of most recent library pairs being stored as fixed points. Also, the better value of α_f can be determined such that the number of most recent library pairs being stored as fixed points is nearly maximal.

6.2 Properties of Forgetting Learning

The following assumptions and notations are used.

- The dimensions (n and p) are large and $p = rn$, where r is a positive constant.
- Each component of the library pairs (X_h, Y_h) is a ± 1 equiprobable independent random variable.
- $EU_{j,t-k}$ is the event that the j-th component of $\text{sgn}(W^{(t)} X_{t-k})$ is equal to the j-th component of Y_{t-k}. Also, $\overline{EU}_{j,t-k}$ is the complement event of $EU_{j,t-k}$.
- $EV_{i,t-k}$ is the event that the i-th component of $\text{sgn}({W^{(t)}}^T X_{t-k})$ is equal to the i-th component of Y_{t-k}. $\overline{EV}_{i,t-k}$ is the complement event of $EV_{i,t-k}$.

We begin with the introduction of two lemmas. Only the proof of the first one is given since the last one can be obtained in a similar way.

Lemma 6.1 *The probability $Prob(\overline{EU}_{j,t-k})$ is less than*

$$\exp\left\{-\frac{(1-\alpha_f^2)\alpha_f^{2k} n}{2}\right\}$$

for $j = 1, \ldots, p$.

Proof of Lemma 6.1

Without loss of generality, we consider that the library pair (X_{t-k}, Y_{t-k}) has all components positive: $X_{t-k} = (1, \ldots, 1)^T$ and $Y_{t-k} = (1, \ldots, 1)^T$. Then, $\text{Prob}(\overline{EU}_{j,t-k})$ is

$$\text{Prob}\left(\sum_{h=1, h\neq t-k}^{t} \alpha_f^{t-h} \frac{S_{j,h}}{\sqrt{n}} > \alpha_f^k \sqrt{n}\right) \tag{6.3}$$

where

$$S_{j,h} = y_{jh} \sum_{i=1}^{n} x_{ih} \ .$$

Note that the distribution of $\frac{S_{j,h}}{\sqrt{n}}$ tends to standard normal as $n \to \infty$. Also, the sum of normal random variables is a normal random variable. Hence, $\text{Prob}(\overline{EU}_{j,t-k})$ is

$$Q\left(\frac{\alpha_f^k \sqrt{n}}{\sqrt{\sum_{h=0, h\neq t-k}^{t-1} \alpha_f^{2h}}}\right) \leq Q\left(\sqrt{1-\alpha_f^2}\,\alpha_f^k \sqrt{n}\right)$$

where

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{x^2}{2}\right\} dx .$$

For large x [77],

$$Q(x) \le \exp\left\{-\frac{x^2}{2}\right\} .$$

Hence, we can conclude that

$$\mathrm{Prob}(\overline{EU}_{j,t-k}) \le \exp\left\{-\frac{(1-\alpha_f^2)\alpha_f^{2k} n}{2}\right\}$$

for $j = 1, \cdots, p$. □

Lemma 6.2 *The probability Prob($\overline{EV}_{i,t-k}$) is less than*

$$\exp\left\{-\frac{(1-\alpha_f^2)\alpha_f^{2k} p}{2}\right\}$$

for $i = 1, \cdots, n$.

With the above two lemmas, Theorem 6.1 can be proven in the following way. We denote the probability that (X_{t-k}, Y_{t-k}) is a fixed point as P_*:

$$\begin{aligned} P_* &= \mathrm{Prob}\left(EU_{1,t-k} \cap \cdots \cap EU_{p,t-k} \cap EV_{1,t-k} \cap \cdots \cap EV_{n,t-k}\right) \\ &= 1 - \mathrm{Prob}\left(\overline{EU}_{1,t-k} \cup \cdots \cup \overline{EU}_{p,t-k} \cup \overline{EV}_{1,t-k} \cup \cdots \cup \overline{EV}_{n,t-k}\right) \\ &\ge 1 - p\mathrm{Prob}\left(\overline{EU}_{1,t-k}\right) - n\mathrm{Prob}\left(\overline{EV}_{1,t-k}\right) . \end{aligned} \quad (6.4)$$

Note that

$$P_* \ne \left(1 - \mathrm{Prob}(\overline{EU}_{1,t-k})\right)^p \left(1 - \mathrm{Prob}(\overline{EV}_{1,t-k})\right)^n \quad (6.5)$$

because the above events are not mutually independent. That can be easily observed by comparing $\overline{EU}_{1,t-k}$ and $\overline{EU}_{2,t-k}$.

With Lemma 6.1 and 6.2, it is easy to prove that if k is less than

$$\min\left(\frac{\log\frac{(1-\alpha_f^2)n}{2\log n}}{2\log\frac{1}{\alpha_f}}, \frac{\log\frac{(1-\alpha_f^2)p}{2\log p}}{2\log\frac{1}{\alpha_f}}\right) = \frac{\log\frac{(1-\alpha_f^2)\min(n,p)}{2\log\min(n,p)}}{2\log\frac{1}{\alpha_f}},$$

then the right-hand side of (6.4) tends to one as $n \to \infty$ (also $p \to \infty$ due to $p = rn$ and r is constant). Thus, Theorem 6.1 is obtained. From Theorem 6.1, we can obtain some properties of the forgetting learning.

Given p and n, define

$$f(\alpha_f) = \frac{\log \frac{(1-\alpha_f^2)\min(n,p)}{2\log\min(n,p)}}{2\log\frac{1}{\alpha_f}}. \tag{6.6}$$

If $\alpha_f \to 0^+$, then $f(\alpha_f) \to 0^+$. That is, as $\alpha_f \to 0^+$, k should be chosen as 0 such that Theorem 6.1 holds. In this case, we can only make the conclusion: the probability that the current library pair is a fixed point tends to one. On the other hand, if $\alpha_f \to 1^-$, then $f(\alpha_f) \to -\infty$. That is : as $\alpha_f \to 1^-$, no positive value of k can be chosen such that Theorem 6.1 holds. In this case, we cannot make any conclusion about the properties of forgetting learning.

In fact, the most interesting point is that what the value of $\alpha_f \in (0,1)$ (denoted as $\alpha_{f,max}$) is such that $f(\alpha_f)$ is maximum. Clearly, $f(\alpha_f)$ is a continuous function between zero and one. Also,

$$\frac{d\,f(\alpha_f)}{d\,\alpha_f}\,|_{\alpha_f=0^+} > 0$$

and

$$\frac{d\,f(\alpha_f)}{d\,\alpha_f}\,|_{\alpha_f=1^-} < 0\ .$$

Hence, $f(\alpha)$ has at least one local maximum point while $0 < \alpha < 1$. A typical plot of $f(\alpha_f)$ is shown in Figure 6.1.

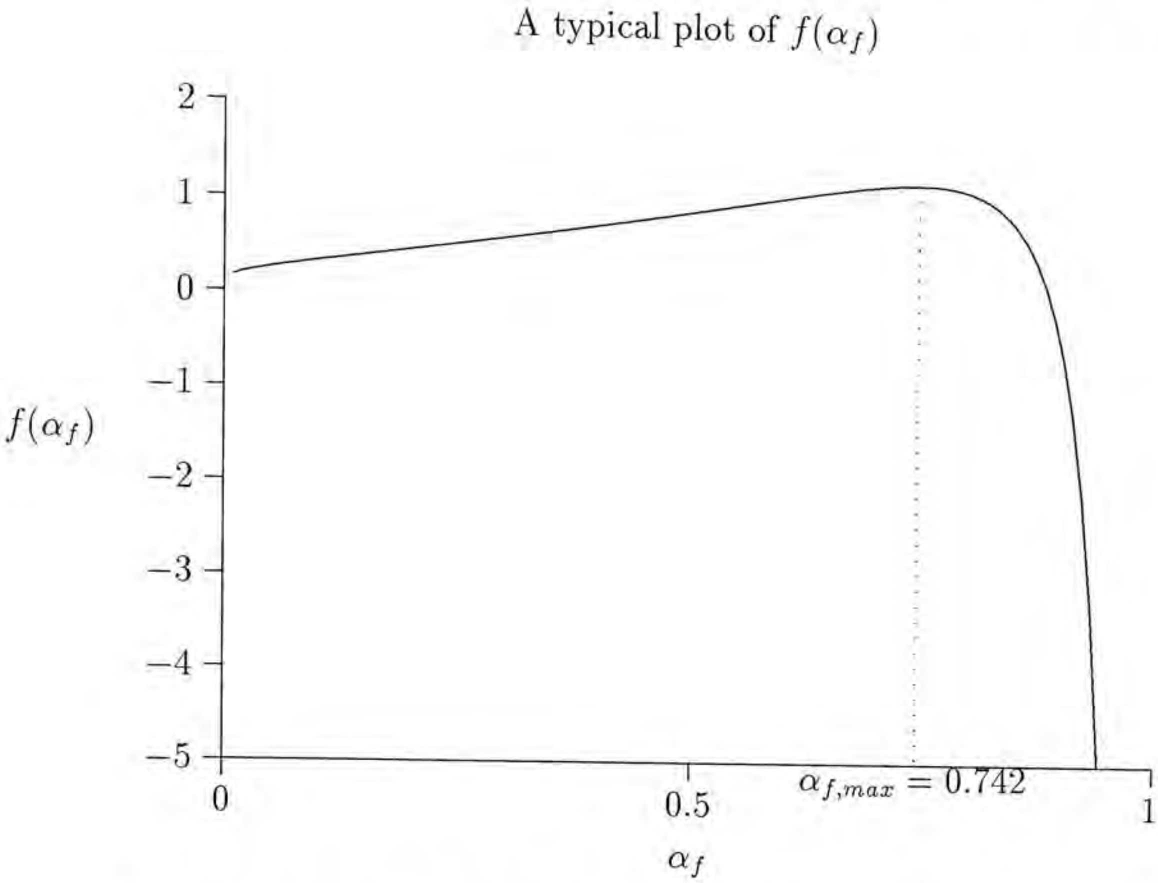


Figure 6.1 A typical $f(\alpha_f)$ where $\alpha_f \in (0,1)$ and $\min(n,p) = 32$.

Using simple numerical method, we obtain Table 6.1 which summarizes the $\alpha_{f,max}$ at different values of $\min(n,p)$.

Table 6.1 Summary of $\alpha_{f,max}$ and $f(\alpha_{f,max})$ found from numerical method.

$\min(n,p)$	$\alpha_{f,max}$	$f(\alpha_{f,max})$
16	0.613	0.6012
32	0.742	1.223
64	0.837	2.3453
128	0.902	4.3611
256	0.943	7.9967
512	0.9675	14.599
1024	0.982	26.673
2048	0.9900	48.906
4096	0.9945	90.079
8192	0.9970	166.721

From the table, as $\min(n,p)$ increases, $\alpha_{f,max}$ increases. For large $\min(n,p)$, $\alpha_{f,max}$ is near to one. Based on this phenomenon, we can further derive the close form solution of $\alpha_{f,max}$ for large $\min(n,p)$:

$$\alpha_{f,max} \approx \sqrt{1 - \frac{2e \log \min(n,p)}{\min(n,p)}} \tag{6.7}$$

Note that (6.7) gives us a guideline how to choose the value of α_f. Table 6.2 shows $\alpha_{f,max}$ at different values of $\min(n,p)$ based on (6.7). Compared Table 6.2 with Table 6.1, (6.7) is a good approximation of $\alpha_{f,max}$ when $\min(n,p)$ is large.

Table 6.2 Summary of $\alpha_{f,max}$ and $f(\alpha_{f,max})$ found from the equation (6.7)

$\min(n,p)$	$\alpha_{f,max}$	$f(\alpha_{f,max})$
16	0.2407	0.35
32	0.6412	1.13
64	0.8042	2.29
128	0.9810	4.33
256	0.9393	7.98
512	0.9663	14.59
1024	0.9814	26.67
2048	0.9989	48.91
4096	0.9945	90.08
8192	0.9970	166.72

Substituting (6.7) into Theorem 6.1, we can obtain the following corollary.

Corollary 6.1 *Under the forgetting learning with*

$$\alpha_f \approx \sqrt{1 - \frac{2e\log\min(n,p)}{\min(n,p)}}$$

and large $\min(n,p)$*, if the BAM has been trained with t library pairs and k is less than*

$$\frac{\min(n,p)}{2e\log\min(n,p)}, \tag{6.8}$$

then the probability that the $(t-k)$-th library pair (X_{t-k}, Y_{t-k}) is stored as a fixed point tends to one, as $n \to \infty$ and $p \to \infty$.

Proof of Corollary 6.1

Substituting (6.7) into Theorem 6.1,

$$f(\alpha_{f,max}) = \frac{\log e}{-\log\left(1 - \frac{2e\log\min(n,p)}{\min(n,p)}\right)}.$$

As $\log(1-x) \approx -x$ for small positive x,

$$f(\alpha_{f,max}) \approx \frac{\min(n,p)}{2e\log\min(n,p)}.$$

Hence, the proof is complete. □

From Corollary 6.1, the storage ability of the forgetting learning is similar to that of Kosko's encoding method. However, Kosko's encoding method can only encode up to $\frac{\min(n,p)}{2\log\min(n,p)}$ library pairs. Further encoding the new library pairs will damage the whole system.

On the other hand, the forgetting learning can encode any number of library pairs and always keeps the $\frac{\min(n,p)}{2e\log\min(n,p)}$ most recent library pairs. Hence, the BAM with the forgetting learning is similar to an adaptive memory, which always keeps recent information from the environment.

6.3 Computer Simulations

We have carried out a computer simulations to verify (6.7) and Corollary 6.1. The dimension is 512. We generate random library pairs and then encode them by the forgetting learning. Figure 6.2 shows the percentage of the last k-th previous library pair being stored as fixed point. In general, the case of $\alpha_f = 0.9663$ is better than other cases. Also, when $\alpha_f = 0.9663$, there is a sharply decreasing change for $k > 14$. Although the case of $\alpha_f = 0.975$ is better than that of $\alpha_f = 0.9663$ when $k > 18$, the corresponding percentage at $k = 18$ is only about 60 %. This simulation result agrees with our theoretical results presented above (see Table 6.1 and Table 6.2).

6.4 Chapter Summary

We have examined the statistical storage behavior of BAM under the forgetting learning. Also, we have derived a formula for choosing the forgetting constant such that the number of most recent library pairs being stored as fixed points is nearly maximal. This is,

$$\alpha_f \approx \sqrt{1 - \frac{2e\log\min(n,p)}{\min(n,p)}}.$$

With this value, the forgetting learning can encode any number of library pairs and always keeps the $\frac{\min(n,p)}{2e \log \min(n,p)}$ most recent library pairs in the model. Computer simulations have been carried out to verify our the theoretical results.

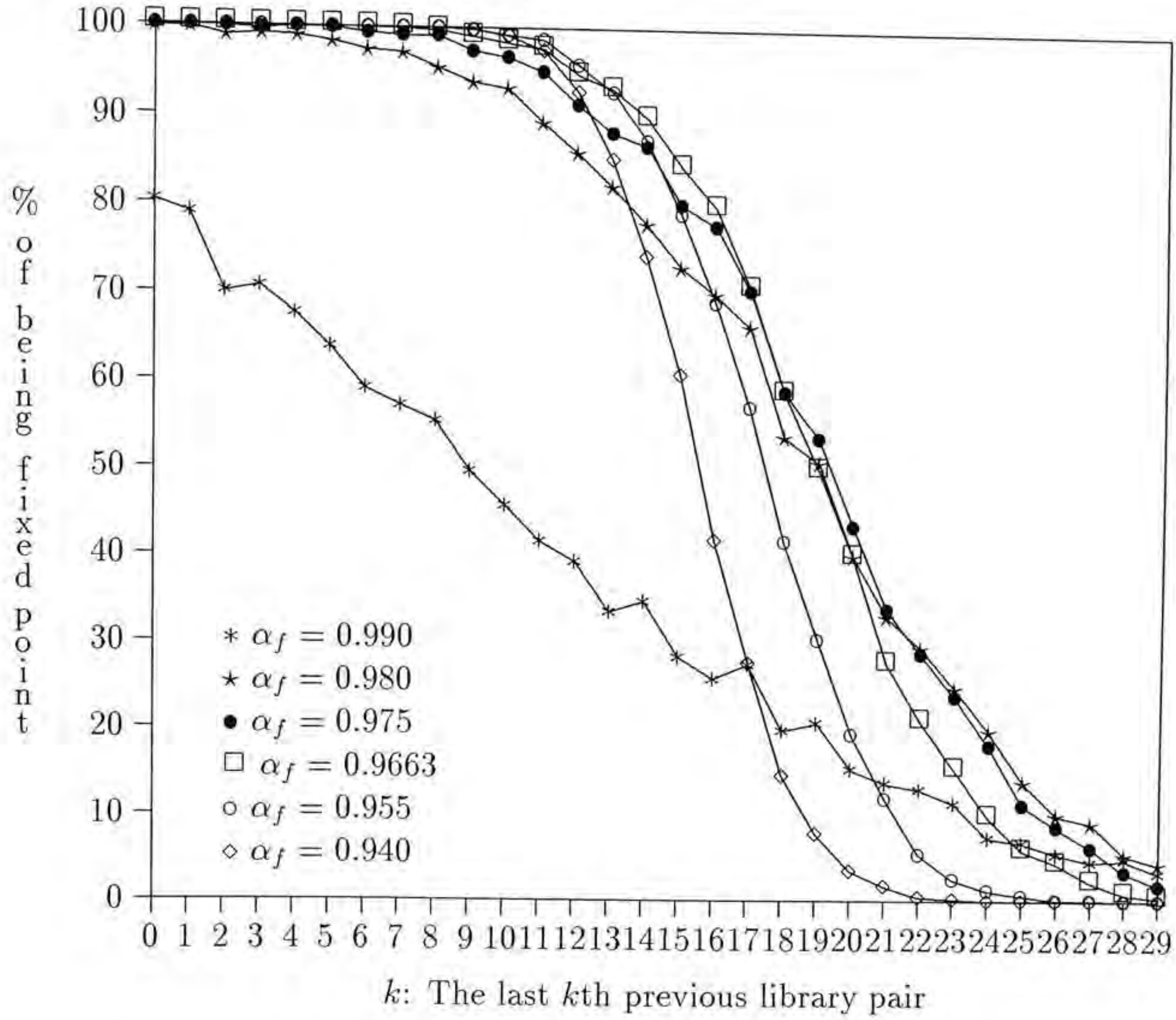


Figure 6.2 The percentage of the last k-th previous library pairs being fixed points where $n = p = 512$, $\alpha_f = \alpha_{f,max} = 0.9663$, and $\alpha_f = 0.940, 0.9550, 0.975, 0.980, 0.990$.

Part II

Kohonen Map: Applications in Data compression and Communications

Chapter 7

Introduction to Vector Quantization and Kohonen Map

This chapter reviews the basic concept of vector quantization and Kohonen map. To illustrate the ordering property of Kohonen map, a computer simulation is carried out.

7.1 Background on Vector quantization

As mentioned by Kohonen [1], vector quantization may be regarded as a special case of associative mapping in which the input patterns are directly mapped on a finite set of representing vectors (codevectors). The use of vector quantization for reducing the transmission bit rate or the storage has been recently investigated extensively [15, 16]. There are several classes of vector quantization: memoryless vector quantizer, feedback type vector quantizer, and trellis type vector quantizer [1]. All these classes are developed from the memoryless vector quantizer.

A memoryless vector quantizer Q can be defined as a mapping from k-dimensional

[1] In this thesis, we are mainly concerned with the memoryless and trellis type vector quantizers. The concept of the feedback vector quantizer will not be involved in this theses so we will not mention it here. The trellis type vector quantizers will be discussed in the next chapter.

Euclidean space $\Re^k$ to a finite subset Y of $\Re^k$:

$$Q : \Re^k \longmapsto \hat{Y}, \tag{7.1}$$

where $\hat{Y} = \{\vec{c}_1, \vec{c}_2, \cdots, \vec{c}_M\}$ is the set of codevectors (codebook) and M is the number of vectors in $\hat{Y}$. The quantizer Q is actually the composite of two separate functions, coder and decoder. The coder C is a mapping from $\Re^k$ to the set J of symbols

$$J = \{s_1, s_2, s_3, \cdots, s_M\}, \tag{7.2}$$

and the decoder D is a mapping from J to the codebook $\hat{Y}$

$$C : \Re^k \to J \text{ and } D : J \to \Re^k. \tag{7.3}$$

The quantity $R = \log_2 M$ is the code rate per input vector and $R_d = \frac{R}{k}$ is the code rate per dimension.

The goal of such a quantizer is to produce the 'best' possible codevector sequence for a given rate R. Hence, we require the idea of a distortion measure which is used to define the performance of the quantizer.

A distortion measure d is an assignment of a cost $d(\vec{x}, \vec{y})$ of reproducing an input vector $\vec{x}$ as a codevector $\vec{y} \in \hat{Y}$. Given such a distortion measure, we can quantify the performance of a quantizer by the average distortion $E[d(\vec{x}, \vec{y})]$. A quantizer will be good if the average distortion is small. In practice, the average is the long terms sample average

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} d(\vec{x}_i, \vec{y}_i). \tag{7.4}$$

Here we do not consider the difficult issues of selecting a distortion measure. For simplicity and ease of exposition, we focus on the squared error distortion measure:

$$d(\vec{x}_i, \vec{y}_i) = \|\vec{x}_i - \vec{y}_i\|^2 = \sum_{j=1}^{k} (x_{ij} - y_{ij})^2. \tag{7.5}$$

This is the simplest distortion measure and is commonly used in the vector quantization. For the squared-error distortion it is common practice to measure the performance by signal-to-quantization-noise ratio (SQNR)

$$SQNR = 10 \log_{10} \frac{E[\|x\|^2}{E[d(\vec{x}, \vec{y})]}. \tag{7.6}$$

This corresponds to normalizing the average distortion by the average energy and plotting it on a logarithmic scale: Large (small) SQNR corresponds to small (large) average distortion.

With a distortion measure and a set of codevectors, the quantizer usually operates in the following manner

$$\vec{y} = \vec{c}_{i^*} \tag{7.7}$$

where $\vec{c}_{i^*}$ is the codevector whose distortion to the input $\vec{x}$ is the smallest and $\vec{y}$ is the output vector.

Given a distortion measure, the performance of the quantizer (average distortion) is depended on the codevectors. There are a number of techniques for designing codebooks [16], such as Kohonen map [1, 7] and LBG algorithm [19]. The united goal of these techniques is to obtain a codebook, such that the vector quantizer minimizes the average distortion. [2] In the following two sections, we will give a review on these two techniques.

7.2 Introduction to LBG algorithm

Given a set of training input vectors, the LBG algorithm iteratively modifies the codebook to reduce the average distortion.

1. Given: the training input vectors $\{\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_n\}$ and an initial codebook.
2. Quantize each training input vector $\vec{x}_i$ into a codevector $\vec{y}_i \in \hat{Y}$.
3. Replace the old codevector by the centroid of all training input vectors which mapped into the same codevector in Step 2. If the new codevectors is the same as the old one or the average distortion is small enough, then quit.

Means of generating initial codevectors will be discussed later. Each step of the algorithm must either reduce the average distortion or leave it unchanged. The algorithm

[2] Note that in the sense of distortion minimization it is difficult to determine which algorithm is the best. We will use a simulation to illustrate this in Section 8.2

usually stops when the average distortion falls below some small threshold or the codebook does not change much. It should be emphasized that such iterative algorithm may not yield the optimum codebook with which the average distortion is the minimum. In general, the algorithm will yield a sub-optimum codebook. It is often useful, therefore, to enhance the algorithm by providing it with good initial codebooks and perhaps by trying it on several different initial codebooks. Note that the performance of other algorithms [16] also depends on the initial codebook.

In general, there are two approaches for the construction of the initial codebook.

- **"Random" Codes**: The first approach is that choosing randomly M vectors from the training input vectors as the initial codevectors [16]. Note that we cannot generate M random vectors as the initial codebook otherwise some codevectors will never be used in Step 2.

