Adaptive Associative Memories Capable of Pattern Segmentation

Qing Ma

Abstract— **This paper presents an adaptive type of associative memory, AAM, that can separate patterns from composite inputs which might be degraded by deficiency or noise, and can recover incomplete or noisy single patterns. We analyze the behavior of AAM in terms of stability, giving the stable solutions (results of recall) and show that the recall of spurious memories (the undesired solutions) can be greatly reduced compaired with earlier types of associative memory that can perform pattern segmentation. We also give two conditions that guarantee the non-existence of undesired solutions. Computer experiments show that the performance of AAM is much better than that of the earlier types of associative memory in terms of pattern segmentation and pattern recovery.**

Keywords— **Associative memory, Adaptive, Pattern segmentation, Pattern recovery, One-to-many associative recall, Stability, Stable solution, Storage capacity, Fast recall.**

I. INTRODUCTION

Associative memories [1]-[16] proposed so far form a promising paradigm in the field of artificial neural networks in which the information is stored and recalled in much the same way as in a human brain. These memories can be classified according to various features such as the network organization (dynamic vs. static) and the nature of the stored associations (autoassociative vs. heteroassociative)[8][9]. They can also be classified according to whether or not they are capable of pattern segmentation in addition to their common attribute of recalling one stored pattern from its incomplete or noisy input.

Pattern segmentation in associative memories is essential for realizing one-to-many associative recall, a basic and widely-used attribute of human associative processing [13], and, for generalizing the meaning of complex patterns to decompose them into functional components for use in subsequent integration [14]. Pattern segmentation cannot be performed by conventional associative memories [1]-[11] which treat composite pattern essentially as a synthetic whole (i.e., all pairs of features are glued together) [14][17][18], but it can be achieved in the competitive network proposed by Hirai [12][13] and the oscillatory networks first proposed by Wang et al. [14][15][16]. By combining a competitive network with Kohonen's heteroassociative memory [1][3], Hirai further proposed an associative memory, HASP [13], that is capable of one-to-many association. It has been shown that when an associative memory is equipped with a one-to-many associative function, it becomes a powerful model that can be used to represent various cognitive processes [19]-[21].

The author is with Kansai Advanced Research Center, Communications Research Laboratory, Ministry of Posts and Telecommunications, Kobe, Japan. E-mail: qma@crl.go.jp

The oscillatory networks are constructed with oscillator instead of the simple units used in conventional associative memories. As Wang et al. [14] and Horn et al. [15] noted, this type of network works well only in the case where a small number of patterns is involved in the segmentation task, because only a few states can be represented in response to a input. It is also difficult to analyze their behavior because of the complexity of its mechanism. The structure of HASP's competitive network is the same as that of Hopfield associative memory, HAM [4][5], except that the units are linked by inhibitory connections. Pattern segmentation is performed by competition among patterns via these inhibitory connections. However, HASP also has some shortcomings. One is that spurious memories are easily recalled in pattern segmentation, even if there is little overlap in the input patterns or the number of superposed patterns is quite small. The other is that patten segmentation is very slow when the degree of pattern overlap is high or the number of input patterns is large, because HASP requires a large time constant for stability.

This paper proposes adaptive associative memory, AAM, which resolves the problems of HASP mentioned above. In addition, the reduction of spurious memories can substantially improve the pattern recovery capability, a fundamental attribute of associative memories. In general, there are two ways of resolving or improving the problem of spurious memories. One is to find a better learning algorithm, so that spurious memories are less likely to occur. The other is to develop a better recall method in which undesired stable states (i.e., spurious memories) are avoided as much as possible. AAM can be regarded as an enhanced version of the HASP's competitive network; it has the same learning algorithm with improved recall performance achieved by introducing adaptive inter-neuron connections. As shown later, the adaptive connections make it possible to switch the competing style, in which the pattern is separated, from a process involving the comparison of perseverance among patterns to a "first come, first served" process. This approach not only reduces the recall of spurious memories, but also allows stability to be guaranteed in the case that the time constant is set to 0. As a result, patterns can be separated much faster.

In this paper, we first describe the AAM architecture, and then analyze its behavior in terms of stability, giving the stable solutions. We further show that the recall of spurious memories (undesired stable solutions) can be greatly reduced compaired with HASP and we give two conditions that guarantee the absence of undesired solutions. In extensive computer simulations, we also show that the performance of AAM is much better than that of HASP at

both pattern segmentation and pattern recovery.