- **Splitting**: One can start with a small codebook and recursively construct larger ones. A sequence of bigger codebooks is constructed. We first find the centroid of the entire training input vectors as the first stage codebook with size one. This codevector is then split to form two codevectors (the initial codebook for the second stage) by adding small random numbers into the old codevector. Then, we apply the LBG algorithm to modify the codebook. The design continues in this way: the final codebook of one stage is split to form an initial codebook for the next stage.

7.3 Introduction to Kohonen Map

Apart from producing a codebook, Kohonen map has a nice property: ordering preserve [1]. That is, when two codevectors are neighbors to each other, their Euclidean distance is usually small. Based on this ordering property, Kohonen map can produce phoneme strings for word recognition in speech recognitions [7]. Also, it is a good way to reduce the dimensionality of the input in pattern recognition [11]. To achieve the ordering preserve, a neighborhood structure is introduced among the codevectors before learning.

Definition 7.1 *Given a set of codevectors $\vec{c}_i$, $i = 1, \cdots, M$, we introduce an $M \times M$ Boolean symmetric matrix $\underline{N}$ to define the neighborhood structure among the codevectors (i.e. the level-1 neighbor). n_{ij} denotes the i-th row and j-th column element of $\underline{N}$. If $\vec{c}_i$ is a level-1 neighbor of $\vec{c}_j$, then $n_{ij} = 1$. Otherwise, $n_{ij} = 0$. $\underline{N}$ is called the level-1 neighborhood matrix. By default, $n_{ii} = 1$. The collection of level-1 neighbors of $\vec{c}_i$ is denoted as $N_i(1)$. We use the word 'neighbor' to refer the level-1 neighbor in this thesis.*

Definition 7.2 *Given a $p \times q$ Boolean matrix $\underline{A}$ and a $q \times r$ Boolean matrix $\underline{B}$, the Boolean outer-product of two Boolean matrices $\underline{C} = \underline{A} \otimes \underline{B}$ is defined as:*

$$c_{ij} = (a_{i1} \wedge b_{1j}) \vee (a_{i1} \wedge b_{2j}) \vee \cdots\cdots (a_{iq} \wedge b_{qj}) \; \forall i = 1, \cdots, p \text{ and } \forall j = 1, \cdots, r,$$

where $\wedge$ and $\vee$ are the Boolean 'and' operator and the Boolean 'or' operator respectively. The notation $\underline{N}^v$ is defined as:

$$\underline{N}^v = \underbrace{\underline{N} \bigotimes \underline{N} \bigotimes \cdots\cdots \bigotimes \underline{N}}_{v},$$

where v is a non-negative integer and $\underline{N}^0$ is an identity matrix. Note that $\underline{N}^v$ is symmetry.

Definition 7.3 *Based on the definition of level-1 neighbor in Definition 7.1, we can define level-v neighbors of a codevector. A codevector $\vec{c}_i$ is a level-v neighbor of $\vec{c}_j$ if the ith row and jth column element of the matrix $\underline{N}^v$ is 1, where v is a non-negative integer. The collection of level-v neighbors of $\vec{c}_i$ is denoted as $N_i(v)$. Note: $N_i(v) \subseteq N_i(v+1)$. Similarly, we can define the neighborhood distance between two codevectors. The neighborhood distance between two codevectors, $\vec{c}_i$ and $\vec{c}_j$, is the smallest integer $\hat{d}_{ij}$ such that $\vec{c}_j \in N_i(\hat{d}_{ij})$.*

Definition 7.4 *The order of $\underline{N}$, v', is the smallest integer such that $\underline{N}^{v'+1} = \underline{N}^{v'}$.*

The neighborhood structure of Kohonen map can be arbitrarily defined. However, we usually use a regular neighborhood structure. Some regular structures are shown in Figure 7.1. Each vertex represents a codevector. If two codevectors are neighbors to each other, we use an edge to join their corresponding vertices together. **Note that the figure is only a graph representation and it does not reflect any actual geometric information of the codevectors.** Figure 7.1(a) is a 2-D grid structure, which can be applied to the reduction of the dimensionality. Figure 7.1(b) is a 1-D circle structure, which can be applied to the traveling salesman problem [78]. Figure 7.1(c) is a hypercube structure (hypercube graph), which can directly map the input vectors into binary words.

With the above definitions of the neighborhood structure, the learning rule of Kohonen map can be defined in the following way. Let $\vec{x}(t)$ be the t-th presentation training vector. We calculate the distortion between $\vec{x}(t)$ and each codevector.

$$d_i = \|\vec{x}(t) - \vec{c}_i(t)\|^2, \tag{7.8}$$

for $i = 1, \cdots, M$. Then we can find out a codevector $\vec{c}_{i^*}$ which is closest to $\vec{x}(t)$ such that $d_{i*} \leq d_i \; \forall\, i \neq i^*$. The updating rule is:

$$\begin{aligned} \vec{c}_i(t+1) &= \vec{c}_i(t) + \alpha_t(\vec{x}(t) - \vec{c}_i(t)) \\ & \qquad \forall \vec{c}_i \in N_{i^*}(v_t) \\ \vec{c}_i(t+1) &= \vec{c}_i(t) \\ & \qquad \forall \vec{c}_i \notin N_{i^*}(v_t) \end{aligned} \tag{7.9}$$

where v_t monotonically decreases from $\frac{v'}{2}$ to zero during the learning progress, where v' is the order of $\underline{N}$ (see Definition 7.4). Also, α_t, where $0 < \alpha_t < 1$, is a scalar parameter which decreases to zero monotonically. One of features of Kohonen map is its ordering preserve [7]. That is, the neighborhood distance between two codevectors (see Definition 7.2) is small, their Euclidean distance is usually small.

Figure 7.2 represents a computer simulation with two dimensional input and the codevectors. The probability density function of training input vectors is a uniform distribution over the $[-0.5, 0.5]^2$. The neighborhood structure Kohonen map is a 2-D 4-by-4 structure. To indicate which codevectors are neighbors, the points are further connected by lines. The initial codevectors of this example is a set of small random vectors shown in Figure 7.2(a).[3] The trained Kohonen map is shown in Figure 7.2(b). From the figure, the trained Kohonen map preserves the ordering in some geometric senses. Also, if the neighborhood distance between two codevectors is small, their Euclidean distance is usually small. For one dimensional case, the ordering property has been theoretically verified by several researchers [8]-[10]. Since

[3]For a better approach, the initial codevectors can be the sampling points of a small 2-D rectangular grid. For the 1-D circle structure, the initial codevectors can be the sampling points of a small circle. In general, the structure of the initial codebook should be the same as the neighborhood structure. It will usually get a better result.

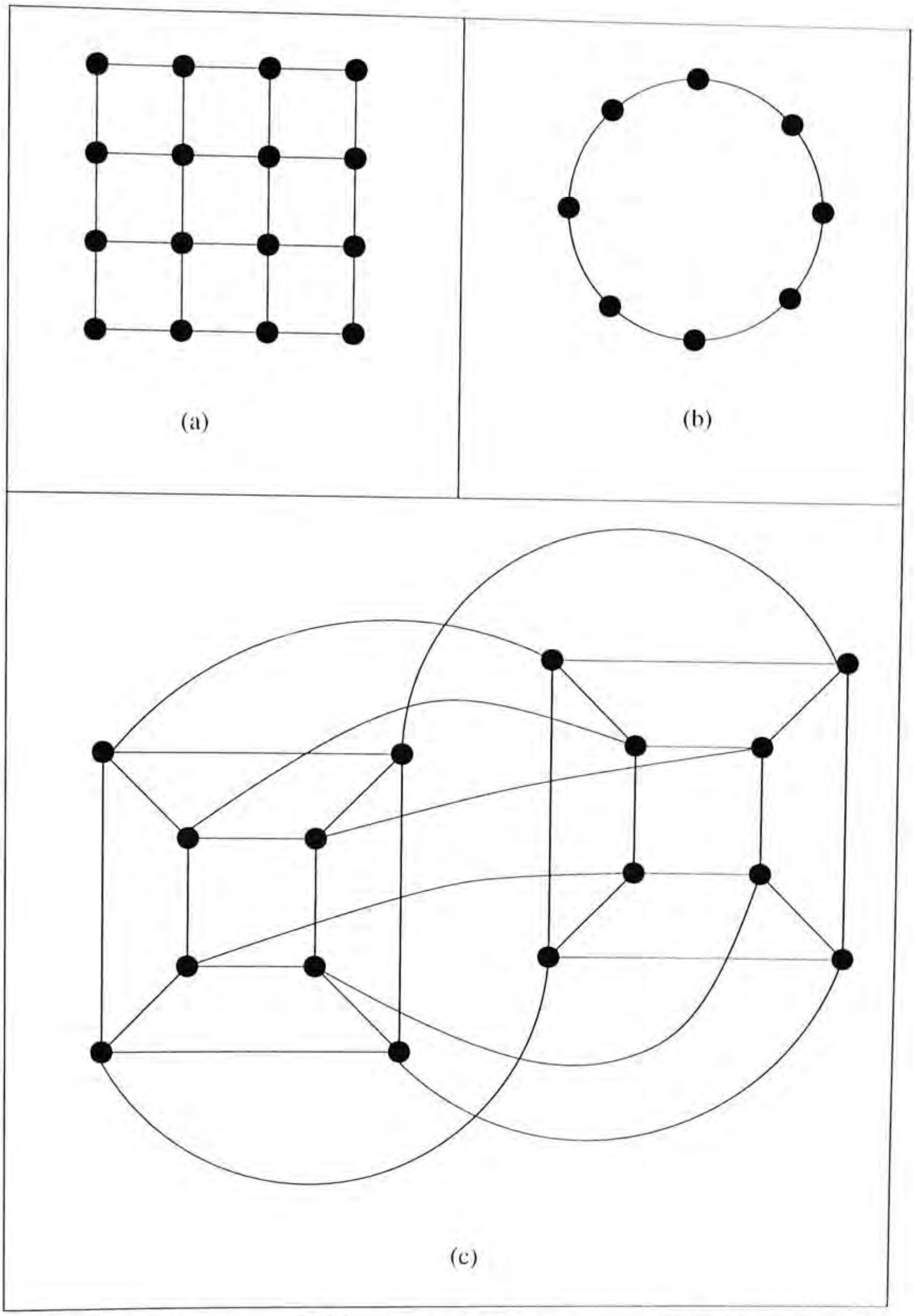


Figure 7.1 Three neighborhood structures of Kohonen map.
(a) 2-D 4-by-4 structure (b) 1-D circle structure
(c) 4-D hypercube structure

the higher dimensional Euclidean space $\Re^k$ is not an ordering space, it is difficult to well define the concept of the ordering in $\Re^k$. Hence, the ordering property in $\Re^k$ cannot be easily investigated mathematically.

Remark: In the field of communications, there is also a similar neighborhood structure among the channel waveforms based on the concept of Delaunay neighborhood [82, 83]. For example, the structure of Figure 7.1(a) is the same as the structure of a 16-QAM (see Figure 8.10(b)) and the structure of Figure 7.1(b) is the same as the structure a 8-QPSK (see Figure 8.11(b)) [77]. Hence, we believe that Kohonen map has some potential applications in the field of communications [29]–[31]. We will present the details of [29]–[31] in the next chapter.

7.4 Chapter Summary

In this chapter, we have described the basic concept of the vector quantization, LBG algorithm, and Kohonen map. We have also carried out a simple computer simulation to illustrate the ordering property of Kohonen map. Finally, we have pointed out that there is a similarity between the neighborhood structure of Kohonen map and the neighborhood of the channel waveforms.

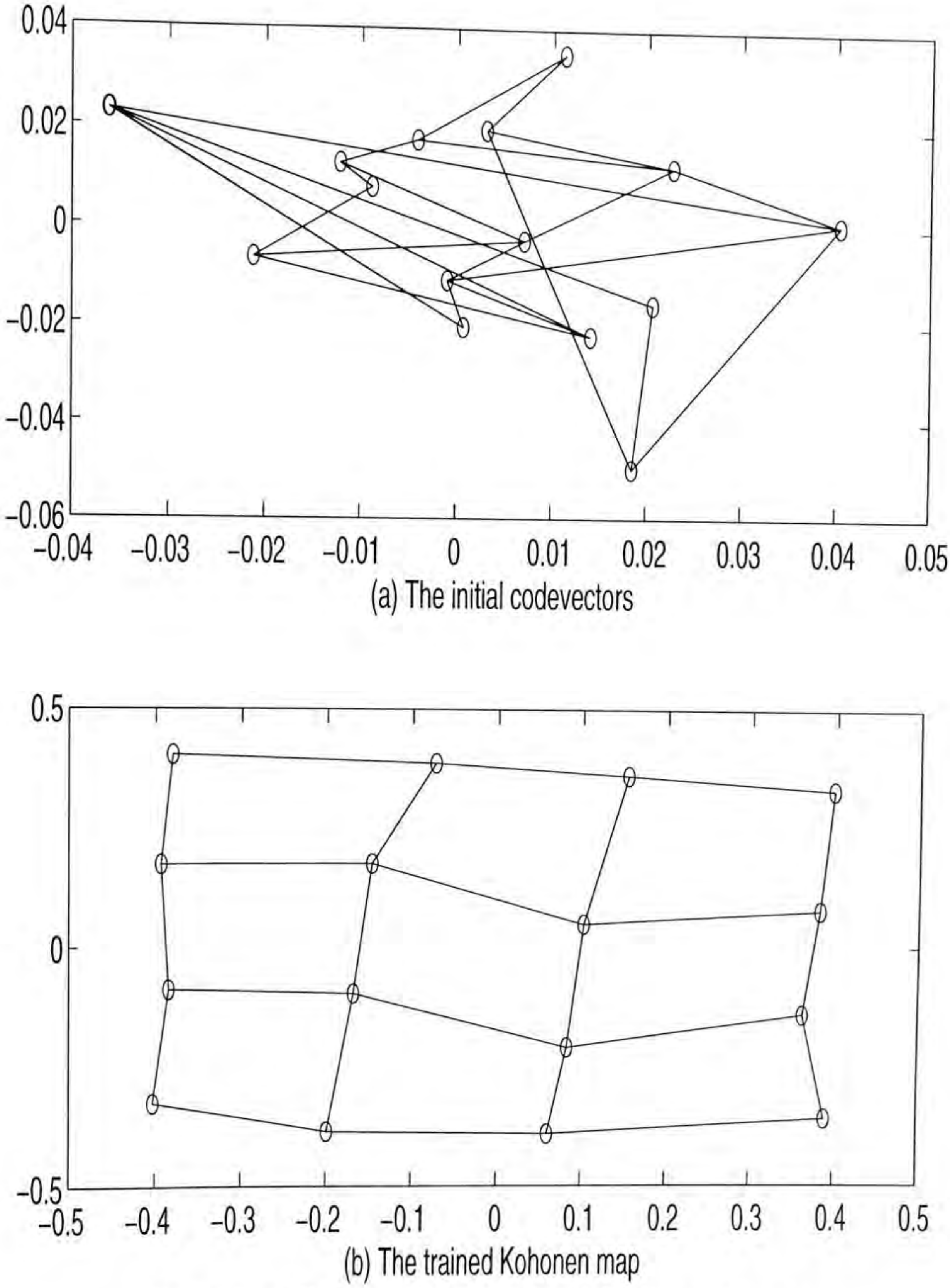


Figure 7.2 Example of a trained Kohonen map and the corresponding initial codevectors

Chapter 8

Applications of Kohonen Map in Data Compression and Communications

In this chapter, we will propose three new cross-relative applications of Kohonen map based on its ordering property. The united goal is to design robust transmission systems for vector-quantized data under noisy channel.

Firstly, we will use the neighborhood structure of Kohonen map to design the trellis type vector quantizer. The design process of our approach is simpler than that of the conventional trellis type vector quantizer.

Secondly, we discuss the way to efficiently transmit the vector-quantized data under noisy channel by considering the association between the codevectors and the channel waveforms. The association is on the basis of the neighborhood structure of Kohonen map and the neighborhood structure of the channel waveforms. Under our proposed approach, the impulsive noise in the received vector-quantized data is greatly reduced.

Lastly, we introduce an error control scheme for the transmission of vector-quantized data based on the concepts of the above first and second applications. Computer simulations show that the impulsive noise in the received data can be further reduced when noisy level in the channel belongs to a suitable range.

8.1 Use Kohonen Map to design Trellis Coded Vector Quantizer

In this section, we will simplify the design of a high performance trellis vector quantizer (TCVQ) [81] based on the neighborhood structure of Kohonen map. This new implementation is called trellis coded Kohonen map (TCKM) [29]. In terms of distortion, the performance of TCVQ is much better than that of the non-trellis vector quantizers. However, its design process, which is based on the Euclidean distance of codevectors, has a certain amount of computational overhead and space overhead.

In the TCKM, we use the neighborhood structure of Kohonen map to design the trellis. As the neighborhood structure of Kohonen map is predefined, different TCKMs with the same neighborhood structure but different codebooks can share the same trellis. The design process of the trellis in the TCKM is simpler than that of the TCVQ. Hence, the TCKM is suitable for adaptive environment. From the computer simulations, the performance of TCKM is comparable to that of TCVQ.

8.1.1 Trellis Coded Vector Quantizer

The trellis coded quantizer (TCQ) [79], which is a scalar quantizer, is motivated by the trellis coded modulation (TCM)[80]. The performance of TCQ is much better than that of conventional source coding techniques and is very close to the theoretical rate-distortion bound [79]. The TCVQ [81], as proposed by Wang and Moayeri, is a vector version of TCQ and can handle the fractional code rate (The code rate per dimension is less than one.).

The TCVQ is different from the classical vector quantization in the following way. Let the dimension of the vector be k and the code rate per dimension be R bits. In the vector quantization, the size of codebook is 2^{kR}, and any source vector can be represented by any codevector from the codebook at a time instant. In the TCVQ, the size of the codebook is extended to $M = 2^{kR+1}$, and at a time instant only 2^{kR} codevectors can be used to represent the source vector.

The TCVQ is described by trellis shown in Figure 8.1 and Figure 8.2. Each state transition (branch) corresponds to a subset of codevectors D_i for the 8-state trellis (C_i for the 4-state trellis), where the subsets D_i's (C_i's) are disjoint. Means of the construction of the trellis will be discussed later. The quantization is done on the sequences of source vectors. Let the source vector sequence be $S = \{\vec{x}_1, \cdots, \vec{x}_n\}$. According to the structure of the trellis, the number of allowable codevector sequences is only $C_o 2^{nkR}$ instead of $2^{n(kR+1)}$, where C_o is the number of states in the trellis. Thus, the code rate is asymptotically equal to R per dimension as n is large. Let the distance between two vector sequences with length n, say S and $\hat{S}$, be

$$d(S, \hat{S}) = \sqrt{\sum_{i=1}^{n} \|\vec{x}_i - \hat{\vec{x}}_i\|^2} \,. \tag{8.1}$$

Given a source vector sequence S, the allowable codevector sequence that is closest to S can be found out by Viterbi algorithm. We can view the input vector sequence as an element of a sequence space (which is a extended space formed by a sequence of vectors), and then we view the allowable codevector sequences as a sequence codebook (or say reproduction sequences) in the sequence space.

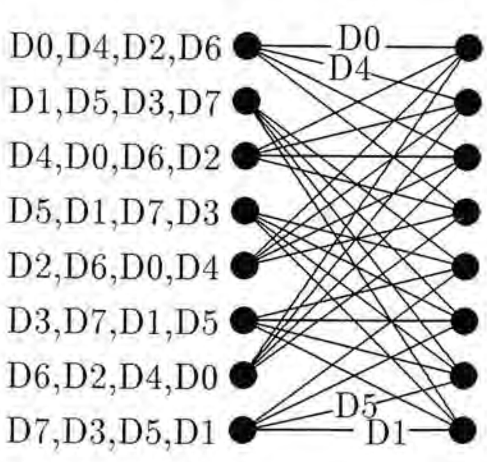


Figure 8.1 An 8-state trellis with four branches entering and leaving each node. Each branch associates with a subset of codevectors, D_i. All the subset D_i's are disjoint.

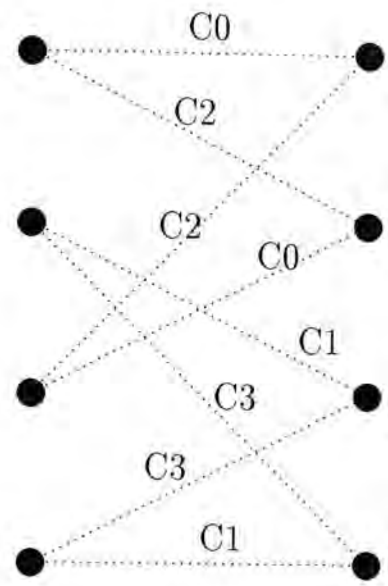


Figure 8.2 A 4-state trellis with two branches entering and leaving leaving each node. Each branch associates with a subset of codevectors, C_i. All the subset C_i's are disjoint.

Let us consider the following example.

- A TCVQ with a 4-state trellis (see Figure 8.3)
- The codevectors are $\vec{c}_1 = -3$, $\vec{c}_2 = -1$, $\vec{c}_3 = 1$, and $\vec{c}_4 = 3$.
- The subsets are $C_0 = \{\vec{c}_1\}$, $C_1 = \{\vec{c}_2\}$, $C_2 = \{\vec{c}_3\}$, and $C_3 = \{\vec{c}_4\}$.
- The input sequence is $S = \{0.5, 1.2, 1.4\}$.
- From Viterbi algorithm (see Figure 8.3), the output sequence is

$$S' = \{\vec{c}_2, \vec{c}_2, \vec{c}_3\} = \{1, 1, 3\}.$$

- As $n = 3$, the number of allowable codevector sequences is

$$C_o 2^{nkR} = 4 \times 2^3.$$

The code rate is $\frac{2+3}{3}$ bits per vector.

In the above example, the length of the input sequence is only 3 and a relative large code rate is obtained. In fact, if the length is very large, the code rate tends to 1 bit per vector even the size of the codebook is 4. In general, if the size of the codebook is M, then the code rate tends to $(\log_2 M - 1)$ bit per vector. Here, we only present the standard TCVQ in which only half of codebook is used at a time instant. In fact, the generalized version is that a fraction of the codebook is used at a time instant. The code rate of the generalized version tends to $(\log_2 M + \log_2 a)$, where a is the fraction.

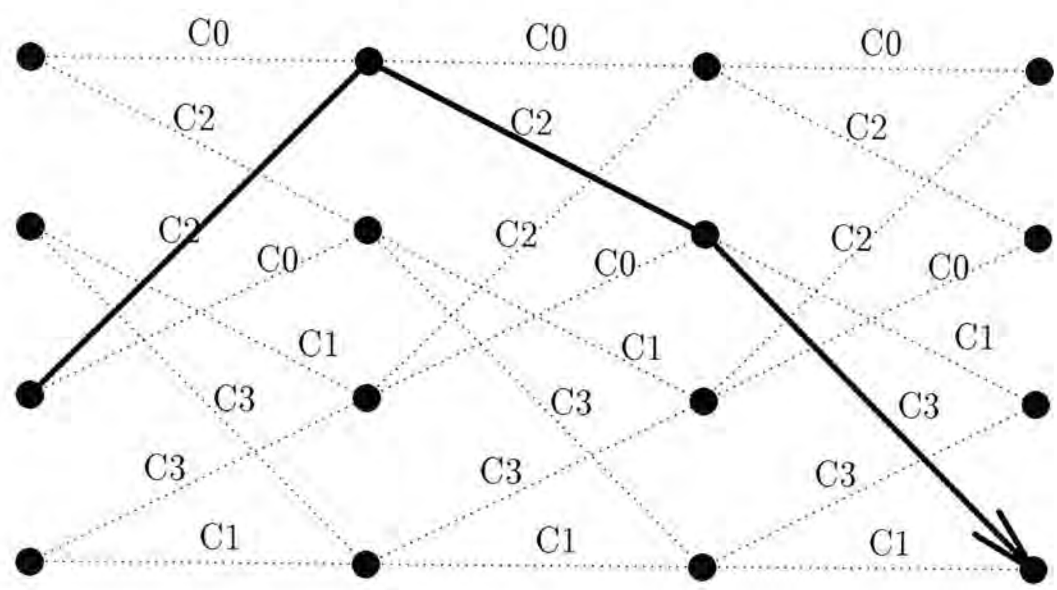


Figure 8.3 Realization of the encoding in the TCVQ. The closest output sequence is denoted by the solid line.

The design process of TCVQ consists of two phases. The first phase is the construction of the codebook. In the TCVQ, the LBG algorithm is used to find the codebook [19]. The second phase is the construction of the trellis in which a subset of codevectors is assigned to each transition edge.

To achieve a small distortion, the allowable codevector sequences should be well distributed over the sequence space. Otherwise, the allowable codevector sequences lie closely together and then only a portion of the sequence space is covered by the allowable codevector sequences. One of the methods to well distribute the allowable codevector sequence is to maximize the minimum distance of two allowable codevector sequences among the all pairs. Note that in the TCM we also want to achieve a similar task by the set partition of the channel waveforms. The set partition in the TCM is very simple since there exists a regular structure among the channel waveforms. As there does not exist a regular structure among the codevectors, Wang [81] proposes the following heuristics to do the set partition of the codevectors.

During partitioning, the distances between all possible pairs of codevectors are calculated and are sorted in ascending order. This generates a table with $\frac{M(M-1)}{2}$ entries. The i-th entry of the table corresponds to two codevectors, $\vec{c}_i$ and $\vec{c}'_i$. Initially, $\vec{c}_0$ and $\vec{c}'_0$ are placed in subsets B_0 and B_1, respectively. The first entry is then removed from the table. The following steps are repeated until the size of one of the subsets, B_0 or B_1, becomes $\frac{M}{2}$.

1. Search the table to find an index j such that $\forall i < j$, $\vec{c}_i$ or $\vec{c}'_i$ do not belong to $B_0 \cup B_1$, but at least one of $\vec{c}_j$ and $\vec{c}'_j$ belongs to $B_0 \cup B_1$.

2. If both $\vec{c}_j$ and $\vec{c}'_j$ belong to $B_0 \cup B_1$, the j-th entry is removed from the table. Goto step (1).

3. If $\vec{c}_j$ belongs to B_0 (or B_1), then add $\vec{c}'_j$ to B_1 (or B_0) and remove the j-th entry from the table. If $\vec{c}'_j$ belongs to B_0 (or B_1), then add $\vec{c}_j$ to B_1 (or B_0) and remove the j-th entry from the table.

4. If the size of B_0 (or B_1) becomes $\frac{M}{2}$, then add the remaining unassigned codevectors to B_1 (or B_0). Otherwise, go to step (1).

B_0 (and B_1) is further partitioned into C_0 and C_2 (and C_1 and C_3), and so on. The whole set of codevectors is partitioned in several stages such that a binary tree of subsets of codevectors, shown in Figure 8.4, is obtained. The number of levels of the tree depends on the structure of the trellis. For example, when the trellis shown in Figure 8.1 is used, the codevectors should be partitioned into eight subsets. In the case of Figure 8.2, the codevectors should be partitioned into four subsets.

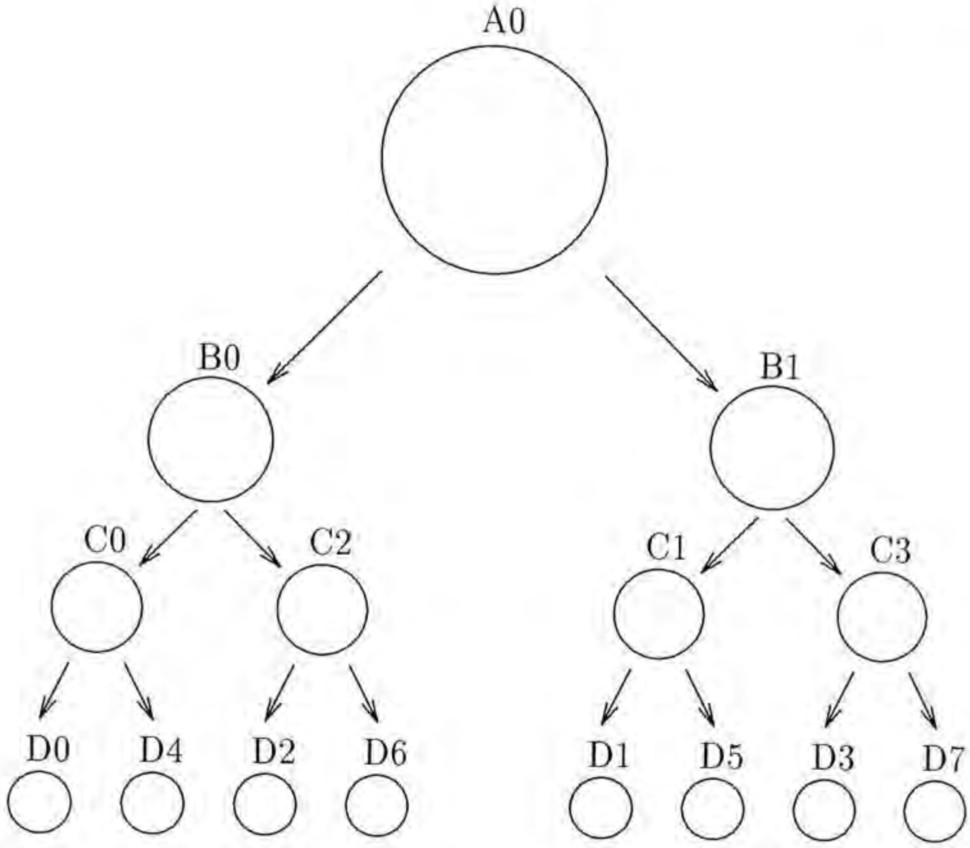


Figure 8.4 The basic idea of set Partition in the TCVQ.

After the partitioning, the assignment can be done based on the following rules.

1. A branch is assigned to the members of the same partition.

2. Adjacent branches are assigned to the members of the next larger partition.

3. All the codevectors are used equally often.

The detailed assignment rules can be found in [79, 80].

The goal of the set partition is to maximize the minimum distance within each subset. Readers are referred to [80] for better understanding why the set partition and the above assignment rules can increase the minimum distance between any pairs of allowable codevector sequences. Let us briefly explain it by using Figure 8.2.

- Firstly, we consider the allowable codevector sequences whose paths in the trellis are the same. For example, the path $\{C_0, C_2, C_1\}$ (see Figure 8.2) represents 8 allowable codevector sequences if the size of C_i is two. So, maximization of the minimum distance within each subset C_i will increase the minimum distance within these 8 sequences.

- Secondly, we consider the two paths in which only one subset is different. For example, the path $\{C_0, C_0, C_0\}$ contains 8 allowable codevector sequences and the path $\{C_0, C_0, C_2\}$ contains 8 allowable codevector sequences. The minimum distance within the subset $B_0 = C_0 \cup C_2$ determines the minimum distance within the above 16 sequences. So maximization of the minimum distance within each subset B_i (together with the Assignment Rule 2) will increase the minimum distances within the 16 sequences.

- Other cases can be explained in a similar manner.

The second phase above-mentioned needs a certain amount of computational and space overhead. For example, if $M = 256$, we must calculate 32640 Euclidean distances within the codevectors and then sort them in ascending order. Also, the size of the table is very large (32640). When we frequently change the codebook, we need to do set partition frequently. Hence, the TCVQ may not be suitable for adaptive environment. Many common signal sets in the TCM have already had their corresponding set partition and trellis. The signal sets of TCM in a communication system are usually very regular. Since the codevectors of TCVQ (or vector quantizer) are usually non-regular, we cannot apply the existing results of TCM to TCVQ. In the following, we will impose a virtual regular structure among the codevectors (based on the neighborhood structure of Kohonen map) to simplify its design process.

8.1.2 Trellis Coded Kohonen Map

According to the ordering property of Kohonen map, the neighborhood structure of Kohonen map reflects certain geometric information of the codevectors. Hence, we can partition the codebook based on the neighborhood structure when we use Kohonen map to construct the codebook. This new implementation is called the TCKM. In fact, the set partition of Kohonen map in the TCKM is not necessary. [1] It is because *the set partition of Kohonen map in the TCKM is the same as the set partition of*

[1]It means that in the TCKM we do not need to partition (the algorithm presented in Section 8.1.1) the codevectors based on the neighborhood distance.

the signal set in TCM [80], provided that the neighborhood structure of Kohonen map is the same as the neighborhood structure of the signal set in the TCM. For example, the neighborhood structure of a 2-D 4-by-4 Kohonen map is the same as the neighborhood structure of the channel waveforms of 16-QAM (see Figure 8.9 and Figure 8.10 (b)). Note that the neighborhood structure of the channel waveforms is created from the concept of the Delaunay neighborhood [82, 83].

Figure 8.5 illustrates the set partition of a 2-D 4-by-4 regular Kohonen map based on the set partition of a 16-QAM. We use a planar graph to represent Kohonen map. Each vertex represents a codevector. If two codevectors are level-1 neighbors to each other, we connect their corresponding vertices by an edge. Since the planar graph representation is used, Figure 8.5 does not reflect the actual geometric information of the codevectors. By the same principle, we have partitioned a 32-codevector Kohonen map shown in Figure 8.6.

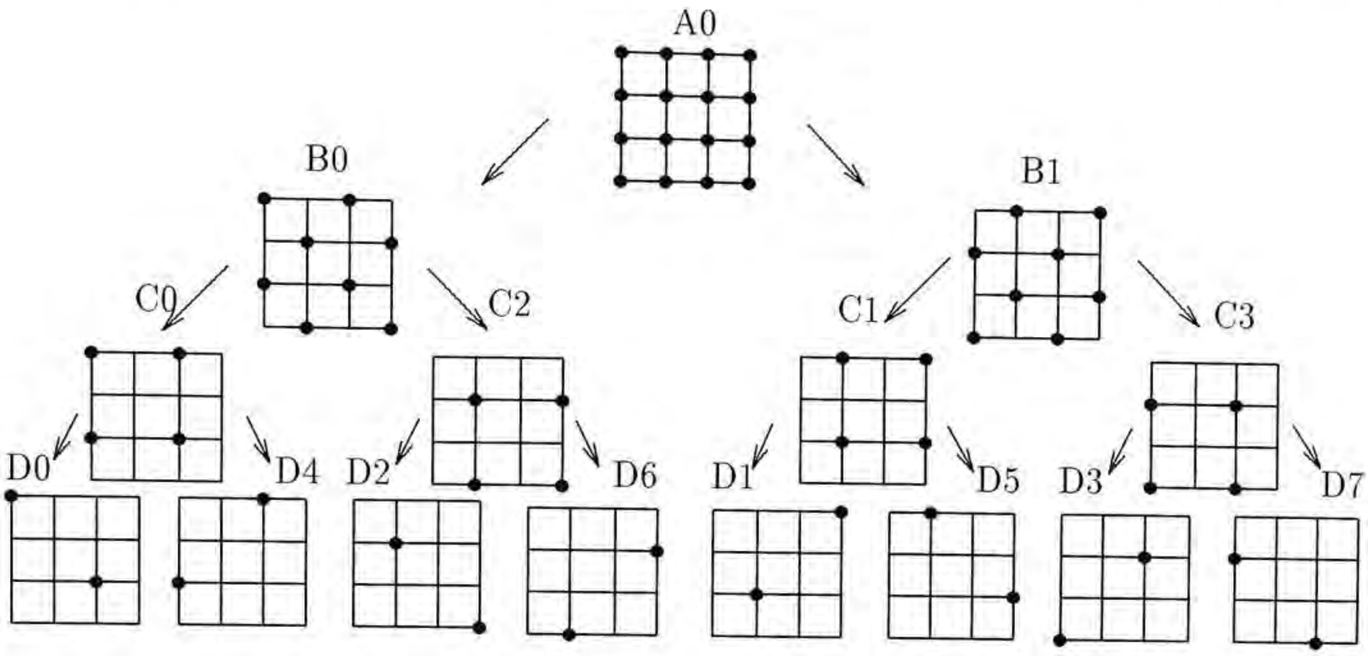


Figure 8.5 Set partition of a 4 by 4 Kohonen map based on the set partition of a 16-QAM signal constellation.

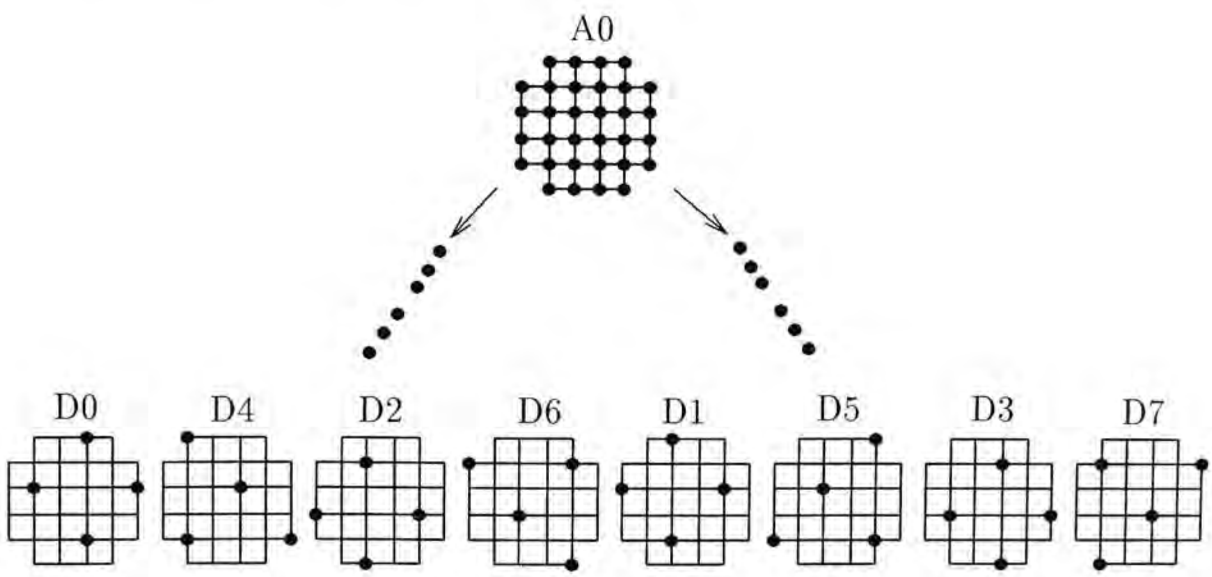


Figure 8.6 Set partition of a 32-codevector Kohonen map

The TCM has been well studied in the field of communications and many common signal sets in the TCM have already had their corresponding set partition and trellis. Hence, the design process of TCKM is very simple when we use the neighborhood structure of the channel waveforms in the TCM to define the neighborhood structure of Kohonen map. The basic operation of TCKM is the same as that of TCVQ. **The difference between them is the design process.**

- In the TCKM, Kohonen map's learning is used to construct the codebook. In the TCVQ, the LBG algorithm is used instead.
- In the TCKM, **the set partition of codebooks is based on the neighborhood structure of the codevectors**. In the TCVQ, it is based on the Euclidean distances among codevectors.

The feature of TCKM is that the trellis can be constructed before the training. Also, different codebooks can share the same trellis provided that their neighborhood structures (i.e. $\underline{N}$) are the same. Hence, the design process of TCKM is simpler than that of TCVQ. The simulation results, shown below, demonstrate that the performance of TCKM is comparable to that of TCVQ.

8.1.3 Computer Simulations

We designed several 2-dimensional LBG, TCVQ's, and TCKM's with different rates for a zero-mean, unit-variance, i.i.d. Gaussian 2-dimensional vector source. The number of training samples is 1024. For all quantizers, the number of learning cycles is 10.[2] In the TCVQ and TCKM, we test two trellis structures with different numbers of states, 4-state trellis and 8-state trellis. We have repeated the above experiment 5 times. Table 8.1 shows the SQNR for the code rate 1.5 bits per dimension. Table 8.2 shows the SQNR for the code rate 2 bits per dimension. Note that a quantizer with a greater value of SQNR has a better performance. These two tables show that the performance of TCKM is comparable to that of TCVQ and that both TCVQ and TCKM can realize an improvement in SQNR from 0.5 to 1 dB over the standard LBG method. Also, with a larger number of states in the trellis, both TCVQ and TCKM can achieve a better performance.

[2]When the number of learning cycles is set to 8 only, the LBG algorithm does not form a good codebook.

Table 8.1 SQNR (in dB) with code rate 1.5 bits per dimension for different quantizers.

Experiment No.	LBG	4-state TCVQ	8-state TCVQ	4-state TCKM	8-state TCKM	theoretical bound
1	6.80	7.45	7.55	7.77	7.83	9.03
2	6.89	7.64	7.79	7.38	7.66	9.03
3	6.98	7.63	7.89	7.56	7.90	9.03
4	7.05	7.51	7.66	7.41	7.80	9.03
5	6.85	7.43	7.55	7.43	7.69	9.03

Table 8.2 SQNR (in dB) with code rate 2 bits per dimension for different quantizers.

Experiment No.	LBG	4-state TCVQ	8-state TCVQ	4-state TCKM	8-state TCKM	theoretical bound
1	9.23	10.50	10.78	10.73	10.80	12.04
2	9.68	10.43	10.67	10.35	10.67	12.04
3	9.95	10.77	10.97	10.89	10.97	12.04
4	9.55	10.69	10.87	10.67	10.89	12.04
5	9.58	10.47	10.60	10.65	10.75	12.04

In another experiment, we test the TCKM with six common natural images: Lena, Baboon, Pepper, Clown, Fruit, and F16. The images are shown in Figure 8.7. We use the first three images to train four different quantizers: a TCKM with a 16-by-16 Kohonen map, a TCKM with Kohonen map which structure is the cartesian product of two 1-D 16-circles, a TCVQ using LBG with 256 codevectors, and a standard LBG with 128 codevectors. Each image is divided into a number of 2×4 blocks. Hence, the dimension of the vectors is 8 and the code rate is $\frac{7}{8}$ bit per dimension (per pixel). Although there are 256 codevectors in TCVQ and TCKM, the code rate of TCVQ and

TCKM is the same as that of the standard LBG with 128 codevectors. It is because both TCVQ and TCKM are sequence base coder and only 128 codevectors are used at a time instant.

The images are then quantized by these three quantizers and the reproduction images are obtained. Table 8.3 summarizes the root mean square error (RMSE) of the reproduction images to original images. Note that a quantizer with a smaller value of RMSE implies that it has a better performance. Again, the performance of the TCVQ and TCKM are similar. Except the image 'Baboon', the TCKM yields a great improvement in RMSE over the standard LBG and its performance is comparable to the TCVQ's case. For the image 'Baboon', the reason may be that the LBG is good for quantization of the training images.[3]

Table 8.3 The RMSE of the quantization images for different quantizers. The code rate is 7/8 bit per pixel.

Image	LBG	8-state TCVQ	8-state TCKM 16-by-16 grid	8-state TCKM product of two circles
Lena	7.4605	7.1754	6.9746	7.0323
Baboon	14.3412	13.7542	14.2805	14.6817
Pepper	7.3950	7.3030	6.9674	7.1141
Clown	13.4465	13.0670	12.3170	12.5561
Fruit	10.4821	10.3771	9.1556	9.1782
F16	9.6672	9.4909	8.8027	8.9248

[3]Table 8.5 (in the next section) summarizes the RMSE of the quantization images under conventional quantizers (i.e., all codevectors are used at a time instant). From Table 8.5, the standard LBG gets a better result for the training images (Lena, Baboon, and Pepper) but a poorer result for the remaining images.

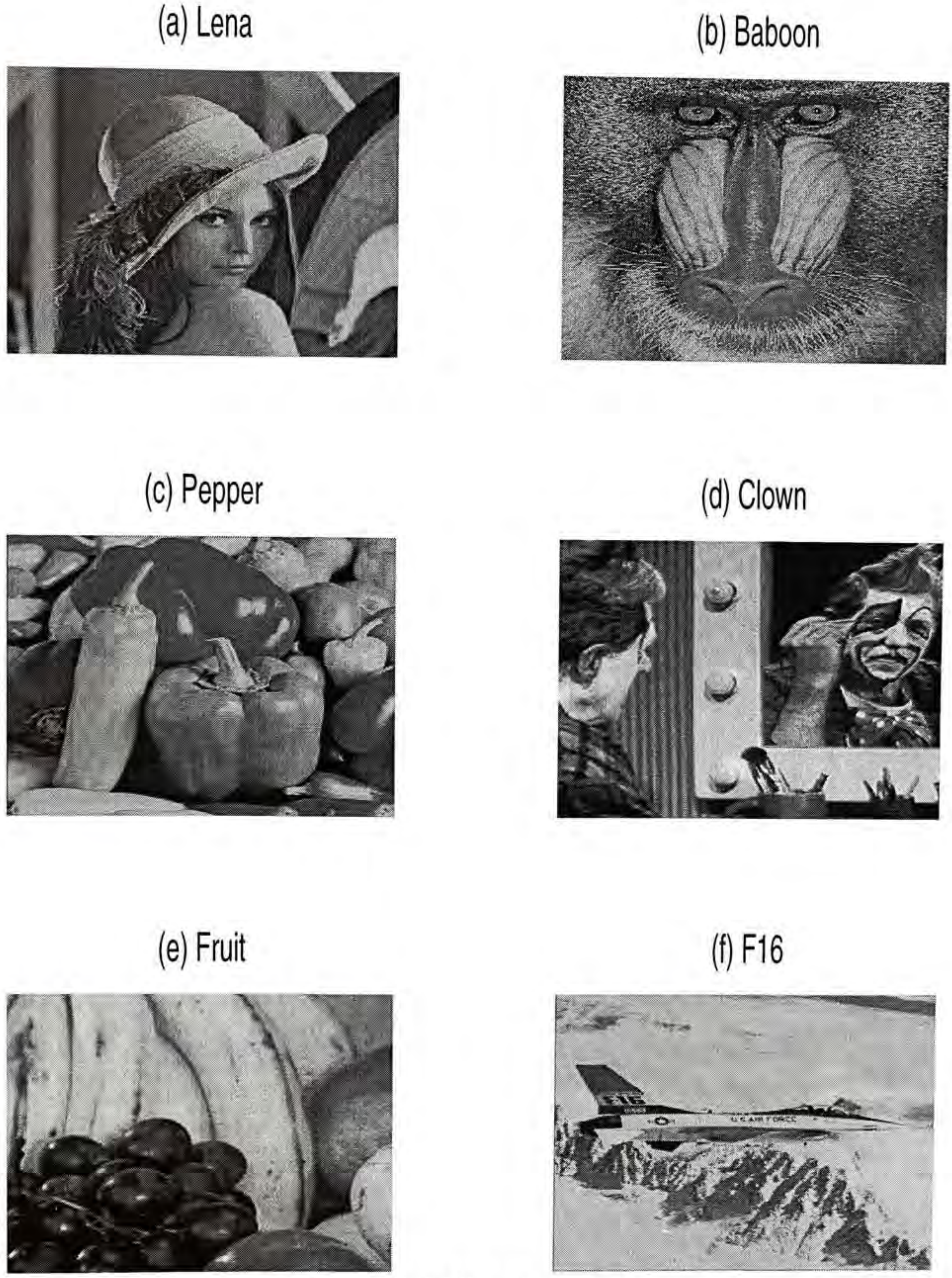


Figure 8.7 The six images.

8.2 Kohonen Map:Combined Vector Quantization and Modulation

When we transmit the vector-quantized images (or speech) under a noisy channel, the noise in the received data is impulsive. Moreover, the impulsive noise is difficultly removed by linear or non-linear filter since vector quantization is used (i.e., the size of the impulse is large). In this section, we present a methodology to reduce the impulsive noise in the received data based on the ordering property of Kohonen map. This can be achieved by considering the association between the neighborhood structure of Kohonen map and the neighborhood structure of the channel waveforms. Computer simulation shows that our approach can reduce the impulsive noise in the received data even if we do not use any error correction scheme in the transmission system or filter. Let us first explain why the impulsive noise is produced.

8.2.1 Impulsive Noise in the received data

When we transmit the quantization data, we transmit the symbols instead of the code-vectors. Hence, a digital transmission system is needed. A simple digital transmission system is shown in Figure 8.8. It consists of four blocks: source coder, modulator, channel, and symbol's detector (demodulator).