II. ADAPTIVE ASSOCIATIVE MEMORY (AAM)

An AAM unit is shown in Fig. 1. It consists of four neurons (E_i, S_i, A_i, I_i), three connections (W_{ii}, W_{ji}, W'_{ji}), and a readout control unit (RCU_i). RCU_i is the same as that used in HASP [12][13] and will be described at the end of this section. E_i is a biased input neuron, S_i is a principal neuron competing with S_j via I_i and I_j for each $j \neq i$, whose state directly represents that of the whole unit, I_i is an inhibitory relay neuron that receives excitatory and inhibitory inputs respectively from S_i and RCU_i and gives inhibition to S_j for each $j \neq i$. A_i is an activity-sum neuron for gathering the activity of S_j for each $j \neq i$. W_{ii} is an adaptive excitatory auto-connection, whose weight is adaptively modified according to the activity of neuron A_i. W_{ji} is a learnable inhibitory connection, whose weight is decided at the learning phase. W'_{ji}, a mirror image of W_{ji}, is an excitatory connection, whose weight is equal to W_{ji} for the coordination.

AAM consists of a fully-interconnected set of such units. It operates synchronously and handles only binary (0-1) patterns. The dynamics of AAM are expressed as follows:

$$S_i(k+1) = \varphi(W_{ii}(k)S_i(k) - \sum_{j \neq i} W_{ji}I_j(k) + E_i), \quad (1)$$

$$I_i(k) = \varphi(S_i(k) - W_{RI}B_i(k)), \quad (2)$$

$$E_i = c_iP_i + b, \quad (3)$$

$$W_{ii}(k) = \alpha A_i(k), \quad (4)$$

$$A_i(k) = \sum_{j \neq i} W'_{ji}\psi(S_j(k)), \quad (5)$$

$$W'_{ji} = W_{ji} \quad (j \neq i), \quad (6)$$

$$S_i(0) = 0, \quad (7)$$

where $S_i(k)$ in (1) denote the activity of neuron S_i (and unit i), $B_i(k)$ and W_{RI} in (2) are the output of RCU_i, which is defined later ((10)-(12)), and the gain of RCU_i, respectively. P_i in (3) is the external input (0 or 1) and c_i and b in (3) are the connecting coefficient from P_i to neuron E_i and the bias of neuron E_i, respectively. α in (4) is a positive coefficient called adaptive weight. The functions $\varphi(\cdot)$ and $\psi(\cdot)$ perform analog and binary thresholding as follows:

$$\varphi(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \quad (8)$$

$$\psi(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \quad (9)$$

As shown in Fig. 1 and equations (4)-(6), the auto-connection weight, W_{ii}, changes adaptively according to the activities of all the other neurons S_j ($j \neq i$) that are weighted with the corresponding inhibitory connections (note that $W_{ji} = W'_{ji}$). On the other hand, the weight of inhibitory connections, W_{ji}, is decided at the learning phase according to the following learning algorithm (which is the same as that of HASP):

(i) Initially, $W_{ji} = W$ for all i and j where $W > 0$
(ii) When a learning pattern is presented,

$$W_{ji} = \begin{cases} 0 & \text{if } E_i \cdot E_j > 0 \\ W_{ji} & \text{if } E_i \cdot E_j = 0 \end{cases}$$

From this learning algorithm, we can see that inhibitory connections are symmetric, i.e., $W_{ij} = W_{ji}$. After the learning phase, the autocorrelations of learned patterns are stored as spatial distributed patterns of reduced inhibitions. As analyzed in the next sections, if one of these stored patterns is input, the pattern itself will appear as a stable state (output) because of the reduced inhibition, whereas if a composite and/or noisy pattern is input, the patterns constituting the input will compete for survival via the intact inhibitory connections until a single stored pattern finally wins the competition (i.e., is recalled).