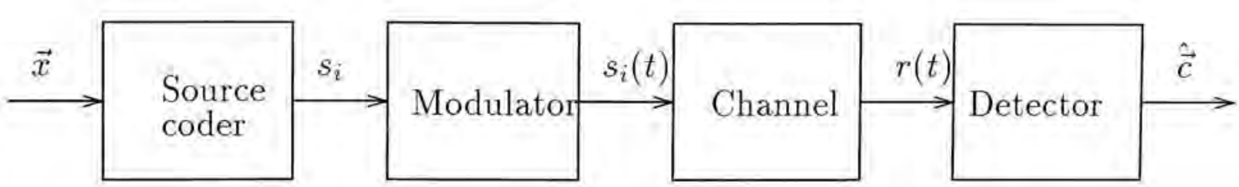


Figure 8.8 A simple digital transmission system

- Based on the input $\vec{x}$, the source coder outputs the corresponding symbol at constant rate r_t (symbols/per sec). The duration of each symbol is $T = \frac{1}{r_t}$. Here, we may treat the coder of the vector quantizer C as a source coder.

- The modulator interfaces the source coder to the channel. It takes in the source outputs and produces channel waveforms that suit the physical nature of the channel.

To consider carrier modulation, each symbol s_i is associated with a channel waveform $s_i(t)$:

$$s_i(t) = s_{i1}\sqrt{\frac{2}{T}}\cos\omega_c t + s_{i2}\sqrt{\frac{2}{T}}\sin\omega_c t \tag{8.2}$$

where ω_c is the carrier frequency. In the field of communication, we usually use a vector notation. That is, each channel waveform is represented by a vector $\vec{s}_i$ in the signal space:

$$\vec{s}_i = \begin{pmatrix} s_{i1} \\ s_{i2} \end{pmatrix} . \tag{8.3}$$

Three common modulation methods are shown in Figure 8.9. Note that we can use the concept of Delaunay neighborhood [82, 83] to introduce a neighborhood structure among the channel waveforms $s_i(t)$'s ($\vec{s}_i$'s).

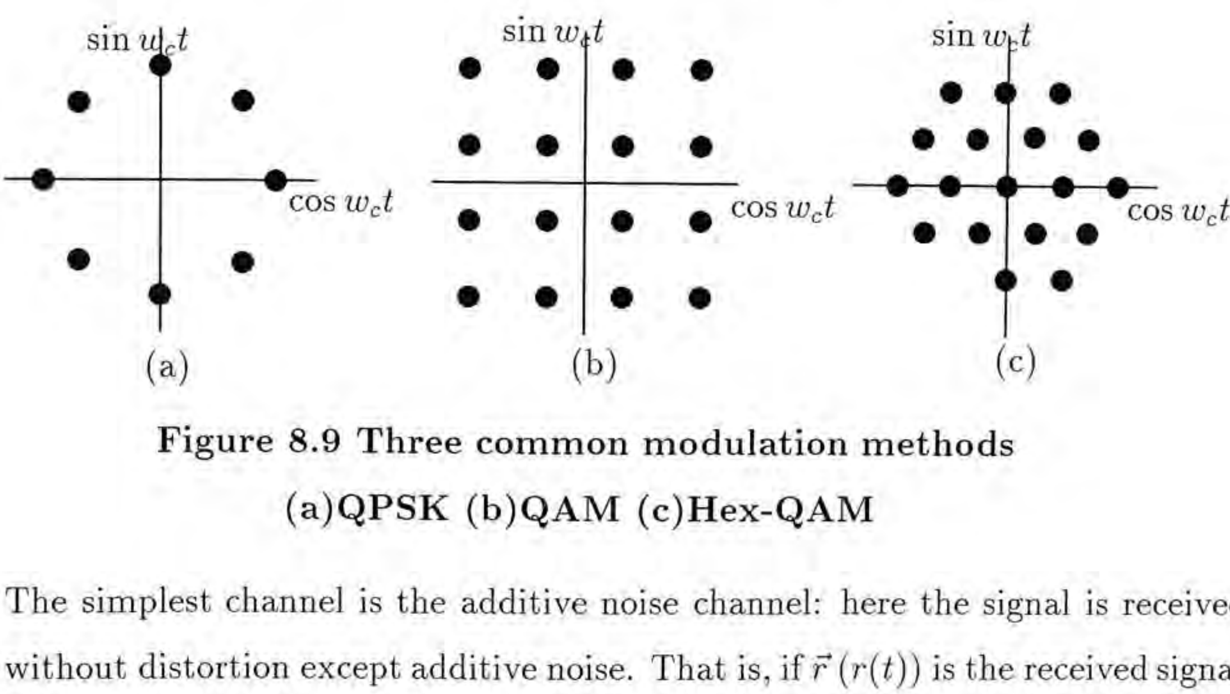


Figure 8.9 Three common modulation methods
(a)QPSK (b)QAM (c)Hex-QAM

- The simplest channel is the additive noise channel: here the signal is received without distortion except additive noise. That is, if $\vec{r}$ ($r(t)$) is the received signal (assuming that $\vec{s}_i$ is transmitted), then

$$\vec{r} = \vec{s}_i + \vec{n}$$

where $\vec{n}$ is the additive noise. Usually, Gaussian noise is assumed.

- Given the received signal $\vec{r}$, the detector estimates the transmitted signal $\hat{\vec{s}}$ based on the properties of the channel. If the above simplest channel is assumed, the best detector is

$$\hat{\vec{s}} = \vec{s}_i \text{ if } d(\vec{s}_i, \vec{r}) \leq d(\vec{s}_j, \vec{r}) \ \forall\, j \neq i \ . \tag{8.4}$$

where $d(\cdot,\cdot)$ is the Euclidean distance between two vectors. Note that if the estimated signal $\vec{s}$ is not equal to the transmitted signal, a symbol error occurs. When an error occurs, the estimated signal usually is a Delaunay neighbor of the transmitted signal (or a signal near the transmitted signal).

If the channel is noisy, the estimation may be incorrect. Given the transmitted codevector $\vec{c}_t$ in original data space (the transmitted waveform is $\vec{s}_t$), the estimated signal $\hat{\vec{s}}$ is usually close to the transmitted signal but the estimated codevector $\hat{\vec{c}}$ may not be close to $\vec{c}_t$. The distance between the transmitted codevector and the estimated codevector depends on the association between $\vec{c}_i$'s and $\vec{s}_i$'s. Under a noise-free channel, we do not need to care the association between the codevectors and the channel waveforms. But for a noisy channel, it is a serious problem.

For image or speech data, we can accept certain amount of smooth noise. Unfortunately, when we transmit the vector-quantized images (or speech) under a noisy channel, the noise in the received data is impulsive if we do not care the association. Besides, the size of the impulse is large. This large impulse cannot be easily removed by a linear or nonlinear filter. One may suggest that we can calculate distances within the codevectors and then do the association based on heuristics or simulated annealing [84, 85]. Note that the problem of the association is NP. However such approaches involve a great amount of computation and space overhead if the size of the codebook is large. Also, we need to know the symbol error probability of the channel in these approach. Hence, if a new codebook is used or the noise level changes, we need to do the association again.

In the following, we will present a methodology to achieve the association based on the neighborhood structure of Kohonen map [31]. [4]

[4]Recently, a similar work [86] has been reported during the review process of my article [31].

8.2.2 Combined Kohonen Map and Modulation

As mentioned in above, one of features of Kohonen map is the ordering preserve. That is, when two codevectors are neighbors to each other, their Euclidean distance is usually small. Based on this feature, we can do the association in the following way. **The neighborhood structure of the channel waveforms, which is created from the concept of Delaunay neighborhood, matches the neighborhood structure of Kohonen map.** That is, if the codevectors $\vec{c}_i$ and $\vec{c}_j$ are neighbors to each other, then the corresponding channel waveforms $\vec{s}_i$ and $\vec{s}_j$ should also be Delaunay neighbors to each other. With such approach, if the symbol detector wrongly estimates a transmitted waveform $\vec{s}_i$ to its neighborhood waveform $\vec{s}_j$, then the distortion from $\vec{c}_j$ to $\vec{c}_i$ is usually small. Hence, the error events in the receiver only cause a small increase in overall RMSE in the received vector-quantized data.

Given the modulation system, we create the neighborhood structure of the channel waveforms based on the Delaunay neighborhood. Then we use this neighborhood structure to train Kohonen map. It should be emphasized that the neighborhood structure of Kohonen map is not created from the concept of the Delaunay relationship in $\Re^k$.

Since the neighborhood structure of Kohonen map is predefined, different codebooks with the same neighborhood structure can share the same association. Under our approach, the design of modulation system and the design of the vector quantizer are considered as a whole together.

Figure 8.10 shows this idea. In Figure 8.10(a), there is a trained 2-D 4-by-4 Kohonen map in the original data space. The corresponding channel waveforms (16 QAM) in the signal space are shown in Figure 8.10(b). We use a line to indicate the Delaunay relationship between two channel waveforms. Note the neighborhood structure of Figure 8.10(a) matches that of Figure 8.11(b). Figure 8.11 shows another case. In Figure 8.11(a), there is a trained 1-D circle Kohonen map in the original data space. The corresponding channel waveforms (8 QPSK) are shown in Figure 8.11(b).

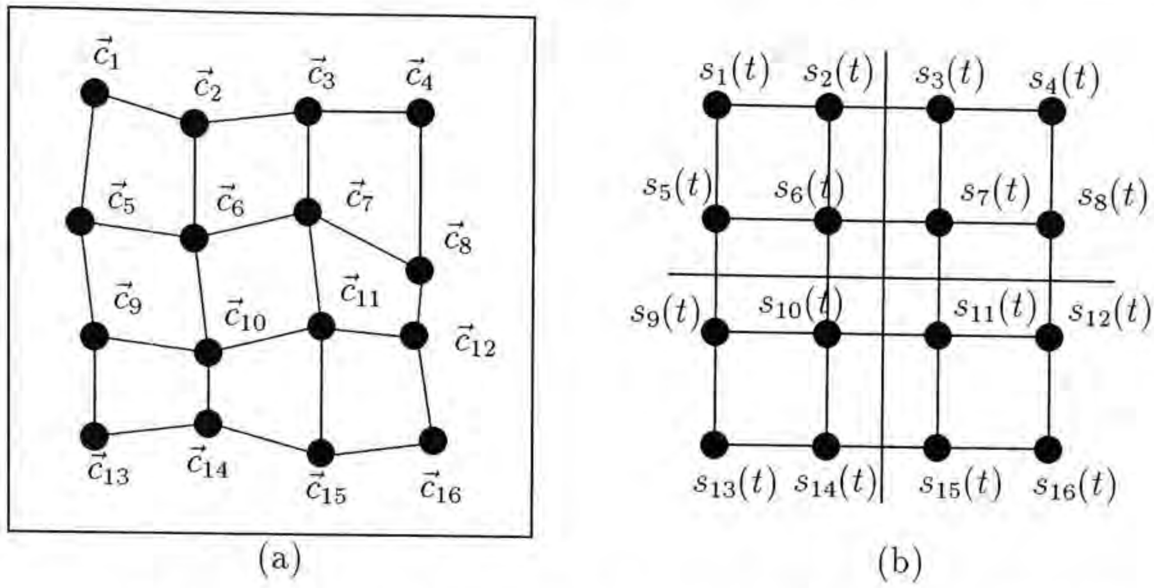


Figure 8.10 The idea of matching between codevectors and channel waveforms. (a)2-D regular 4-by-4 Kohonen map (b)The corresponding channel waveforms

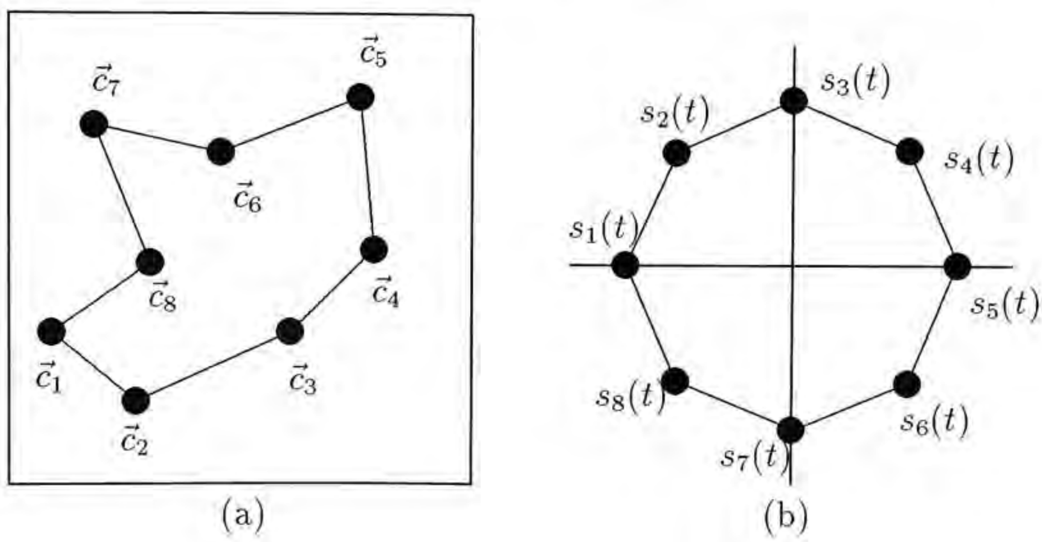


Figure 8.11 The idea of matching between codevectors and channel waveforms. (a)1-D circle Kohonen map (b)The corresponding channel waveforms

If the modulation scheme is M-QAM, we should use a 2-D regular Kohonen map for the codebook. In the case of M-QPSK, we should use a 1-D circle Kohonen map. If the transmission system is binary, we can use a hypercube Kohonen map whose neighborhood structure is the same as a hypercube graph.

If the number of channel waveforms is large, it may create some difficulties in the implementation of the transmission system. In such case, we can use two channel waveforms to represent a codevector[5]. For example, if we use 16-QAM in the communication system, we can take the cartesian product of two 2-D 4×4 regular Kohonen maps as the neighborhood structure of a 256-codevectors Kohonen map, the neighborhood structure of which is a 4-D regular Kohonen map. In the case of 16-QPSK, we can take the cartesian product of two 1-D circles 16-codevectors Kohonen maps as the neighborhood structure of a 256-codevectors Kohonen map, the neighborhood structure of which resembles an elastic doughnut. Table 8.4 shows some suggestions on the the communication system and the corresponding Kohonen map.

Table 8.4 Suggestions on the communication system and the corresponding Kohonen maps.

Communication system	No. of codevectors	Kohonen map
16-QAM	16	4 by 4 grid
8-QPSK	8	1-D circle
256-QAM	256	16 by 16 grid
16-QAM	256	cartesian product of two 4 by 4 grids
16-QPSK	256	cartesian product of two 1-D circles
binary	256	8-D hypercube

8.2.3 Computer Simulations

We will use six common natural images (which are Lena, Baboon, Pepper, Clown, Fruit, and F16) to study the noise in the received data for different communication systems: 256-QAM, 16-QAM, 16-QPSK, and binary channel. We use the first three images to train five vector quantizers: 2-D 16×16 regular Kohonen map (for 256-QAM), 4-D $4 \times 4 \times 4 \times 4$ regular Kohonen map (for 16-QAM), the cartesian product of two 1-D circles Kohonen map (for 16-QPSK), hypercube Kohonen map, and LBG

[5]When we use two channel waveforms to represent a codevector, the transmission rate of communication system will be twice the rate of input vectors.

(for comparison). Each image is divided into a number of 2×4 blocks. There are 256 codevectors with dimension 8. The image is transmitted through a noise channel. The structure of Kohonen maps matches that of the channel waveforms. In the LBG's cases, there is no neighborhood structure among the codevectors. Thus, we use the following assignment schemes for the LBG's case. In the first scheme, we randomly assign a channel waveform to each codevectors in the case of LBG. In the second scheme, we use simulated annealing (SA) [84, 85] to carry out the association.

Table 8.5 summarizes the RMSE of the reproduction images under the noise-free channel. From the table, it is difficult for us to determine which quantizer is the best.

Table 8.5 The RMSE of the quantization images for different quantizers. The code rate is 1 bit per pixel.

Image	LBG	2-D regular Kohonen map	4-D regular Kohonen map	Hypercube Kohonen map	the cartesian product of two 1-D circles
Lena	6.5030	6.5436	6.7190	6.9246	6.6795
Baboon	13.0717	13.8219	13.9386	14.0197	14.2567
Pepper	6.5065	6.4925	6.7632	7.0287	6.6749
Clown	12.4689	11.9173	12.2198	12.7004	12.2496
Fruit	9.6516	8.6087	9.1645	9.6956	8.8468
F16	8.7842	8.4689	8.5366	8.7701	8.6463

However, when the communication channel is noisy, the results are totally different. Figure 8.12 and Figure 8.13 show the typical reproduction images under the noisy channel of a 256-QAM communication system. Figure 8.12 is the case of using 2-D regular Kohonen map with considering the matching. Figure 8.13 is the case of using LBG without considering the matching (random assignment). In Figure 8.12, there is no significant impulsive noise. On the other hand, there is a lot of impulsive noise in Figure 8.13. The RMSE of the received images under different communication systems and noise levels are shown in Figure 8.14 to Figure 8.17.

The matching neighborhood structure between the channel waveforms and the codevectors can greatly reduce the RMSE in the received images under a noisy channel. The improvement is dramatic when the noise level of the communication system is large. The performance of using Kohonen map is still a little bit better that of using SA. **Note that SA is an computational intensive method. Also, we need to do the SA again once we change the codebook.**

It seems that the better improvement can be achieved by using a Kohonen map with a lower dimension. For example, compared with other Kohonen maps, the RMSE of using hypercube Kohonen map is relatively higher (but, in general, it is still better than LBG'cases with SA and random assignment). It may be because the hypercube Kohonen map does not form a good ordering preserve. From Figure 8.14 to Figure 8.16 (in the cases of carrier modulation), under the same RMSE value in the received images, our approach is nearly 4-5 dB lower in terms of signal-to-noise-ratio in the channel.[6] In Figure 8.14, the RMSE of the received image 'Lena' under our approach is about 8.5 when the signal-to-noise-ratio in the channel is 22 dB. For LBG's case with random assignment, to maintain the same RMSE value in the received image, the signal-to-noise-ratio in the channel should increase to 26 dB or more.

[6] Note that dB is in the logarithmic scale.

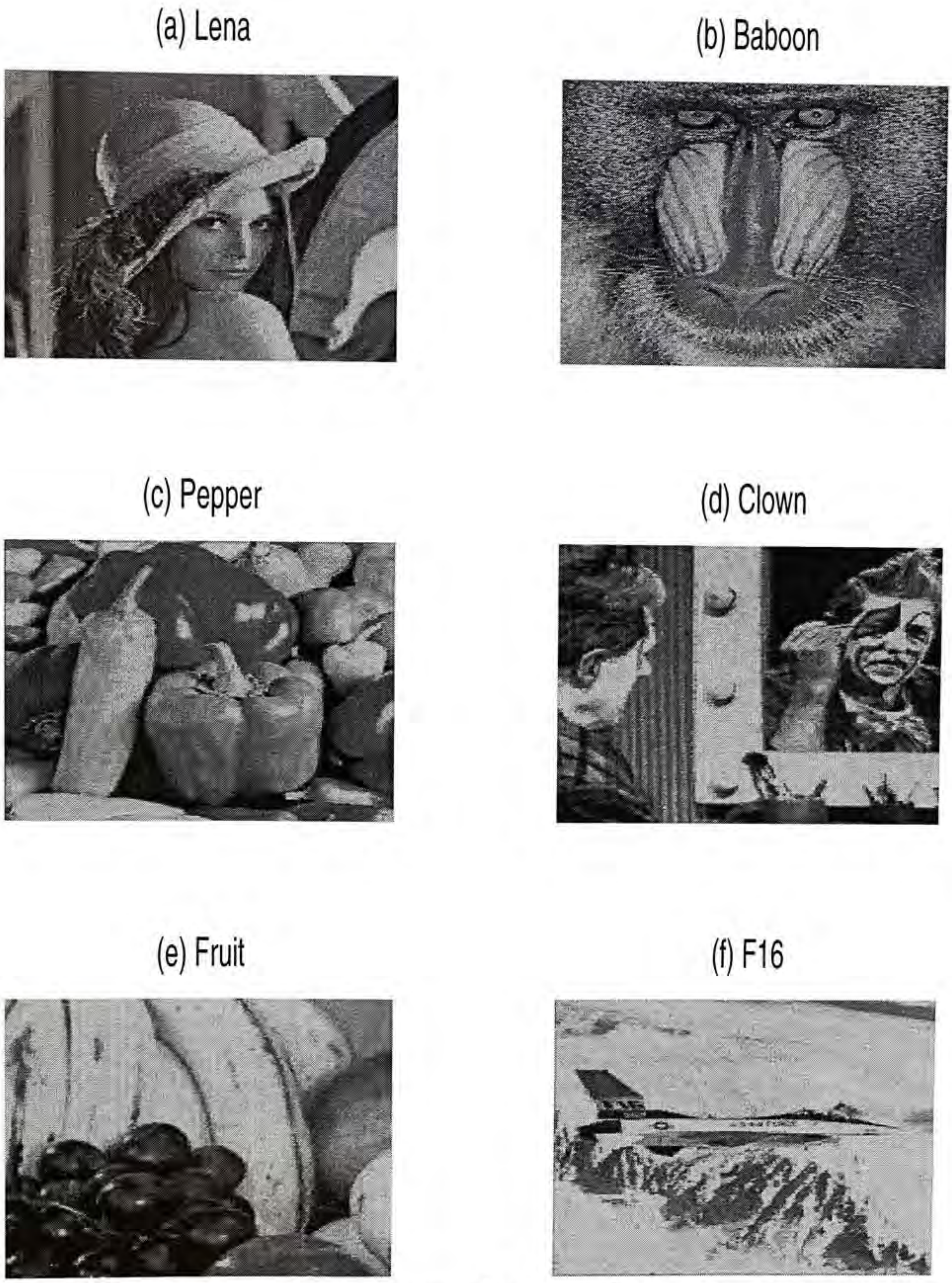


Figure 8.12 The received images of using 2-D regular Kohonen map with considering the matching under a noisy 256-QAM communication system, where the signal-to-noise-ratio in the channel is 23.3 dB.