To recall all the patterns constituting the input, it is necessary to have a readout controller that selects them one by one; this is performed by the readout control unit, RCU_i. As shown in Fig. 1 and equation (2), if the output of RCU_i turns on, $I_i(k)$ is suppressed. That is, the competitive power of the currently active pattern can be weakened. As a result, the other pattern get a chance to become active. The structure of RCU_i is shown in Fig. 2. It is controlled by an external supplied readout control signal (rcs), and consists of three elements, G_i, D_i and B_i. G_i receives inputs from S_i and rcs and performs a gating function. D_i memorizes the state of neuron S_i in a positive feedback loop to avoid the recalling of the same pattern again. B_i receives inputs from G_i and D_i and suppresses the I_i neuron. The dynamics are defined as follows:

$$B_i(k) = \psi(G_i(k) + D_i(k)), \quad (10)$$

$$D_i(k) = \psi(G_i(k) + D_i(k)), \quad (11)$$

$$G_i(k) = rcs(k) \cdot \psi(rcs^*(k) \cdot S_i(k) + G_i(k)), \quad (12)$$

where $rcs^*(k)$ is a positive signal that turns on for a brief period when $rcs(k)$ is on (see [12] for detail).

The main differences between AAM and HASP are (i)the time constant of AAM is set to 0, whereas HASP has a non-zero time constant; (ii)the adaptive weight α in (4) is set to 0 in HASP.

III. STABILITY OF AAM

In this section, we examine the stability of AAM and what kinds of pattern can be recalled (i.e., what are the stable solutions) when a single stored pattern or a composite pattern is input. Here, we consider the case where there is no output from RCU_i (i.e., $B_i(t) = 0$), no bias for input (i.e., $b = 0$), and the adaptive weight is set to 1 (i.e., $\alpha = 1$). To avoid a state that all units having a same value and the competition therefore cannot be triggered, c_i of (3) is set randomly to a value between 1 and $1+\delta$ ($0 < \delta \ll 1$). According to whether units i and j are linked by inhibitory connections each other, we can set $W_{ji} = W$ or 0 (see ($L-1$)). Thus, by substituting (2)-(6) and (9)

into (1), we obtain the following dynamic equation for unit i:

$$S_i(k+1) = \varphi(Wn_i(k)S_i(k) - W\sum_{j=1}^{n_i(k)} S_{j^*}(k) + (1+\delta)P_i), \quad (13)$$

where $S_i(k)$ and $S_i(k+1)$ are the activities of unit i at time k and $k+1$, respectively, $S_{j^*}(k)$ denotes the activity of unit j^* that is active and has connection with unit i at time k, and $n_i(k)$ denotes the number of such active units j^* at time k.

Theorem 1: When a single stored pattern is input, the input itself appears as a stable pattern.

Proof: Since $S_i(0) = 0$ for each i by (7), from (13) we obtain $S_i(1) = (1+\delta)P_i$, where $P_i = 1$ or 0. Since none of the stored patterns contain connections among 1-bits as a result of the learning $(L-1)$, $n_i(1)$ of (13) is 0 for each i such that $S_i(1) = (1+\delta)P_i = 1+\delta$. Thus, from (13) we obtain $S_i(2) = (1+\delta)P_i = 1+\delta$ for each i such that $S_i(1) = 1+\delta$, and $S_j(2) = 0$ for each j such that $S_j(1) = (1+\delta)P_j = 0$. The input itself therefore appears as a stable pattern. ◊

Next, let us analyze the behavior of AAM in terms of stability when a composite pattern is input. Here we only consider the case where the input consists of two stored patterns. However, the analysis of the two-pattern case can be extended to n patterns ($n \geq 2$) by treating $n-1$ patterns as one pattern and the remaining one as the other pattern. When a composite pattern of A and B is input, a general competing state at time k ($k \geq 1$) can be represented by a diagram in Fig. 3. In this figure, $P^{(A)}$ and $P^{(B)}$ express the sets of units of AAM corresponding to the 1-bit (i.e., bit i such that $P_i = 1$) of patterns A and B, respectively, and Ω expresses the set of common units between $P^{(A)}$ and $P^{(B)}$. Later, we use denotations, A and B, to express the sets of units in $P^{(A)} - \Omega$ and $P^{(B)} - \Omega$, respectively. A and B are further divided into three subsets, respectively (In the figure, time variable k is omitted): $H^{(A)}(k)$ and $H^{(B)}(k)$ express the groups of active units that have no connections with any other units active at time k, $C^{(A)}(k)$ and $C^{(B)}(k)$ express the groups of active units that have connections with other units active at time k (so, any unit of $C^{(A)}(k)$ ($C^{(B)}(k)$) must have connections with $C^{(B)}(k)$ ($C^{(A)}(k)$), respectively, by the symmetry of connections), and the remaining areas, $R^{(A)}(k)$ and $R^{(B)}(k)$, represent the sets of inactive units. Later, for convenience we use two notations X and Y such that X denoting either A or B, i.e., $X = A$ or B, and (X, Y) denoting either (A, B) or (B, A), i.e., $(X, Y) = (A, B)$ or (B, A).

Since $S_i(0) = 0$ for each i by (7), from (13) we obtain $S_i(1) = (1+\delta)P_i = 1+\delta$ for the $i \in P^{(A)} \bigcup P^{(B)}$. So, as the initial competing state after a pattern is inputed, we have $R^{(X)}(1) = \phi$ and $C^{(X)}(1) \neq \phi$, where ϕ denotes empty set. It should be noted that $C^{(A)}(k)$ and $C^{(B)}(k)$ must coexist by the symmetry of connections. Using this general diagram, we will show that $C^{(X)}(k)$ and all the other sets that are variable with time will reduce with time k increasing and finally only the stable sets remain.

For unit $i \in \Omega$, since it has no connections with any units of $P^{(A)}$ and $P^{(B)}$ as a result of learning $(L-1)$, $n_i(k)$ of (13) is 0 at any time k. Thus, from (13) we obtain $S_i(k+1) = 1+\delta$. So, Ω is stable.

For unit $i \in C^{(X)}(k)$, since $S_i^{(X)}(k) > 0$ and $n_i(k) \neq 0$, from (13) we have

$$S_i(k+1) = \varphi(Wn_i(k)S_i^{(C^{(X)}(k))}(k) - W\sum_{j=1}^{n_i(k)} S_j^{(C^{(Y)}(k))}(k) + (1+\delta))$$

$$= Wn_i(k)\varphi(S_i^{(C^{(X)}(k))}(k) - \bar{S}_i^{(C^{(Y)}(k))}(k) + \frac{1+\delta}{Wn_i(k)}), \quad (14)$$

where $\bar{S}_i^{(C^{(Y)}(k))}(k)$ is the average activity of units that have connections with unit i at time k. Supposing $S_i(k) \geq 1+\delta$ for $i \in C^{(X)}(k) \bigcup C^{(Y)}(k)$, we have $\bar{S}_i^{(C^{(Y)}(k))}(k) \geq 1+\delta$. By setting $W \gg 1$, the 3rd term of (14) is much smaller than the other two terms. Thus, this term can be ignored. So, whether $S_i(k+1)$ is larger than or equates to 0 can be judged as follows:

$$Sign\{S_i(k+1)\} = \psi'(S_i^{(C^{(X)}(k))}(k) - \bar{S}_i^{(C^{(Y)}(k))}(k)), \quad (15)$$

where $Sign\{x\} = 1$ means $x > 0$, $Sign\{x\} = 0$ means $x = 0$, and $\psi'(\cdot)$ is defined as follows:

$$\psi'(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (16)$$

From (14) and (15), we know $S_i(k+1) \geq 1+\delta$ if $S_i(k+1) > 0$. Since $S_i(k+1) > 0$ for $i \in C^{(X)}(k+1)$, we obtain $S_i^{(C^{(X)}(k+1))}(k+1) \geq 1+\delta$. That is, if the activities of the units of $C^{(X)}(k)$ are not smaller than $1+\delta$, the activities of the units of $C^{(X)}(k+1)$ must also be not smaller than $1+\delta$. Thus, if (15) can be used at time k, it can also be used at time $k+1$. By considering that $S_i(1) = (1+\delta)P_i = 1+\delta$ for the $i \in P^{(A)} \bigcup P^{(B)}$, we obtain a conclusion that at any time k ($k \geq 1$), $S_i^{(C^{(X)}(k))}(k) \geq 1+\delta$ for $i \in C^{(X)}(k)$ and (15) can be used.

Theorem 2: It is impossible that both the following two relations:

$$S_i^{(C^{(X)(k)})}(k) \geq \bar{S}_i^{(C^{(Y)(k)})}(k) \text{ and } S_j^{(C^{(Y)(k)})}(k) \geq \bar{S}_j^{(C^{(X)(k)})}(k)$$

or the following two relations:

$$S_i^{(C^{(X)(k)})}(k) < \bar{S}_i^{(C^{(Y)(k)})}(k) \text{ and } S_j^{(C^{(Y)(k)})}(k) < \bar{S}_j^{(C^{(X)(k)})}(k)$$

hold for each $i \in C^{(X)}(k)$ and $j \in C^{(Y)}(k)$ at any time k.