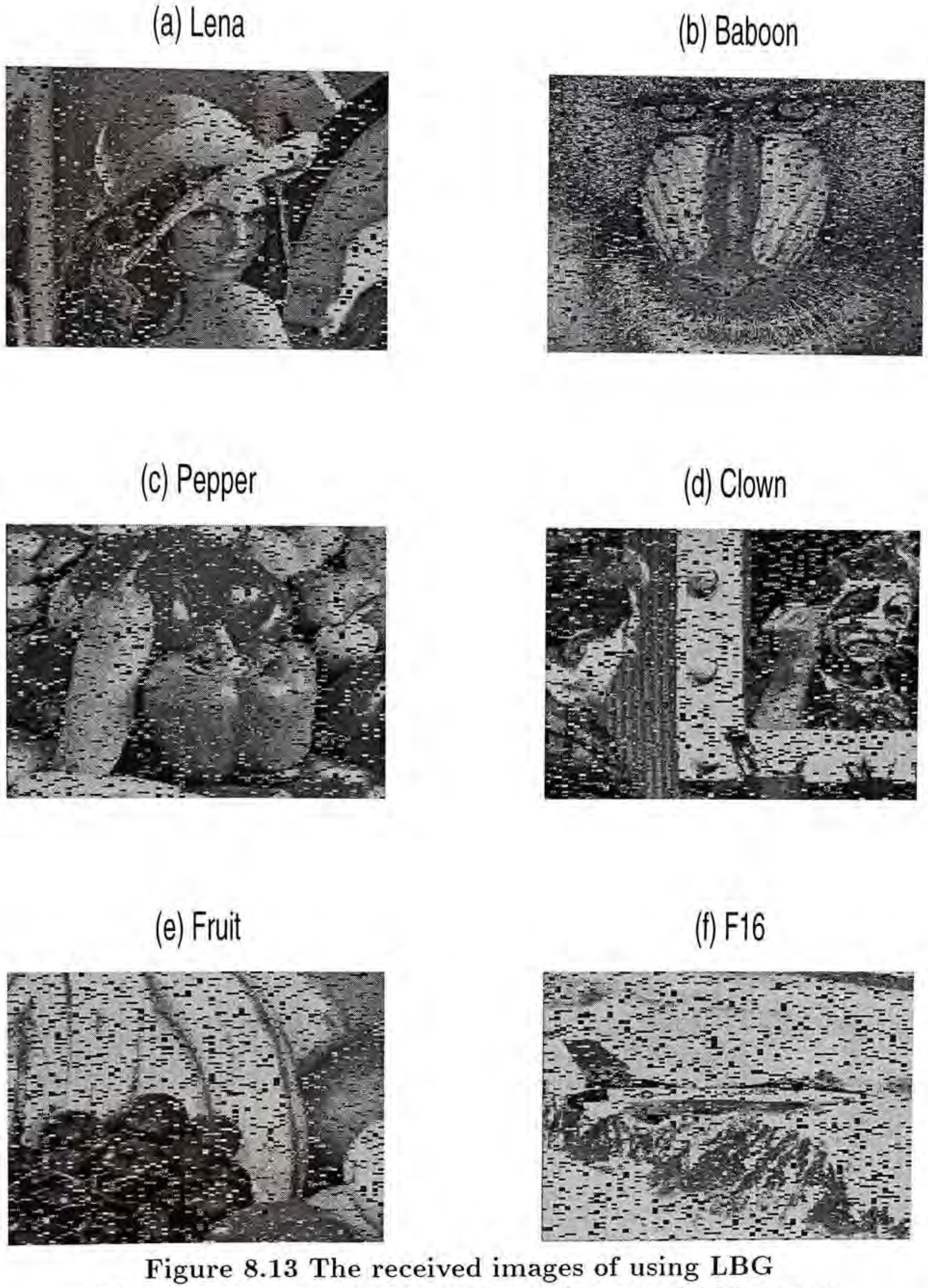


Figure 8.13 The received images of using LBG with random assignment under a noisy 256-QAM system, where the signal-to-noise-ratio in the channel is 23.3 dB.

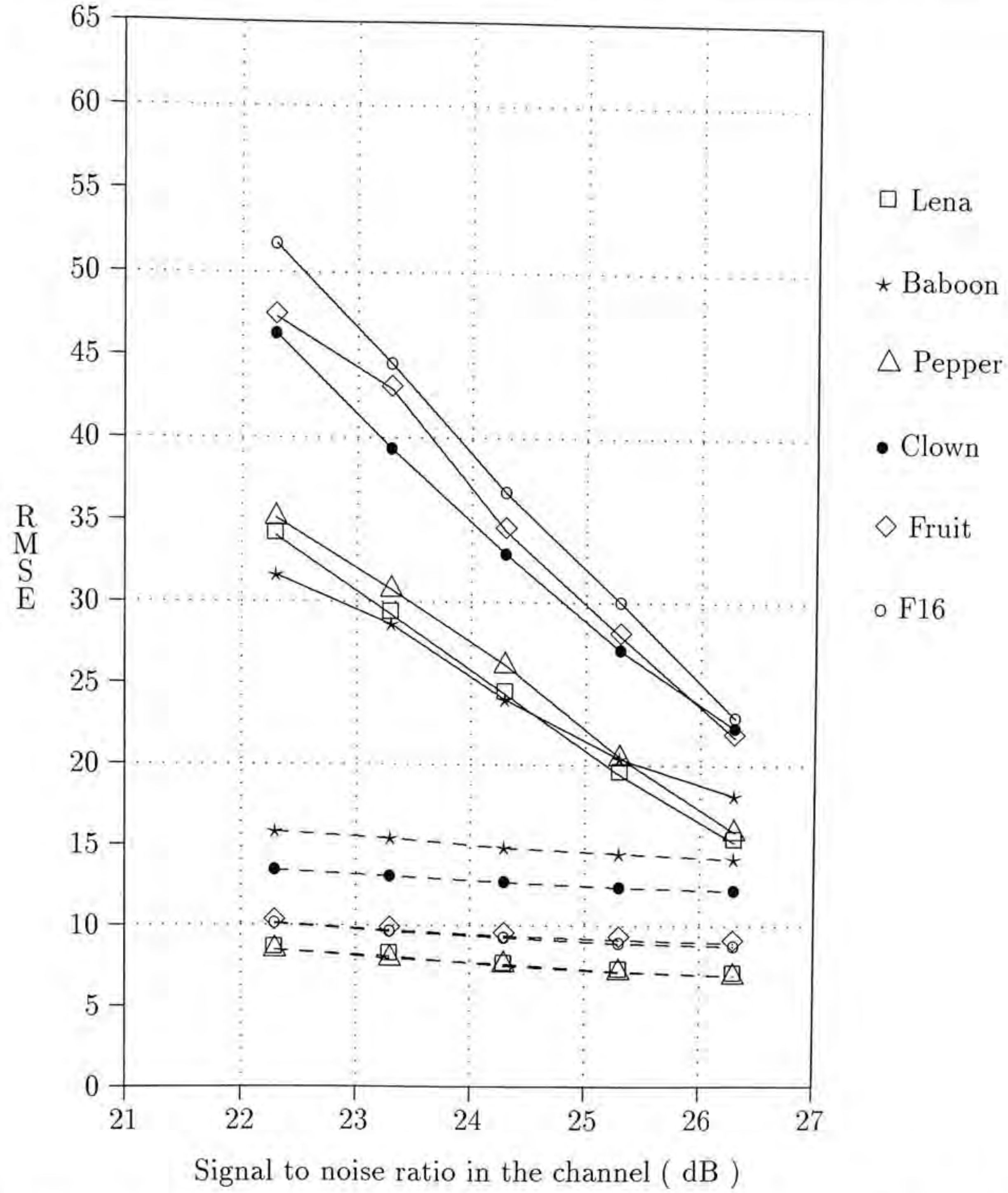


Figure 8.14(a) The RMSE of the received images under a noisy 256-QAM system. The solid-line is for the LBG with random assignment. The dashes-line is for the 2-D 16×16 regular Kohonen map with considering the matching.

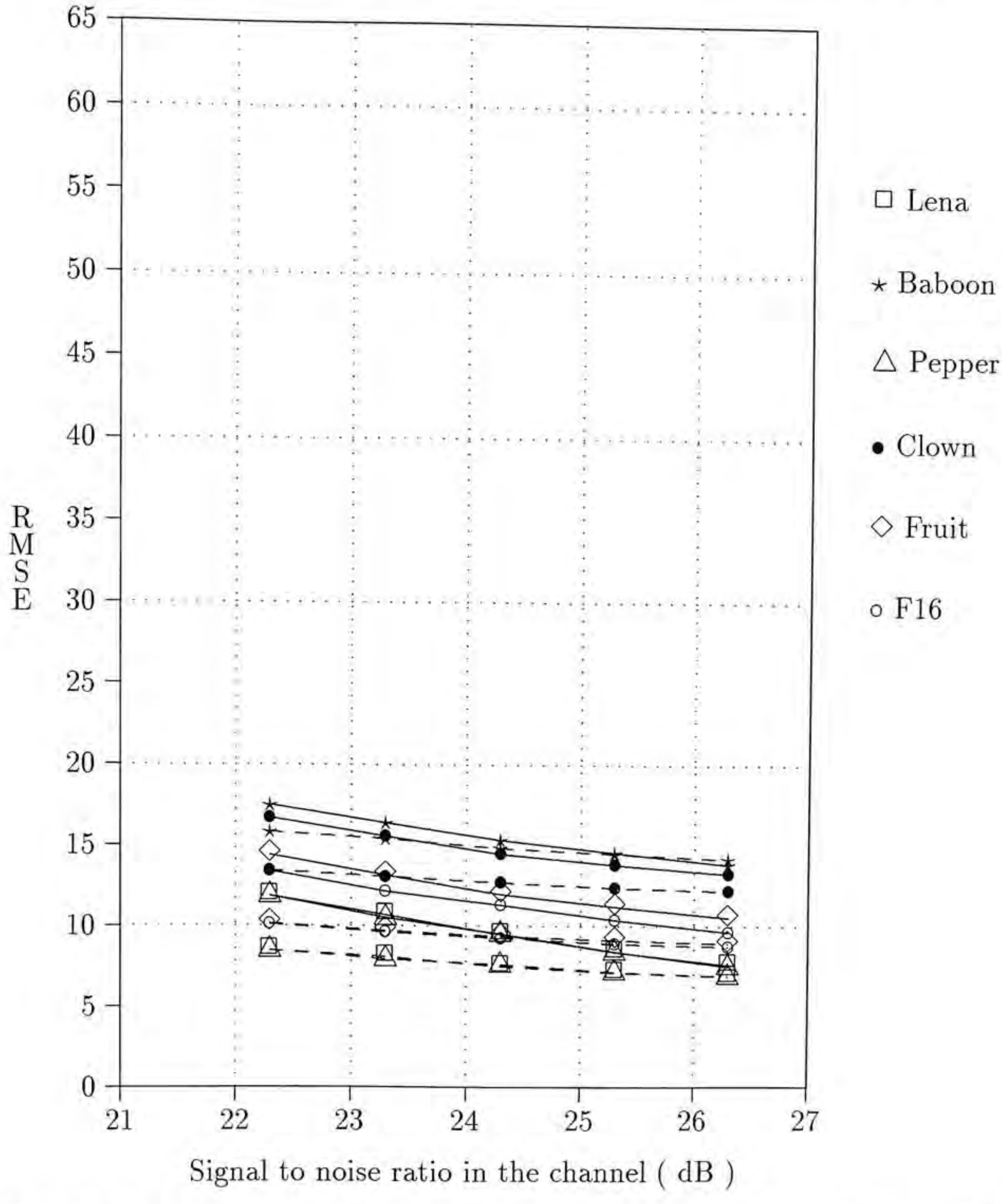


Figure 8.14(b) The RMSE of the received images under a noisy 256-QAM system. The solid-line is for the LBG with SA. The dashes-line is for the 2-D 16×16 **regular Kohonen map with considering the matching.**

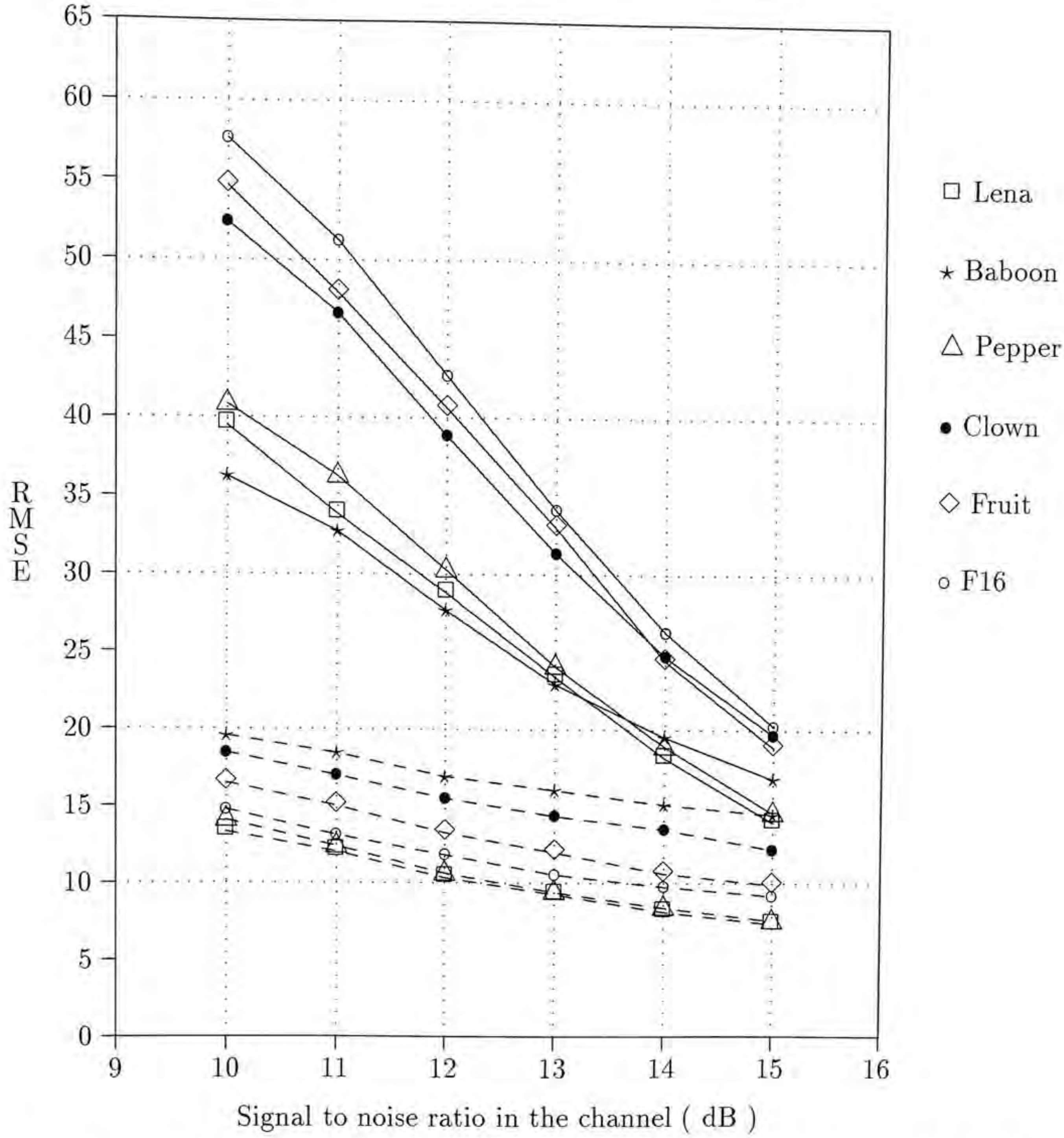


Figure 8.15(a) The RMSE of the received images under a noisy 16-QAM communication system. The solid-line is for the LBG with random assignment. The dashes-line is for the 4-D $4 \times 4 \times 4 \times 4$ regular Kohonen map with considering the matching.

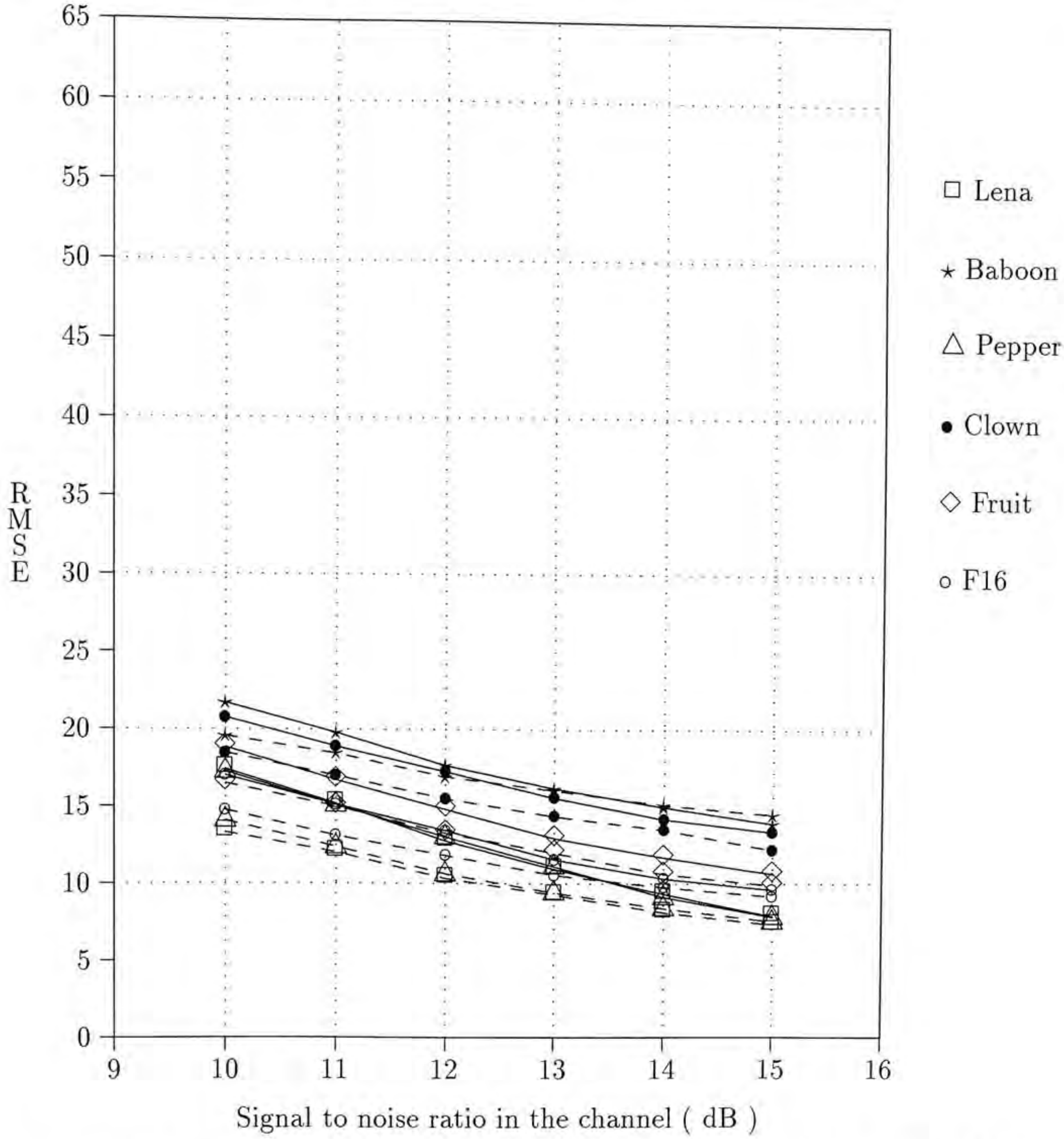


Figure 8.15(b) The RMSE of the received images under a noisy 16-QAM communication system. The solid-line is for the LBG with SA. The dashes-line is for the 4-D $4 \times 4 \times 4 \times 4$ regular Kohonen map with considering the matching.

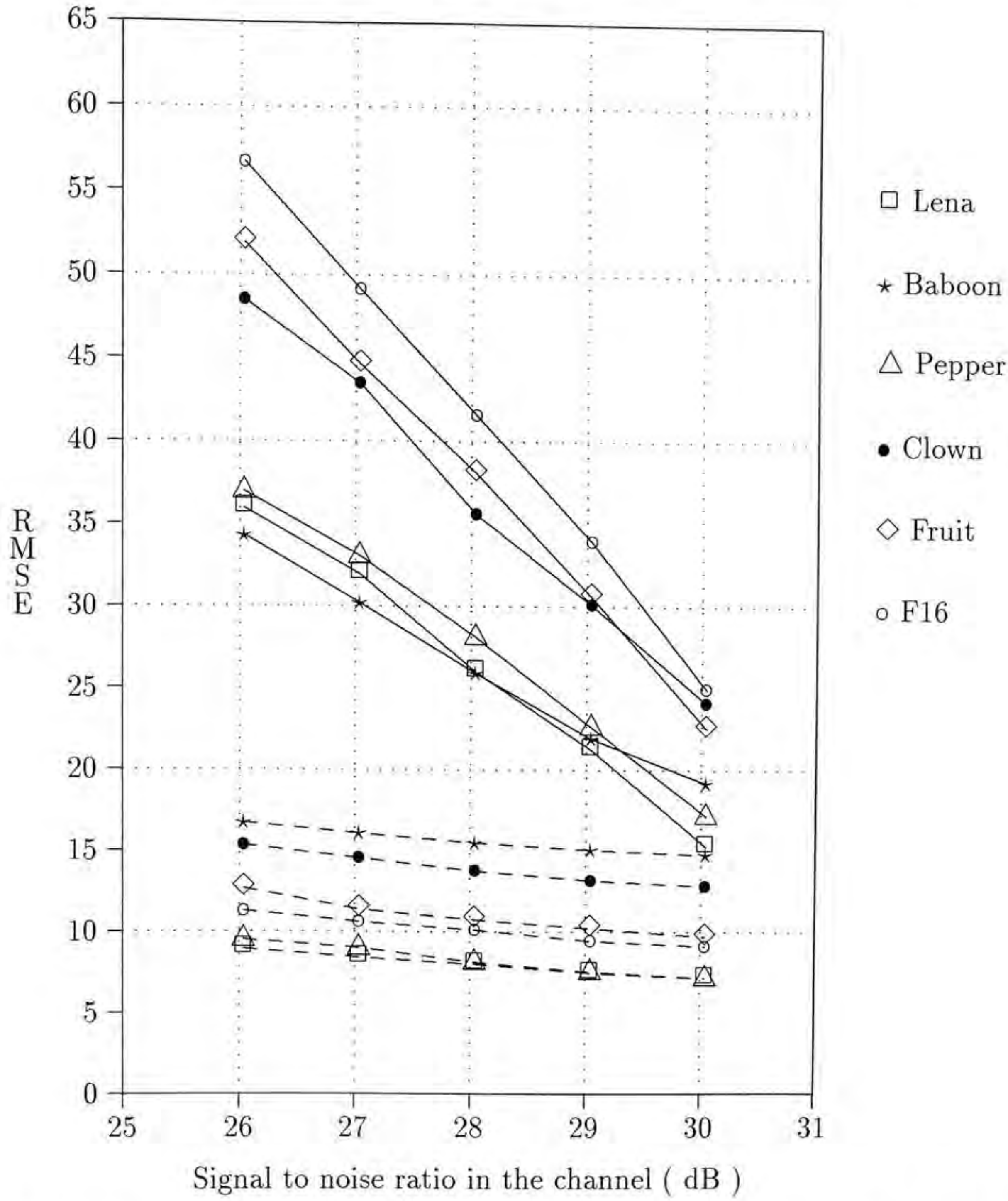


Figure 8.16(a) The RMSE of the received images under a noisy 16-QPSK communication system. The solid-line is for the LBG with random assignment. The dashes-line is for the cartesian product of two 1-D circles Kohonen map with considering the matching.

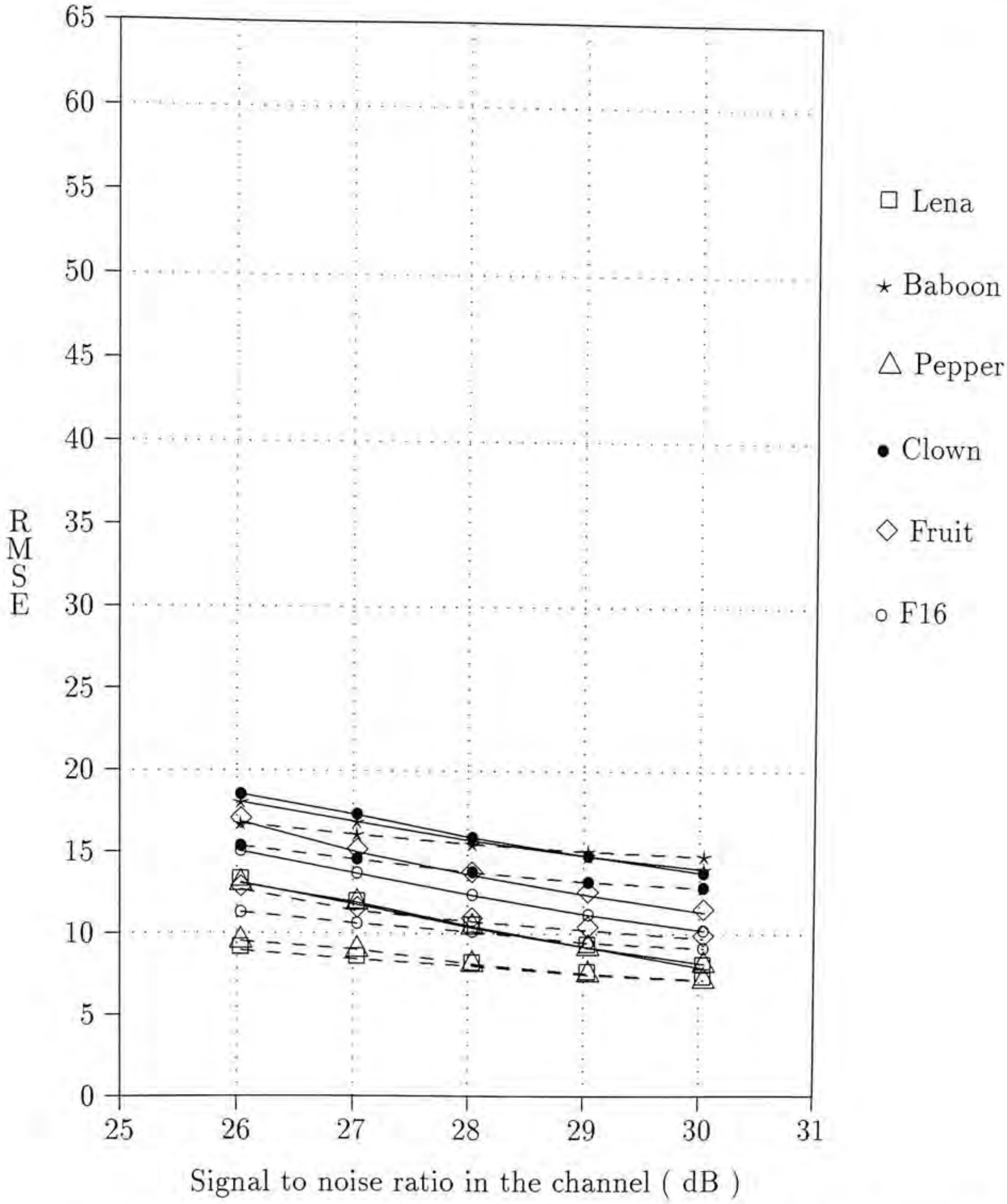


Figure 8.16(b) The RMSE of the received images under a noisy 16-QPSK communication system. The solid-line is for LBG with SA. The dashes-line is for the cartesian product of two 1-D circles Kohonen map with considering the matching.

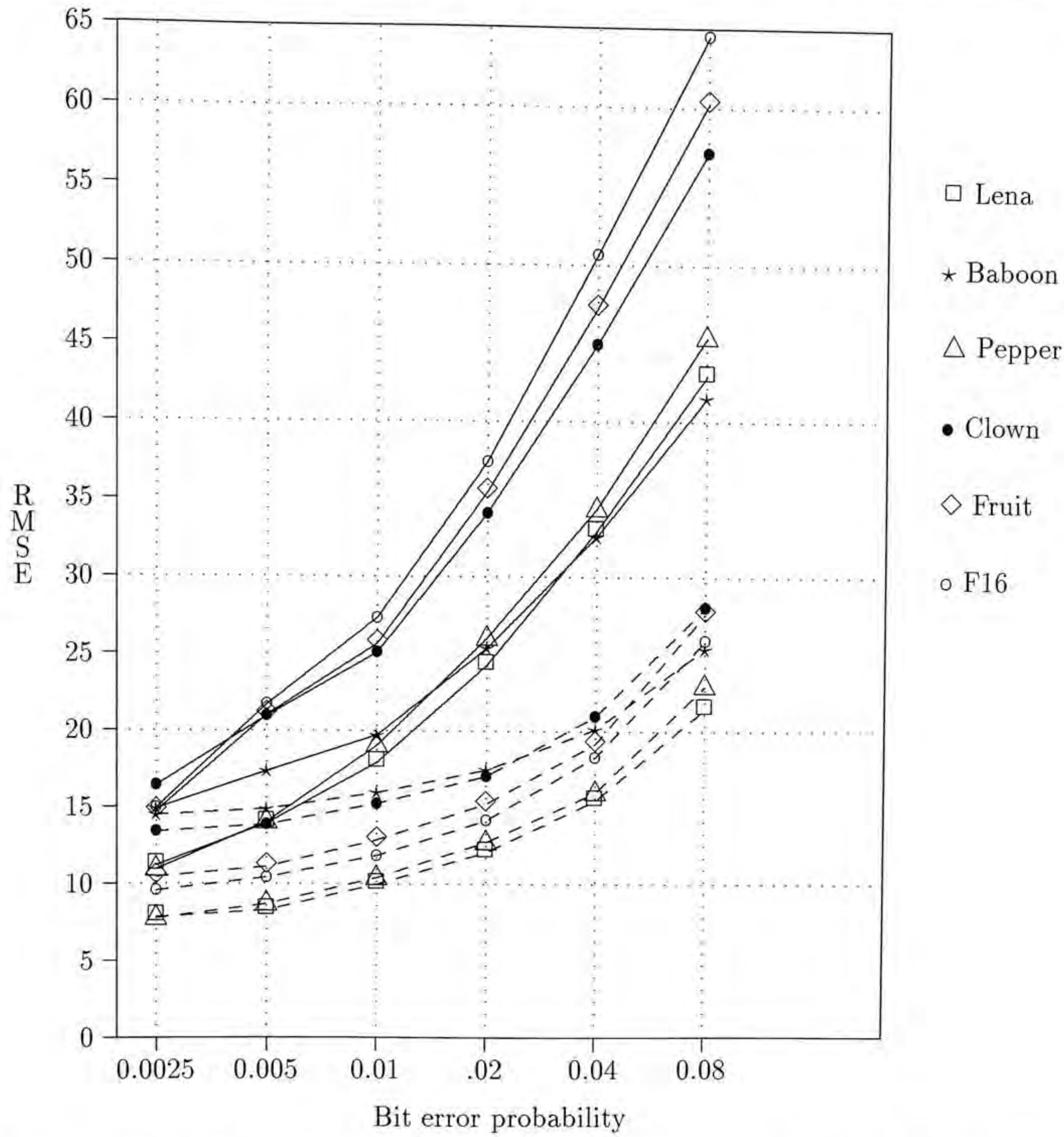


Figure 8.17(a) The RMSE of the received images under a noisy binary channel. The solid-line is for the LBG with random assignment. The dashes-line is for the hypercube Kohonen map with considering the matching.

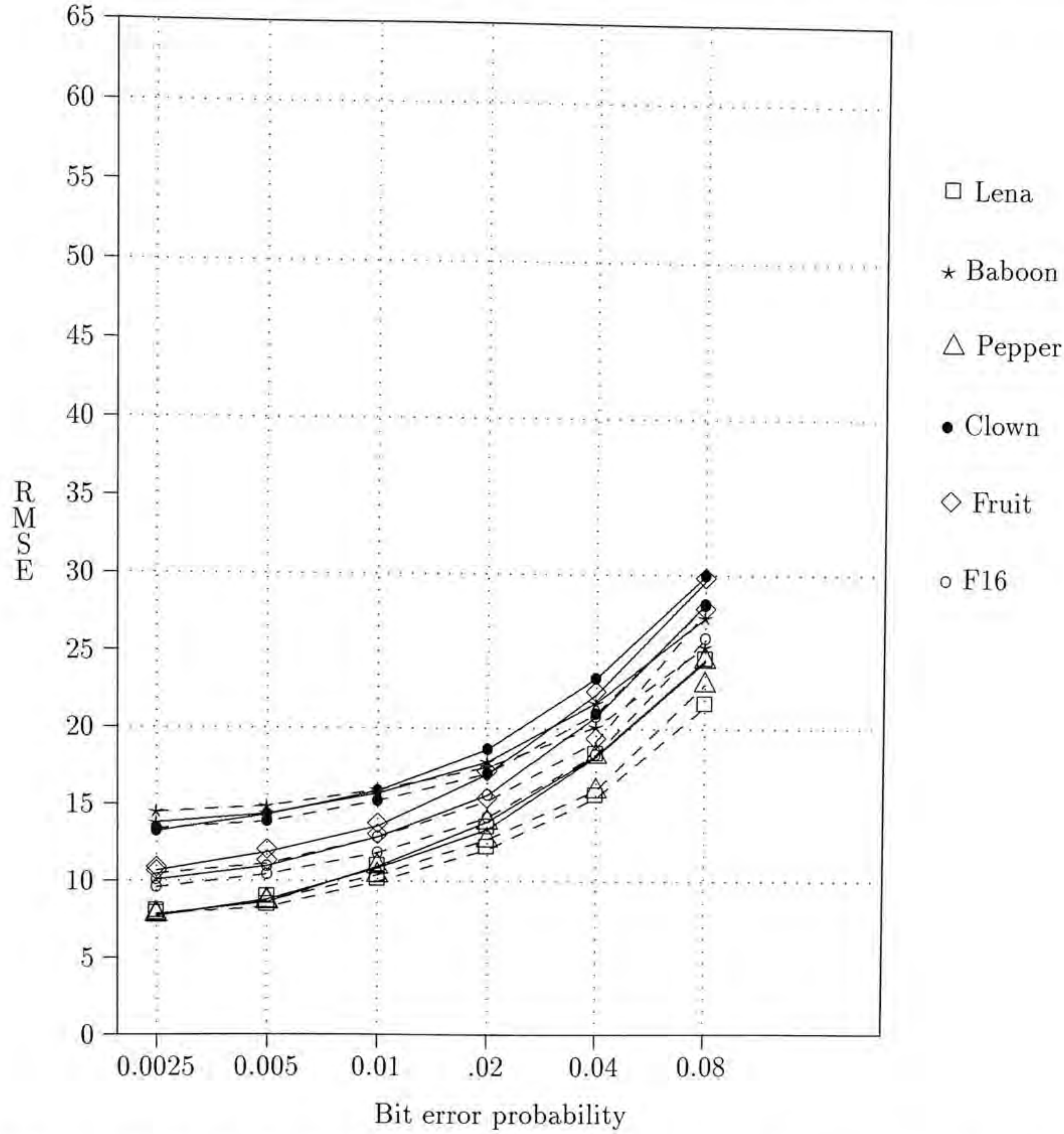


Figure 8.17(b) The RMSE of the received images under a noisy binary channel. The solid-line is for the LBG with SA. The dashes-line is for the hypercube Kohonen map with considering the matching.

8.3 Error Control Scheme for the Transmission of Vector Quantized Data

Nowadays, there is a trend to combine all the elements in the communication system as a whole such that more improvement can be achieved. A typical example is the concept of TCM [80] which combines the modulation and error control together so that the bit error rate is lower. Another example is the work presented in Section 8.2 or [84, 85] which combines the source coding (vector quantizer) and modulation together so that the noise in the data received is lower.

The last example is a joint source/channel coding system for the transmission of the scalar quantized data [87]. In this example, the source coding, the error control, and the modulation are designed as a whole. In [87], a joint source/channel coding system constructed using TCQ and TCM is described. The same trellises are used in the TCQ and TCM systems. There is a straightforward mapping of TCQ codewords to TCM symbols. Hence, the RMSE in the data received is very small. This approach is well suitable for the scalar quantizer because there is an ordering property in the scalar quantizer. Hence, the association between the TCQ's outputs and the 1-D TCM's channel waveforms can be done in a simple way. However, for the vector quantizer or TCVQ, there does not exist such regular structure among the codevectors. We cannot easily combine all the three elements (the source coding, the error control, and the modulation) together. In this section, we suggest to use the neighborhood structure of Kohonen map to achieve this task.

In Section 8.2, we have presented a transmission scheme for vector-quantized data by exploiting the matching between the neighborhood structure of the codebook and that of the channel waveforms. In this transmission scheme, the impulsive noise in the received data can be greatly reduced under a noisy channel even if we do not use any error control scheme. In this section, we will present an error control scheme which is best fit this transmission scheme. The proposed error control scheme is based on the concepts of TCM, TCKM, and matching neighborhood structure between the codebook and the channel waveforms. Under this error control scheme, the source coding, the error control, and the modulation are designed as a whole such that the impulsive noise in the received data can be further reduced.

8.3.1 Motivation and Background

The task of a digital communication system is to provide a cost-effective system for transmitting information from a sender at a rate and a level of reliability that is acceptable to an user. In the transmission of binary data, the error control code[77] is usually used when the communication channel is noisy. The goal of the error control code is to minimize the effect of channel noise, that is, to minimize the number of errors in the received binary data.

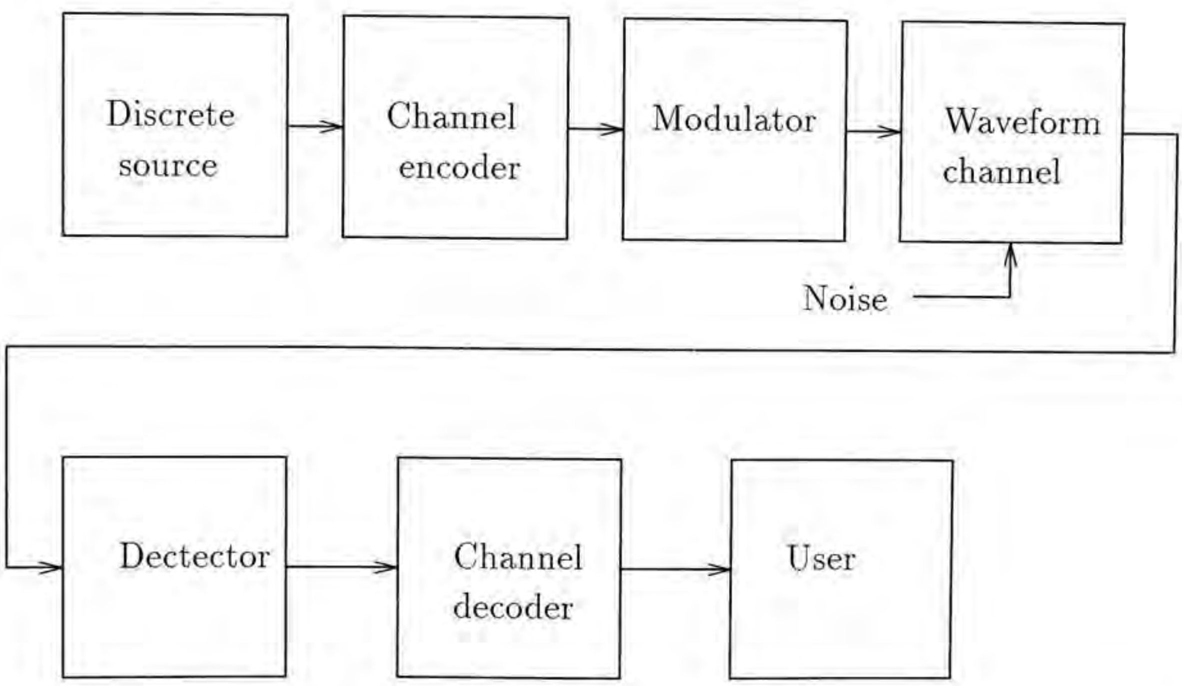


Figure 8.18 Simplified model digital transmission system. Coding and modulation performed separately.

Figure 8.18 shows the model of a digital communication system with an error control scheme. The channel encoder accepts message bits and adds redundancy according to a prescribed rule, thereby producing encoding data at a higher bit rate. The channel decoder exploits the redundancy to decide which message bit was actually transmitted. Practically, there are many different error-correcting codes [77]. Historically, these codes have been classified into block codes and convolutional codes. In the model depicted in Figure 8.19, the channel coding and modulation are performed separately. Recently, many researchers discover that the most effective method of error control coding is to combine it with modulation as a single function [80], as shown in Figure 8.19. In such approach, an error control code is redefined as a process of imposing certain patterns on the transmitted signal. The issue of combined modulation and error control code is called the TCM. All the error control techniques mentioned above is fitted for the transmission of binary data (minimizing the bit error rate).

However, in the transmission of vector-quantized data, lower bit error rate does not imply a lower impulsive noise in the received vector-quantized data. **When we look back the concepts of TCM and TCKM, we find that the output sequence of TCKM can be regard as the output of TCM.** Hence, it may be possible to create an error control scheme for the transmission of vector-quantized data on the basis of TCM and TCKM. Before we develop this error control scheme, we will first briefly introduce the concept of TCM.

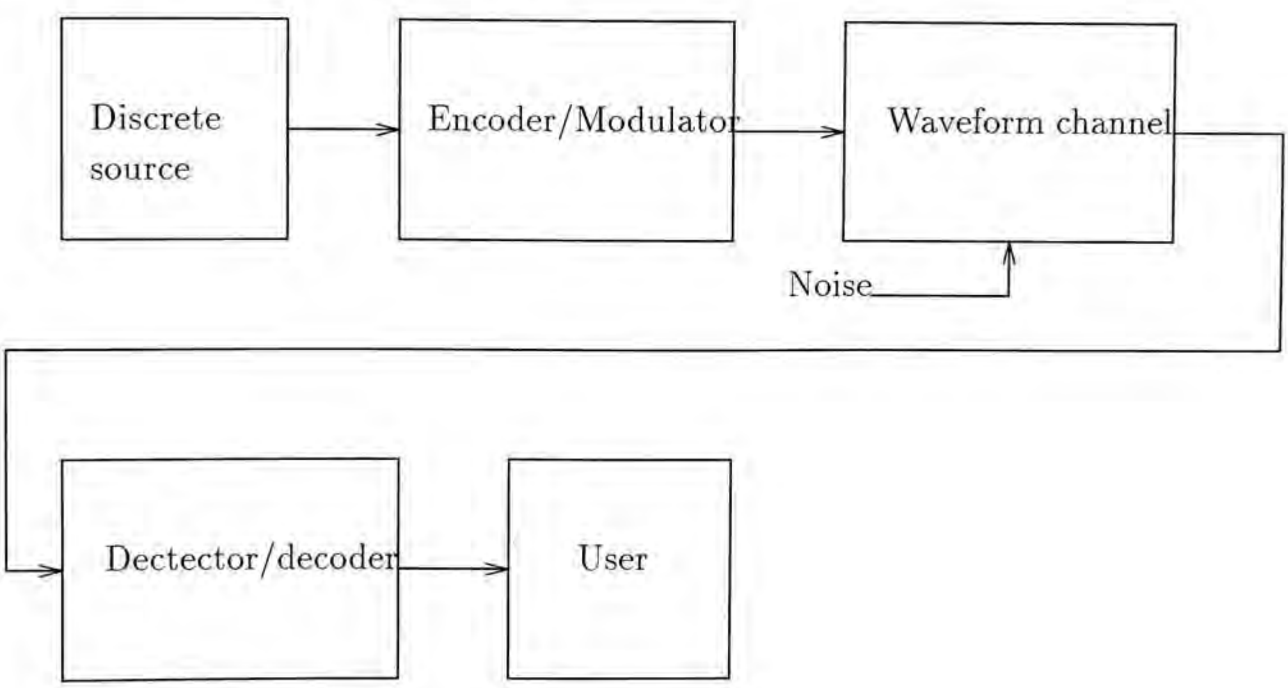


Figure 8.19 Simplified model digital transmission system. Coding and modulation combined.

8.3.2 Trellis Coded Modulation

We will introduce the concept of TCM [80] by an example. Consider a digital communication scheme used to transmit data from a source which emits two information bits every T seconds. Several solutions are possible (Figure 8.20).

- (a) Use a 4-QPSK modulation, with one signal every T seconds. Hence, every signal carries two information bits.
- (b) Use a convolutional code with rate 2/3 and 4-QPSK modulation. Now, every signal carries 4/3 information bits and then it must have a duration of $2T/3$ to match the information rate of the source. Hence, the bandwidth increases by a factor of 3/2.
- (c) Use a convolutional code with rate 2/3 and 8-QPSK modulation to avoid reducing the signal duration. Each signal carries two information bits, and hence no bandwidth expansion is incurred because 8-QPSK and 4-QPSK are with the same bandwidth.

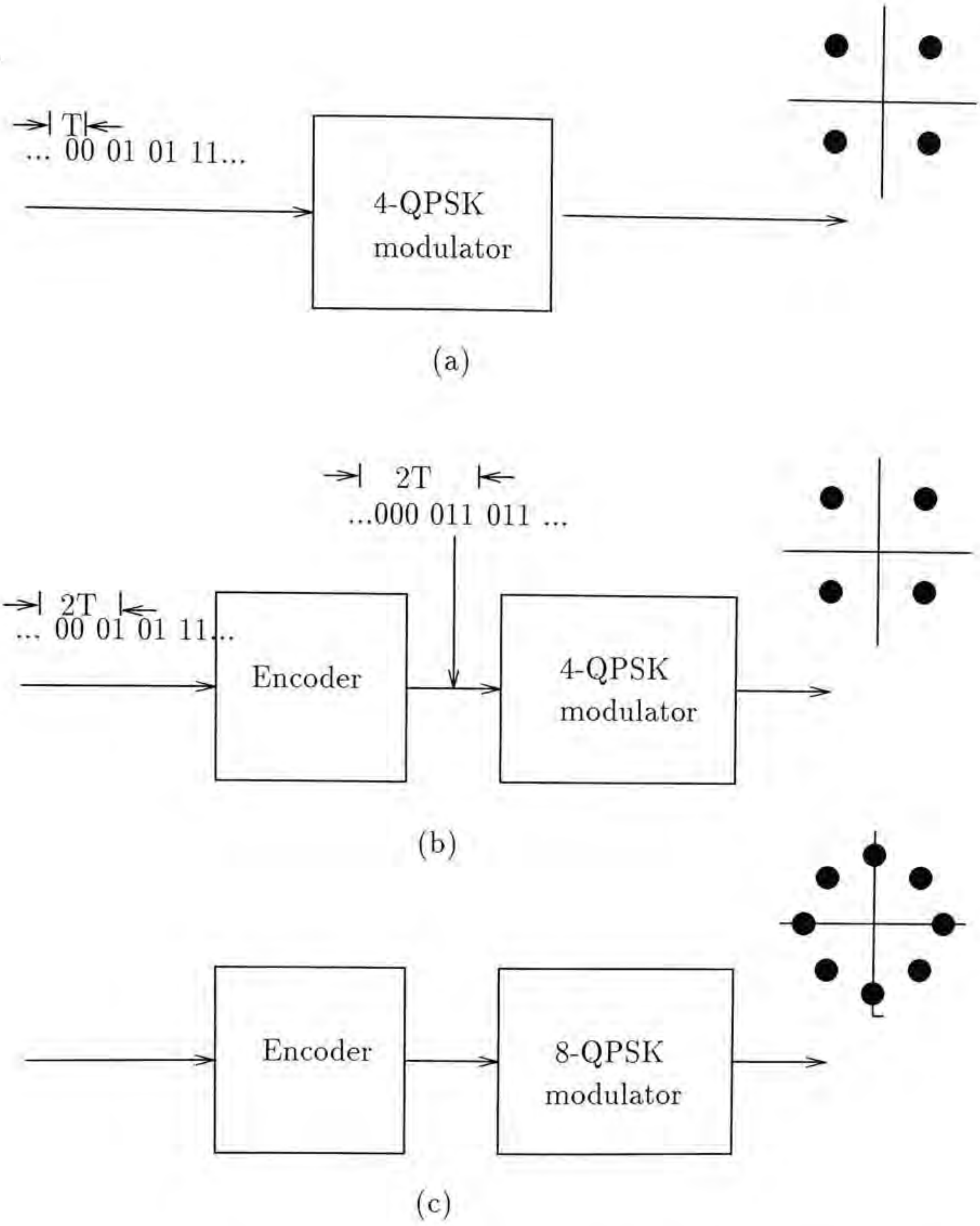


Figure 2.20 Three digital communication schemes transmitting 2 bits every T seconds: (a) uncoded transmission with 4-QPSK; (b) 4-QPSK with a rate 2/3 convolutional encoder and bandwidth expansion; (c) 8-QPSK with a rate 2/3 convolutional encoder and no bandwidth expansion.

With solution (c), we can use a coding scheme without bandwidth expansion. We might expect that the use of a larger set of signals (channel waveforms) would involve a power penalty with respect to 4-QPSK and then the coding gain achieved by the convolutional code should be offset by this penalty. However, the overall results is that we can get some coding gain at no price in bandwidth. [7]

The key aspect of TCM is the concept that convolutional encoding and modulation should not be treated as separate entities, but as an unique operation. Thus, the received waveforms, instead of being first demodulated and then decoded, is processed by a receiver that carries out the demodulation and decoding in a single step. Consequently, the parameter governing the performance is not the free Hamming distance of the convolutional code, but rather the free Euclidean distance between transmitted waveform sequences. Thus, the design of TCM will be based on Euclidean distances rather than on Hamming distances, so that the choice of the code and of the signal set (the set of channel waveforms) will not be performed separately. Finally, the detection process will involve soft decisions rather than hard.[8]

Consider a source emitting one of M' symbols (the set of source symbols is Ω') at a time instant. In the TCM, the number of the channel waveforms used, denoted as M, is twice the size of Ω'. The set of channel waveforms is denoted as $\Omega = \{s_1(t), \cdots, s_M(t)\}$. The waveform $y_n(t) \in \Omega$ transmitted at the n-th time interval (the discrete time n) depends not only on the source symbol a_n transmitted at the same time interval, but also on a finite number of previous source symbols:

$$y_n(t) = f(a_n, a_{n-1}, \cdots, a_{n-L}). \tag{8.5}$$

where $y_n(t) \in \{s_1(t), \cdots, s_M(t)\}$. By defining

$$\beta_n = (a_{n-1}, \cdots, a_{n-L}) \tag{8.6}$$

[7]The coding gain of an error control code is, under a fixed bit error rate in the received data, the reduction of the signal-to-noise-ratio in the channel with respect to the uncoding transmission scheme.

[8]In hard decision, given a received signal sequence, we will first individually demodulate each received waveform of the received signal sequence into a symbol (or binary bits), and then decode the symbol sequence, based on the structure of the convolutional code and Hamming distance, into the original information sequence. In soft decision, given a received signal sequence, we will directly estimate the original information sequence, based on the whole received signal sequence and Euclidean distance.

as the state of the encoder at the discrete time n, a more compact form is

$$y_n(t) = f(a_n, \beta_n) \tag{8.7}$$

$$\beta_{n+1} = g(a_n, \beta_n)\,. \tag{8.8}$$

The function $f(\cdot,\cdot)$ describes the fact that each transmitted waveforms depends not only on the corresponding source symbol, but also on the parameter β_n. In other words, at a time instant the transmitted waveform is chosen from a subset of Ω that is determined by the value of β_n. The function $g(\cdot)$ describes the memory part of the encoder and shows the evolution of the modulator (Figure 8.21).

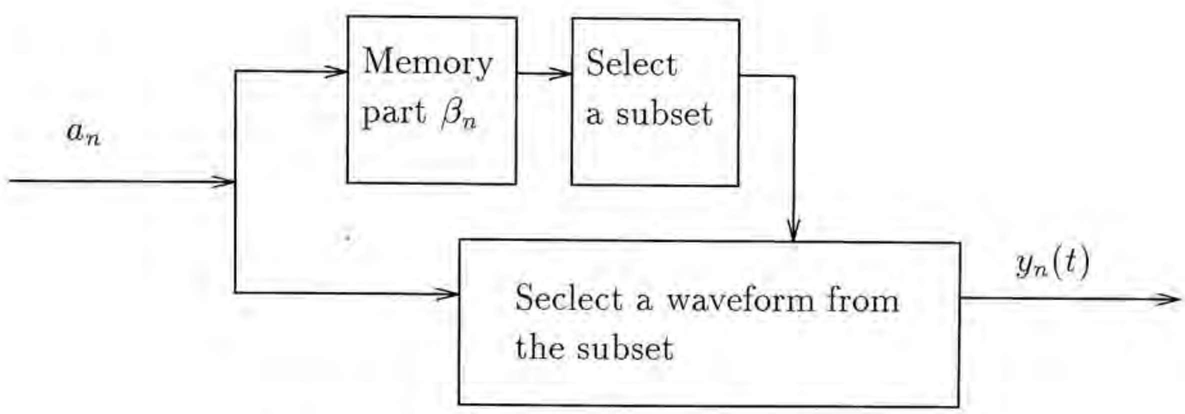


Figure 8.21 General model for the TCM.

For the graphical representation of the functions f and g it is convenient to use a trellis. The values that can be taken by β_n, the encoder state at time n, are the nodes of the trellis. With each source symbol we associate a branch that stems from each modulator state at time n and reaches the encoder state at time $n+1$. The branch is labeled by the corresponding values of f (that is, the corresponding channel waveforms $s_i(t)$). The trellis structure is determined by the function g, while f describes how channel waveforms are associated with each branch along the trellis. The association between the branches and channel waveforms is based on the concept of the set partition (see Section 8.1 or [80]).

If the source symbols are M'-ary, each node must have M' branches stemming from it (one per source symbol). This implies that in some cases two or more branches connect the same pair of nodes; when this occurs, we say that parallel transitions take place. In other notations, we can use a branch to represent two or more waveforms instead of using the concepts of 'parallel branches'.

Figure 8.22 shows an example of this representation. It is assumed that the encoder has four states, the source emits binary symbols, and four channel waveforms: $s_1(t), \cdots, s_4(t)$ are used. The number (inside the bracket) above each branch shows the corresponding source symbol. For example, if the current state is the top state in Figure 8.22 and the current input symbol is "0", then the corresponding output waveform is $s_1(t)$. The optimum decoding is the search for the most likely path through the trellis once the received waveform sequence has been observed at the channel output. This search is best done using Viterbi algorithm.

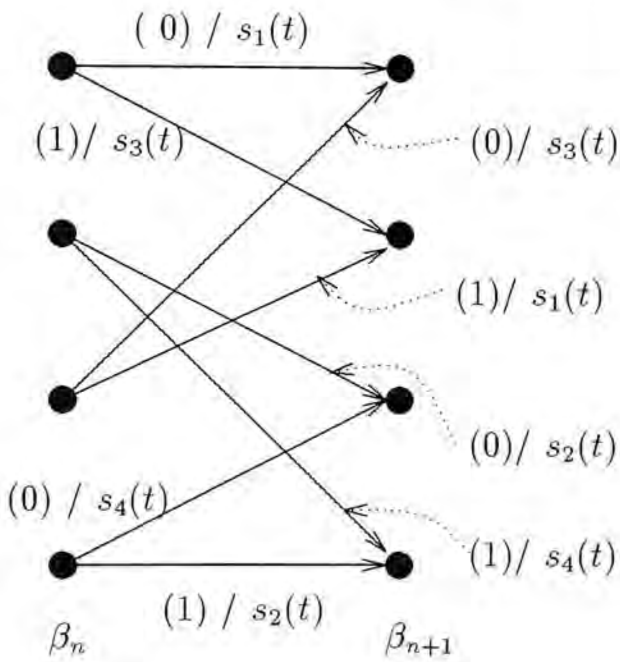


Figure 8.22 Example of a trellis describing a TCM scheme with four states and four channel symbols used to transmit from a binary source.

8.3.3 Combined Vector Quantization, Error Control, and Modulation

Consider the TCKM proposed in Section 8.1, the source vector sequence $S = \{\vec{x}_1, \cdots, \vec{x}_n\}$ is quantized to a codevector sequence $\hat{S} = \{\vec{y}_1, \cdots, \vec{y}_n\}$ based on Viterbi algorithm, where $\vec{y}_i \in \{\vec{c}_1, \cdots, \vec{c}_M\}$. For a digital communication system, we can encode the codevector sequence to a symbol sequence $\hat{S}'$ and then transmit it. If the channel is noise-free, we can use a symbol set with size $\frac{M}{2}$ to encode each output codevector of TCKM, where M is the size of the codebook. (In fact, we should transmit the initial state of the output vector sequence in the trellis of TCKM.). It is because, at a time instant, only $\frac{M}{2}$ codevectors are used to represent the source vector. Together with the initial state of the output vector sequence, we can decode the received symbol sequence back to the codevector sequence based on the structure of TCKM.

However, if the channel is noisy, there may be some errors in the received symbol sequence. Because of using trellis structure in the TCKM, an error in the received symbol sequence may cause many errors in the received codevector sequence. Also, the errors in the received codevector sequence are impulsive. To lower the error rate in the received symbol sequence, we can design an error control code for the transmission of the symbol sequence $\hat{S}'$. However, we still cannot reduce the impulsive noise in the received codevector sequence if the error control code is separately designed.

Now, if we use a symbol set J with size M

$$J = \{s_1, s_2, s_3, \cdots, s_M\} \tag{8.9}$$

to describe the codevector sequence $\hat{S} = \{\vec{y}_1, \cdots, \vec{y}_n\}$ of TCKM,[9] then we can obtain a new symbol sequence $\hat{S}'' = \{y_1, \cdots, y_n\}$ where $y_i \in J$. The symbol sequence $\hat{S}''$ can be regarded as the output of a convolutional coder with input $\hat{S}'$. Furthermore, if we associate each symbol s_i with a channel waveform $s_i = s_i(t)$ **based on the concept described in Section 8.2**, then the overall effect is that the TCM is used to code the symbol sequence $\hat{S}'$. In this view, the trellis of TCM is the same as the trellis of TCKM provided that we replace the output label in the trellis of TCKM (s_i) with the waveforms ($s_i(t)$). Hence, likely TCM error events (which are similar to the transmitted channel waveforms) cause only a small increase in overall RMSE in the received vector-quantized data,

Figure 8.23 shows the idea of this TCKM error control transmission system. For the sender, we use the TCKM to encode the input source sequence $S = \{\vec{x}_1, \cdots, \vec{x}_n\}$ into a waveform sequence $\hat{S} = \{y_1(t), \cdots, y_n(t)\}$, where $y_i(t) \in \{s_1(t), \cdots, s_M(t)\}$.

[9] That is, we associate each codevector $\vec{c}_i$ with the symbol $s_i \in J$.

Note that the codevector sequence $\hat{S} = \{\vec{y}_1, \cdots, \vec{y}_n\}$ and the symbol sequence $\hat{S''} = \{y_1, \cdots, y_n\}$ are hidden in the sender since the TCKM directly encodes the input vector sequence $S = \{\vec{x}_1, \cdots, \vec{x}_n\}$ to the waveform sequence $\hat{S}$. In other words, the TCKM in Figure 8.23 performs two functions: quantizer and channel encoder (for error control). In the receiver, we estimate the transmitted waveforms based on the trellis of TCM once the received waveform sequence has been observed at the channel output. Then, the output of the receiver is the estimated codevector sequence.

For the sender, the Viterbi algorithm is used to search the best codevector sequence in the original data space. On the receiver side, the Viterbi algorithm is used to search the best waveform sequence in the channel waveform space. In Figure 8.23, the source coder (TCKM), the error control code (TCM), and the modulation are designed as a whole system.

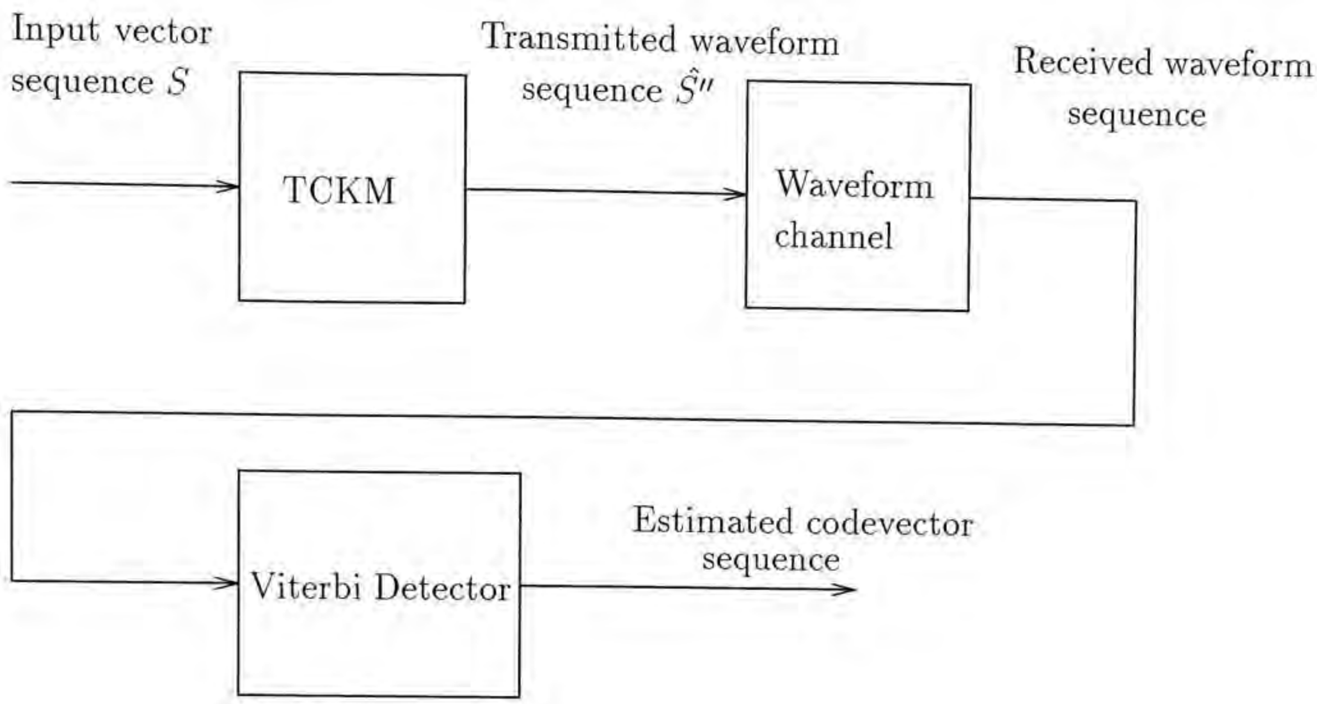


Figure 8.23 General model of the TCKM error control transmission system.

In this TCKM error control transmission system, the size of the codebook is M but at a time instant only $\frac{M}{2}$ codevectors can be used to represent the source vector. One might expect that the use of TCKM (at a time instant only $\frac{M}{2}$ codevectors can be used to represent the source vector) would involve an error penalty with respect to the use of Kohonen map presented in Section 8.2 (at a time instant all the M codevectors can be used to represent the source vector) with M channel waveforms. In the next section, the simulation shows that the gain of TCM (due to the error control scheme) in the TCKM error control transmission system can offset this penalty (due to the use of half codebook at a time instant). The net result is that the RMSE in the received codevector sequence can further be reduced when the signal-to-noise ratio in the channel belongs to a suitable range.

8.3.4 Computer Simulations

In this section, we will use the six common natural images (which are Lena, Baboon, Pepper, Clown, Fruit, and F16) to study the noise in the received data for the 8-state TCKM error control transmission system. There are two TCKM error control transmission systems: the 16-QPSK modulation (Kohonen map in TCKM is the cartesian product of two 1-D circles Kohonen maps), and the 256-QAM modulation (the Kohonen map in the TCKM is a 16×16 2-D regular Kohonen map). As a comparison, we also show the results of Kohonen map presented in Section 8.2.

Figure 8.24 and Figure 8.25 show the RMSE of the received images under the two approaches: the TCKM error control transmission systems, and the Kohonen map with considering the matching. Compared with our approach presented in Section 8.2, the TCKM error control transmission systems further reduce RMSE provided that the signal-to-noise-ratio in the channel belongs to a suitable range. For the case of 256-QAM, the improvement only occurs when the signal-to-noise-ratio in the channel is between 22.3–25.3 dB. For the case of 16-QPSK, the improvement occurs when the signal-to-noise-ratio is between 26–29 dB. Note that dB is in logarithmic scale. To illustrate the reason why the improvement only occurs in specific dB interval, we could chase the symbol error rate as shown in Table 8.6 and Table 8.7.

Table 8.6 The symbol error rates of the TCKM error control scheme and Kohonen map, where 256-QAM is used.

Symbol error rate		
signal-to-noise ratio in dB	TCKM 256-QAM	Kohonen map 256-QAM
21.3	0.35	0.37
22.3	0.21	0.28
23.3	0.061	0.20
24.3	0.0065	0.14
25.3	0.0011	0.083
26.3	0.0000	0.046

Table 8.7 The symbol error rates of the TCKM error control scheme and the Kohonen map, where 16-QPSK is used.

Symbol error rate		
signal-to-noise ratio in dB	TCKM 16-QPSK	Kohonen map 16-QPSK
25	0.31	0.39
26	0.16	0.30
27	0.041	0.23
28	0.0089	0.16
29	0.00011	0.093
30	0.0000	0.053

When the channel is noiseless (greater than 26 dB for 256-QAM), both symbol error rates of TCKM's case and Kohonen map's case are very small and then the RMSE in the received images mainly come from the quantization noise. Hence, the performance of TCKM error control transmission system will be poorer than that of Kohonen map with considering the matching. It is because in the TCKM only half of codebook is used at a time instant.

From Table 8.6 and Table 8.7, the error control scheme in the TCKM cannot correct the symbol errors when the channel is very noisy (less than 22 dB for 256-QAM, or less than 25 dB for 16-QPSK). Also, the symbol error rate of TCKM's case is similar to that of Kohonen map's case under very noisy channel. Therefore, the quantization noise from the quantizers also determines the overall RMSE. Since only half of codebook is used at a time instant in the TCKM, the performance of TCKM's case will be poorer than that of Kohonen map's case when the channel is very noisy.

Figure 8.26 and Figure 8.27 show the received images under the TCKM error control transmission systems when the signal-to-noise ratio in the channel is within the range above-mentioned. There is no significant impulsive noise in the received images because we also consider the matching neighborhood structure between the codevectors and channel waveforms in the TCKM error control transmission systems.

8.4 Chapter Summary

In this chapter, we have presented three new cross-relative applications of Kohonen map for data compress and communications.

- Based on the ordering property of Kohonen map, we have introduced a new trellis type coder: TCKM. The performance of TCKM is similar to that of the conventional TCVQ. The set partition of the codevectors in the TCKM is very simple. Particularly, the set partition in the TCKM is not necessary since we can use the existing trellis of TCM as the trellis of TCKM. Also, different TCKMs with the same neighborhood structure can share the same trellis.

- We have introduced a method to design transmission system for vector-quantized data by making use of the ordering property of Kohonen map. The basic philosophy is that the association between the codevectors and the channel waveforms is based on their neighborhood structures. Simulation shows that there is a significant reduction of noise in the received data by using this new approach. The advantage of this approach is that we no longer need an error correction algorithm in the communication system and yet can still obtain a great reduction of noise in the received data under the noisy channels. Also, different Kohonen maps with the same neighborhood can use the same association. It means that we do not need to carry out the association again even if we use a new codebook.

- Based on similarity between TCKM and TCM, we have developed an error control scheme for the transmission of vector-quantized data, called TCKM error control transmission system. The main feature of this transmission system is, the source coder, the error control, and the modulation together are designed as a whole. Compared with the approach presented in Section 8.2, the error in the received data can be further reduced when the signal-to-noise ratio in the channel belongs to a suitable range.

  The disadvantage of TCKM error control transmission system is its computational cost. It is because Viterbi algorithm has been used in both the sender side and the receiver side. If we can afford this cost, we can use this system. Otherwise, we can only use the approach proposed in Section 8.2 for the transmission of vector-quantized data. It is because the approach presented in Section 8.2 can still greatly reduce the RMSE in the received vector-quantized data without any error control scheme. Another disadvantage is that the improvement of RMSE only occurs in a specific noise interval in the channel. It will limit the operation range of TCKM error control transmission system. If the noise level in the channel is out of this specific interval, we should use the approach presented in Section 8.2.

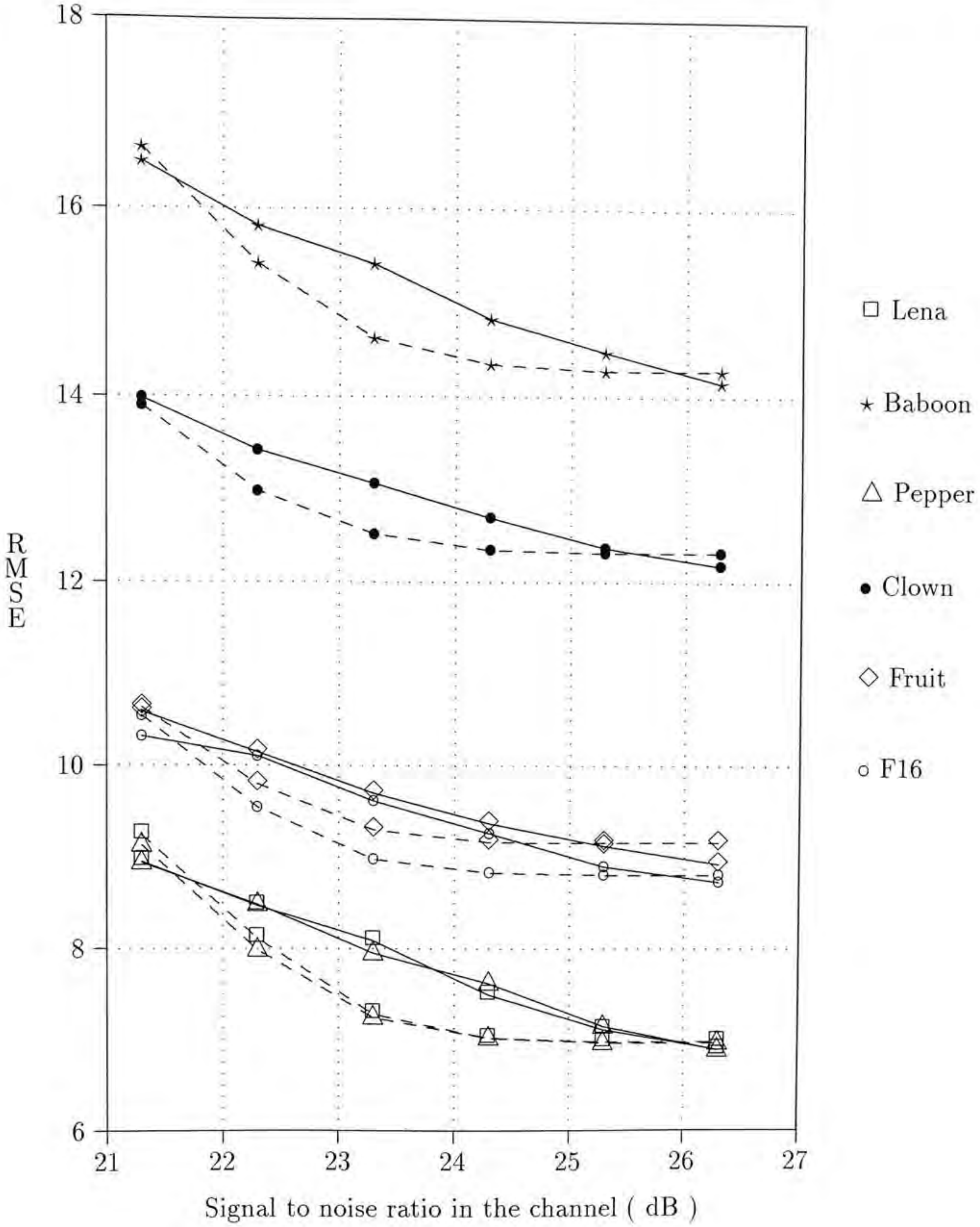


Figure 8.24 The RMSE of the received images under under a noisy 256-QAM system. The solid line is for the 2-D 16×16 regular Kohonen map with considering the matching. The dashes line is for the TCKM with 2-D 16×16 regular Kohonen map.

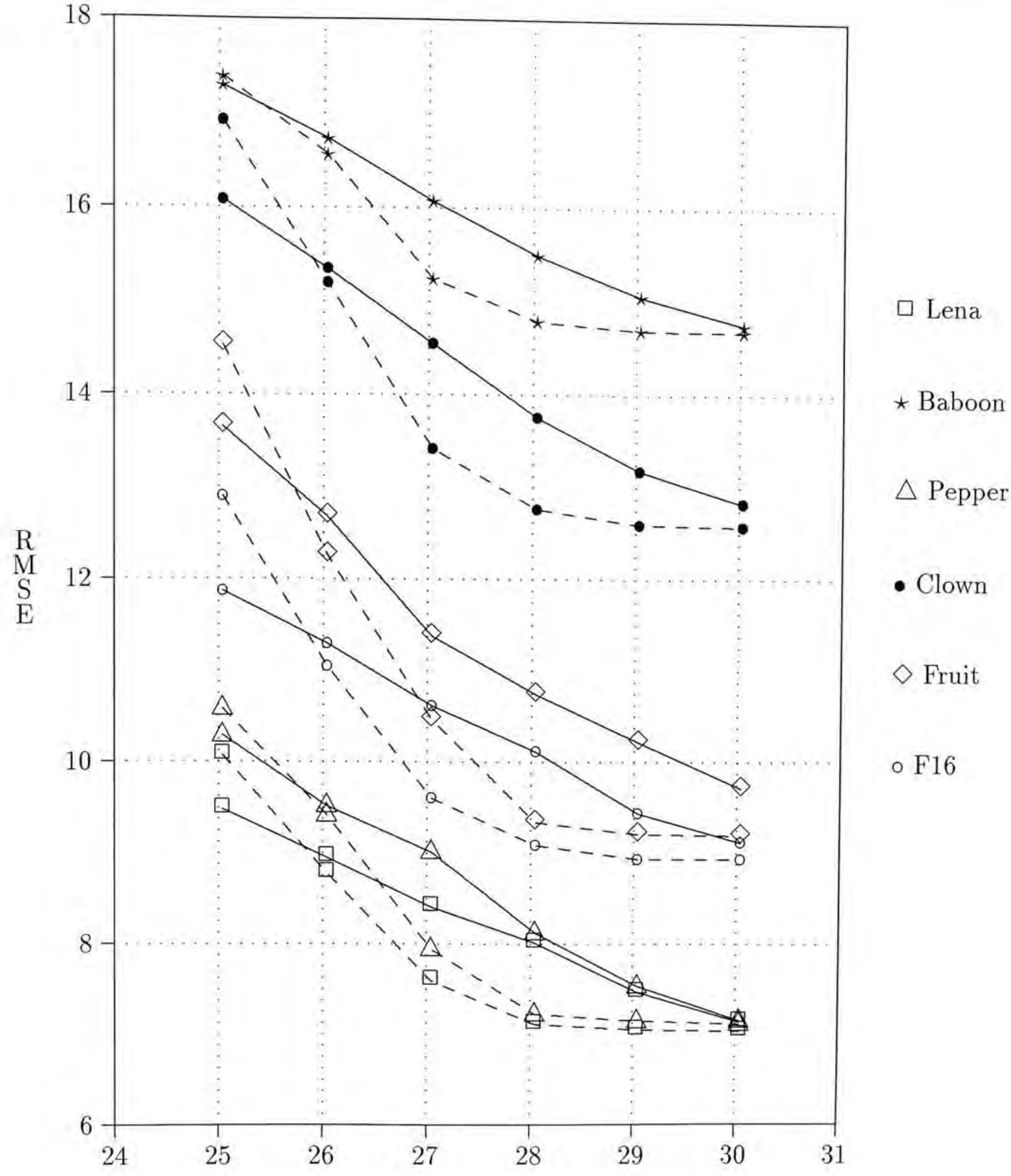


Figure 8.25 The RMSE of the received images under under a noisy 16-QPSK system. The solid line is for the cartesian product of two 1-D circles Kohonen map with considering the matching. The dashes line is for the TCKM with Cartesian product of two 1-D 16 circles Kohonen map.

Figure 8.26 The received image: Lena. TCKM with a 2-D regular Kohonen map is used. The communication system is a noisy 256-QAM and the signal-to-noise-ratio in the channel is 23.3 db.

Figure 8.27 The received image: Lena. TCKM with a Cartesian product of two 1-D circles Kohonen maps is used. The communication system is a noisy 16-QPSK and the signal-to-noise-ratio in the channel is 26.04 db.

Chapter 9

Conclusion

In this thesis, we have examined two associative neural networks: BAM and Kohonen map. For the BAM, the statistical properties of the first order case, as well as the general higher order case, have been examined. Also, four new encoding methods have been proposed to enhance the recall performance of the first order BAM. Besides, we have addressed the statistical storage behavior of the first order BAM under the forgetting learning. For the Kohonen map, we are mainly concerned with the utilization of its ordering properties for the data compression and transmission under noisy channel. Here are the highlights:

- For the first order BAM, we have examined its statistical properties from two different approaches: energy barrier and statistical dynamics.

  The approach of energy barrier is able to estimate the memory capacity and the number of errors in the retrieval pairs when a small number of errors are allowed in the retrieval pairs. However, the attraction basin for the worst case errors cannot be determined from this approach.

  Hence, we have presented the statistical dynamics about the confidence interval of the number of errors. Then we discuss a way to estimate the memory capacity, the attraction basin, and the number of errors in the retrieval pairs from the statistical dynamics. Although the statistical dynamics approach is not direct method, we can determine the attraction basin for the worst case errors from the statistical dynamics. Also, it can be used to analyze the associative memories without finding a suitable energy function.

For the second order BAM, we first use an example to illustrate that its state may converge to limit cycles. We have also derived the statistical dynamics for the second order BAM. Hence, we can estimate the memory capacity, the attraction basin, and the number of errors in the retrieval pairs. From the numerical and theoretical results, the ratio of the dimensions affects the properties of the second order BAM in a similar manner as it affects those of the first order BAM. Moreover, we have extended our results to the general higher order BAM.

There remains some interesting mathematical problems concerning the BAM which have not been resolved yet. They are, for example, the average number of spurious stable pairs, the distribution of the spurious stable pairs, how the ratio of the dimensions in the BAM affects the statistical dynamics of the average number of errors [66], and the better bound of the memory capacity and the attraction basin.

- The concepts of the four proposed encoding methods (HCA, EHCA, BL, and AHKBL) for the first order BAM have been described and shown to have significant improvements in the recall of the library pairs. The theoretical development underlying the encoding methods indicates that the memory capacity of HCA tends to $\min(n, p)$ as n and $p \to \infty$, and the BL is the optimal encoding method in terms of memory capacity. The EHCA is used to reduce the two connection matrices (found by the HCA) into one connection matrix and then the stable property of BAM can be surely maintained during recall. In view of the relatively weak error correction nature of BL, the AHKBL is developed to improve the error correction capability. Also, we have derived the convergent conditions of AHKBL.

  Besides, we have made a comparison among our four encoding methods and other existing approaches in different aspects: stability during recall, hardware implementation, information ratio, memory capacity, error correction capability and learning speed. This comparison gives us a guideline to choose a method which better meets out requirement.

Although the four proposed encoding algorithms can greatly improve the memory capacity and error correction capability, it is still an open problem in how to keep the number of the spurious stable pairs (which are the fixed points not in the set of the library pairs) as small as possible. Further work is to first understand the nature of the spurious stable pairs, and then to introduce a mechanism in the learning rules to remove the spurious stable pairs.

- We have theoretically estimated the number of most recent library pairs that can be stored as fixed points in the BAM under the forgetting learning. Also, we have discussed the way to choose the forgetting constant, such that the number of most recent library pairs being correctly stored is nearly maximal. Simulations have been carried out to verify our theoretical results.

  One interesting result is that the capacity of the forgetting learning is similar to that of the outer product rule. Moreover, the forgetting learning is an incremental learning rule with the ability of forgetting the old library pairs. Hence, the BAM under the forgetting learning can keep most recent information from the environment. Further goal is the study of its storage behavior when errors are allowed in the retrieval pairs.

- By utilizing the order property of Kohonen map, we have presented three new cross-relative applications of Kohonen map.

  1) We have proposed a new trellis type quantizer: TCKM. The design process of trellis in the TCKM is based on the neighborhood structure of Kohonen map instead of the Euclidean distances among the codevectors. From the observation that there is a similarity between the neighborhood structure of Kohonen map and the neighborhood structure of the channel waveforms in the TCM, we can use the existing trellis in the TCM as the trellis of the TCKM. Hence, the design process of TCKM is simpler than that of the conventional approach.

Simulation results show that the performance of our model is comparable to that of the conventional approach.

2) Also, we have introduced a new approach for the transmission of vector-quantized data under a noisy channel. The key point of this approach is that the neighborhood structure of Kohonen map should match the neighborhood structure of the channel waveforms. In other words, the new approach combines the source coder (vector quantizer) and modulation together. Under this new approach, the impulsive noise in the received data is greatly reduced. Moreover, the design process of this approach is also very simple since the association between the codevectors and the channel waveforms is based on their neighborhood structure.

3) Based on the similarity between the TCKM and TCM, we have introduced an error control scheme for transmission of vector-quantized data under noisy channel. To achieve the error control mechanism, identical trellis is used in the TCKM and TCM. To reduce the impulsive noise, the association between codevectors of TCKM and the channel waveforms of TCM is based on their neighborhood structures. In our approach, the source coder (vector quantizer), the error control, and the modulation are designed as a whole. Compared with the concept "combined source coder and modulation", the proposed error control scheme can further reduce the impulsive noise in the received vector-quantized data when the signal-to-noise ratio in the channel belongs to a suitable range.

The development of the above three applications is based on the observation that there exists a similarity between the neighborhood structure of Kohonen map and the neighborhood structure of the channel waveforms in the field of communications. We believe that more applications of Kohonen map in communications will be discovered in the near future based on this similarity.

Bibliography

[1] T. Kohonen, *Self-Organization and Associative Memory*, Berlin: Springer-Verlag, 1984.

[2] G. Palm, "On Associative Memory," *Biolog. Cybern.*, Vol. 36, pp. 19–31, 1980.

[3] T. Kohonen, "Correlation Matrix Memories," *IEEE Trans. Comput.*, Vol. 21, pp.353–359,1972.

[4] R. Rosenblatt, *Principles of Neurodynamics.* Spartan Books, New York, 1959.

[5] R. R. Hecht-Nielson, *Neurocomputing*, Addison-Wesley, 1990.

[6] P. K. Simpson, *Artificial Neural Systems: Foundations, Paradigms, Applications, and Implementations*, Pergamon Press, 1990.

[7] T. Kohonen, "The Neural Phonetic Typewriter,"*IEEE Computer*, March, pp. 11-22, 1988.

[8] C. Bouton and G. Pages, "Self-Organizing and a.s convergence of one-dimensional Kohonen algorithm with non-uniform distributed stimuli," *Stochastic Processes and their Application,* Vol. 47, 00. 249-274, 1993.

[9] M. Cottrell, J. C. Fort, and G. Pages, "Two or three things that we know about the Kohonen algorithm," *Technical report*, Samos University Paris, 1993.

[10] E. A. Ferran, " On Kohonen's ordering theorem for one-dimensional self-organized mappings," *Network*, Vol. 4, pp. 337-354, 1993.

[11] R. Togneri, E. Lai, and Y. Attikiouzel, "Kohonen's algorithm for the numerical parametrisation of manifolds," *Pattern Recognition Letters*, vol.11, no.5 pp313-319, 1990.

[12] J. J. Hopfield, "Neural Networks and Physical System with Emergent Collective Computation Abilities,*in Proc. Nat. Acad. Sci. U.S.*, Vol. 19, pp. 2253–2558, 1982.

[13] B. Kosko, "Bidirectional Associative Memories," *IEEE Trans. Syst. Man, and Cybern.*, Vol. 18, pp.49–60, 1988.

[14] J. Anderson, "A Simple Neural Network generating Interactive Memory," *Math. Biosci.*, Vol. 1, pp. 63–74, 1988.

[15] N. M. Nasrabadi, and R. A. King, "Image Coding Using Vector Quantization: A Review," *IEEE Trans. Commun.*, vol. 36, pp.957-971, 1988.

[16] R. M. Gray," Vector Quantization," *IEEE ASSP Mag.*, pp.4-29, April, 1984.

[17] Yoh–Han Pao. *Adaptive Pattern Recognition and Neural Network.* Addison Wesley, 1989.

[18] Rumelhart, McClelland. *Parallel Distributed Processing* vol. I, MIT press, 1986.

[19] Y. Linde, A. Buzo, and R. M. Gray, " An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, pp. 84–95, 1980.

[20] B. Bavarian, "Introduction to neural networks for intelligent control, " *IEEE Control Systems Magazine,* pp. 3-7, April, 1988.

[21] C. H. Wu, C. J. Wang, H. M. Tai, and D. A. Roland, " An investigation of high-order bidirectional associative memories: performance, applications, and parallel Implementations," *In Proc. 1990 IEEE International Symposium on Circuits and Systems*, pp.495-498.

[22] G. Mathai and B.R. Upadhyaya, " Performance analysis and application of the bidirectional associative memory to industrial spectral signatures," *In Proc. IJCNN'89*, pp. 33-37.

[23] B. H. Wang, and G. Vachtsevanos, "Storage Capacity of Bidirectional Associative Memories," in *Proc. IJCNN 91, Singapore*, pp. 1831–1836,1991.

[24] K. Haines, and R. Hecht–Nielsen, "A BAM with Increased Information Storage Capacity," in *Proc. of the 1988 IEEE Int. Conf. on Neural Networks*, pp.181–190,1988.

[25] C. S. Leung, L. W. Chan, and M. K. Lai,"Stability, Capacity, and Statistical Dynamics of Second Order BAM Bidirectional Associative Memory," to appear in *IEEE Trans. Syst. Man, and Cybern.*, Vol.25 No. 10, 1995.

[26] C. S. Leung, L. W. Chan, and M. K. Lai,"Stability and Statistical Properties of Second Order BAM Bidirectional Associative Memory," submitted to *IEEE Trans. Neural Networks.*

[27] C. S. Leung, "Forgetting Learning: Can the last k-th previous pattern be stored as a fixed point in Associative Memory ?", in *Proc. ICONIP'94-Seoul*, pp. 1086–1089, 1994.

[28] C. S. Leung, "Memory capacity and Statistical Dynamics of the First Order BAM Bidirectional Associative Memory," in preparation.

[29] C. S. Leung, "Trellis Coded Kohonen Map", in *Proc. ICONIP'94-Seoul*, pp. 955-959, 1994.

[30] C. S. Leung, "Design Trellis Coded Vector Quantizer using Kohonen Map", to be submitted.

[31] C. S. Leung, "Kohonen Map: Combined Vector Quantization and Modulation", in *Proc. ICONIP'94-Seoul*, pp. 242-247, 1994.

[32] C. S. Leung and K. F. Cheung," Householder Encoding for Discrete Bidirectional Associative Memory Associative Memory," in Proc. IJCNN 91 Singapore, Vol. 1, pp. 237–241,1991.

[33] C. S. Leung, "Encoding Method for Bidirectional Associative Memory using projection on convex sets," in *Proc. IJCNN 92 Beijing*, Vol. 2 pp. 81–85, 1992.

[34] C. S. Leung, "Encoding Method for Bidirectional Associative Memory using Projection on Convex Sets," *IEEE Trans. Neural Networks,* Vol. 4, September, pp.879–881, 1993.

[35] C. S. Leung, "Optimum Learning for Bidirectional Associative Memory in the Sense of Capacity," *IEEE Trans. Syst. Man, and Cybern.* , Vol 24, No. 5, pp.791-796, 1994.

[36] C. S. Leung, "Optimum Learning Rule in Bidirectional Associative Memory," in *Proc. of the 2th Pacific Rim International Conference on AI 1992*, Vol 2, pp.940–946, 1992.

[37] Andrew C. S. Leung and M. Klassen, "A Delta–Rule Encoding for Bidirectional Associative Memory," in *Proc. IJCNN 91 Seattle*, Vol. 2, pp. 954,1991.

[38] C. S. Leung, "Robust Learning Rule for Bidirectional Associative Memory," in *Proc. IJCNN 93 Nagoya*, Vol. 3 pp. 2686-2689, 1993.

[39] C. S. Leung, "The Performance of Dummy Augmentation Encoding," in *Proc. IJCNN 93 Nagoya*, Vol. 3 pp. 2674-2677, 1993.

[40] V. Srinivasan and C. S. Chia, "Improving Bidirectional Associative Memory Performance by Unlearning," *Proc. IJCNN 91, Singapore*, pp. 2472-2477.

[41] Y.F. Wang, J.B. Cruz, Jr., and J. H. Mulligan, Jr., "Two Coding Strategies for Bidirectional Associative Memory," *IEEE Trans. Neural Networks,* Vol. 1, pp.81–92, 1990.

[42] Y.F. Wang, J.B. Cruz, Jr., and J. H. Mulligan, Jr., "Guaranteed recall of all training pairs for Bidirectional Associative Memory," *IEEE Trans. Neural Networks,* Vol. 1, pp.559–567, 1991.

[43] T. Wang, X. Zhunag, and X. Xing, "Designing Bidirectional Associative Memories with Optimal Stability," *IEEE Trans. Syst. Man, and Cybern.* , Vol 24, No. 5, pp.778–790, 1994.

[44] K. Shanmukh, and Y. V. Venkatesh," On an Optimal Learning Scheme for Bidirectional Associative Memories", *Proc. IJCNN 93 Nagoya*, Vol. 3 pp. 2670-2673, 1993.

[45] P. K. Simpson, "Higher-Ordered and Intraconnected Bidirectional Associative Memories", *IEEE Trans. System, Man, and Cybern.,* Vol. 20, pp. 637–653, 1990.

[46] H. M. Tai, C. H. Wu, and T. L. Jong,"High-Order Bidirectional Associative Memory," *Electronics Letters*, Vol. 25, pp. 1424-1423, 1989.

[47] Y. J. Jeng and C. C. Yeh, " Modified Intraconnected Bidirectional Associative Memory," *Electronics Letters*, Vol.27, no.20 p1818-1819, 1991.

[48] Seong-Sik Min and Soo-Young Lee,"Multi-layer Bidirectional Associative Memory," in *Proc. ICFLNN-90*, pp.251–254. 1990.

[49] Seong-Sik Min and Soo-Young Lee," Supervised Learning with Multi-layer Bidirectional Associative Memory," in *Proc. ICANN-91*, 1991.

[50] B. L. Zhang, B. Z. Xu, and C. P. Kwong,"Performance Analysis of Bidirectional Associative Memory and an Improved Model from the Matched-Filtering Viewpoint," *IEEE Trans. Neural Networks,* Vol. 4, September, pp.864–872, 1993.

[51] G. Q. Hu, C. P. Kwong, and Z. B. Xu, "Two Iterative Encoding Schemes for Bidirectional Associative Memory," In *Proceeding ISSIPNN'94*, pp. 93–96.

[52] H. Oh and S. C. Kothari, "A Pseudo-relaxation Learning Algorithm for Bidirectional Associative Memory," in *Proc. IJCNN-Baltimore-92*, Vol. 2, pp.208-213, 1992.

[53] Y. J. Jeng and C. C. Yeh, " Exponential Bidirectional Associative Memories," *Electronics Letters*, Vol.26, no.11 pp.717-718,1990.

[54] Mohamad H. Hassoun, " Dynamic Heteroassociative Neural Memories,",*Neural Networks,* VOL 2., pp.275-289,1989.

[55] B. Maundy and E.l. Masry, " A Switched Capacitor Bidirectional Associative Memory," *IEEE Trans. Circuits and Systems*, Vol.37, pp. 1568-72,1990.

[56] Demetri Psaltis, Cheol Hoon Park, and John Hong," Higher Order Associative Memories and Their Optical Implementations," *Neural Networks,* Vol. 1, pp. 149–163, 1988.

[57] J. M. Kinser, H. J. Caulfield, and J. Shamir, " Design for a massive all-optical Bidirectional Associative: the big BAM," *Applied Optics,* Vol.27, no.16 pp.3442-4,1988.

[58] H. Lui, "Perceptron learning on Hopfield nets," *Neural Networks Supplement:INNS Abstract*, Vol. 1, pp. 198, 1988.

[59] Charles M. Newman, "Memory Capacity in Neural Models: Rigorous Lower Bounds,", *Neural Networks*, Vol. 1, pp. 223–238, 1988.

[60] Janos Komlos and Ramamohan Paturi, "Convergence Results in an Associative Memory Model," *Neural Networks*, Vol. 1, pp. 229–250, 1988.

[61] R. J. McEliece, E. C. Posner, E. R. Rodemich, and S. S. Venkatesh, "The Capacity of the Hopfield Associative Memory," *IEEE Trans. Inform. Theory*, Vol. 33, pp.461–482,1987.

[62] S. S. Venkatesh, and D. Psaltis, " Linear and Logarithmic Capacities in Associative Neural Networks," *IEEE Trans. Inform. Theory*, Vol. 35, pp.558–568, 1989.

[63] A. Dembo,"On the Capacity of Associative Memories with Linear Threshold Functions," *IEEE Trans. Inform. Theory*, Vol. 35, pp.709–720,1989.

[64] S. Amari,"Mathematical Foundations of Neurocomputing," *Proceedings of IEEE*, Vol. 78, pp.1443–1463, 1990.

[65] S. Amari and K. Maginu, "Statistical Neurodynamics of Associative Memory, " *Neural Networks*, Vol. 1, pp. 63–73, 1988.

[66] S. Amari, " Statistical Neurodynamics of various versions of correlation associative memory," *In Proc. IEEE ICNN'1988*, pp.633–640.

[67] H. Yanai and Y. Sawada, " Associative Memory Network Composed of Neurons with Hysteretic Property," *Neural Networks*, Vol. 3, pp. 223–228, 1990.

[68] S. S. Venkatesh and P. Bald," Programmed Interactions in Higher-order Neural Networks: The Outer-Product Algorithm," *Journal of Complexity*, Vol 7, pp.443–479, 1991.

[69] Althanasios Papoulis, *Probability, Random Variables, and Stochastic Process*, McGraw-Hill, 1985.

[70] A. O. Steinhardt, "Householder Transforms in Signal Processing," *IEEE ASSP Magazine*, July, pp.4–12,1988.

[71] K. E. Atkinson, *An Introduction to Numerical Analysis*, Wiley, 1989.

[72] D. C. Youla and H. Webb."Image restoration by the method of convex projection: Part 1– Theory." *IEEE Trans. Med Imaging*. Vol MI–1. pp. 81–94, 1982.

[73] H. Stark, *Image Recovery*. New York:Academic 1987.

[74] Mohamad H. Hassoun, and Jing Song, "Adaptive Ho-Kashyap Rules for Perceptron Training," *IEEE Trans. Neural Networks,*, VOL 3., pp.51–61,1992.

[75] S. Agmon, "The Relaxation Method for Linear Inequalities," *Canadian J. Math.,* VOL. 6, No. 3, 1954.

[76] G. Yen, and A. N. Michel, "A Learning and Forgetting Algorithm in Associative Memories: results involving pseudo-inverse," *IEEE Trans. on Circuits and Systems,* VOL 38, pp. 1193–1205,1991.

[77] A. Bruce Carlson, *Communication Systems*, McGraw-Hill, 1986.

[78] K. Fujimura, H. Tokutaka, and Y. Ohshima, " The Traveling Salesman Problem Applied to the Self-Organizing Feature Map,", accepted for publication in *Proc. ICONIP'94*, Seoul.

[79] M. W. Marcellin and T. R. Fischer, "Trellis coded quantization of memoryless and Gauss-Markov sources," *IEEE Trans. Commun.*, vol. 38, pp. 82-93, 1990.

[80] E. Biglieri, D.Divsalar, P. J. Mclane, and Marvin K. Simon, *Introduction to trellis-coded modulation with application,* Maxwell Macmillan, 1991.

[81] H. S. Wang and N. Moayeri, " Trellis coded vector quantization," *IEEE Trans. Commun.*, vol. 40, pp. 1273–1276, 1992.

[82] Herbert Edelsbrunner, *Algorithms in Combinatorial Geometry*, Berlin: Springer-Verlag, 1987.

[83] D. Avis, and B. K. Bhattacharya, "Algorithms for Computing d-Dimensional Voronoi Diagram and their Duals", *Advances in Computing Researches,*, Vol. 1, pp. 159-180, JAI Press Inc., 1983.

[84] N. Farvardin, "A study of Vector Quantization for Noisy Channel," *IEEE Trans. Information Theory,* Vol. 36, pp. 799–809, 1990.

[85] V. A. Vaishampayan and N. Farvardin, "Joint Design of Block Source Codeds and Modulation Signal Sets," *IEEE Trans. Information Theory*, Vol. 38, pp. 1230–1248, 1992.

[86] , H. Skinnemoen, " Modulation Organized Vector Quantization," in *Proc ISIT'94*, pp. 238, 1994.

[87] , T.R. Fischer,"Joint trellis coded quantization/modulation," *IEEE Trans. Comm.*, Vol. 39, pp.172–176, 1991.