Firstly, since the δ of (13) is random number, it is impossible that both the following two relations: $S_i^{(C^{(X)(k)})}(k) = \bar{S}_i^{(C^{(Y)(k)})}(k)$ and $S_j^{(C^{(Y)(k)})}(k) = \bar{S}_j^{(C^{(X)(k)})}(k)$ hold for each $i \in C^{(X)}(k)$ and $j \in C^{(Y)}(k)$, which means that all the units have a same value. Thus, excluding the previous two relations, the remaining relations of $(T2-1)$ and $(T2-2)$ can be easily proved by giving a converse proposition and the detail proof is omitted.

From $(T2-1)$ of *Theorem 2* and (15), we can know that there must be some relative weak units that become inactive at time $k+1$. That is, there must be some relative weak units in $C^{(X)}(k)$ that become the new members of $R^{(X)}(k+1)$. Further, if there are units in $C^{(X)}(k)$ that only have connections with these weak units, these active units become the new members of $H^{(X)}(k+1)$.

$R^{(X)}(k)$ is divided into two parts: $R^{(X)}_{CH}(k)$ and $R^{(X)}_{R}(k)$. The units of $R^{(X)}_{CH}(k)$ have no connections with the units active at time k, whereas the units of $R^{(X)}_{R}(k)$ are connected with the units active at time k.

(i)For unit $i \in R^{(X)}_{CH}(k)$, since $n_i(k)=0$, from (13) we have $S_i(k+1)=1+\delta>0$. That is, the unit i is reactivated at time $k+1$. So, the unit i belongs to $C^{(X)}(k+1)$ or $H^{(X)}(k+1)$ at time $k+1$. Thus, $R^{(X)}_{CH}(k)$ can be further divided into two subsets denoted by $R^{(X)}_{C}(k)$ and $R^{(X)}_{H}(k)$, respectively. $R^{(X)}_{C}(k)$ is used to express the group of units that become the members of $C^{(X)}(k+1)$ at time $k+1$. $R^{(X)}_{H}(k)$ is used to express the group of units that become the members of $H^{(X)}(k+1)$ at time $k+1$.

(ii)For unit $i \in R^{(X)}_{R}(k)$, since $S_i(k)=0$ and $n_i(k)\neq 0$, from (13) we have

$$S_i(k+1)=\varphi(Wn_i(k)\cdot 0-W\sum_{j=1}^{n_i(k)} S_j^{(C^{(Y)}(k)+H^{(Y)}(k))}(k)+(1+\delta))$$

$$=\varphi(-W\sum_{j=1}^{n_i(k)} S_j^{(C^{(Y)}(k)+H^{(Y)}(k))}(k)+(1+\delta)) \qquad (17)$$

Since $S_j^{(C^{(Y)}(k))}(k) \geq 1+\delta$ as described above and $S_j^{(H^{(Y)}(k))}(k)=1+\delta$, which will be described later, from (17) we obtain $S_i(k+1)=0$, if $W>1$. So, the unit i of $R^{(X)}_{R}(k)$ remains in $R^{(X)}(k+1)$. Thus, $R^{(X)}_{R}(k)$ can be further divided into three subsets denoted by $R^{(X)}_{R_C}(k)$, $R^{(X)}_{R_H}(k)$, and $R^{(X)}_{R_R}(k)$, respectively. $R^{(X)}_{R_C}(k)$ is used to express the group of units that become the members of $R^{(X)}_{C}(k+1)$ at time $k+1$, $R^{(X)}_{R_H}(k)$ is used to express the group of units that become the members of $R^{(X)}_{H}(k+1)$ at time $k+1$, and $R^{(X)}_{R_R}(k)$ is used to express the group of units that remain in the same set, $R^{(X)}_{R_R}(k+1)$, at time $k+1$.

For unit $i \in H^{(X)}(k)$, since $n_i(k)=0$, from (13) we have $S_i(k+1)=1+\delta>0$. As described in the previous paragraph, we know that only the units of $R^{(X)}_{CH}(k)$ can be reactivated at time $k+1$. Since the units of $R^{(X)}_{CH}(k)$ have no connections with the units active at time k, $n_i(k+1)$ cannot increase, i.e., $n_i(k+1)=n_i(k)=0$. So, the unit i of $H^{(X)}(k)$ remains in the same set at time $k+1$.

Here, we define a relation $U \Rightarrow V$ to express that set U is transformed into set V at next time.

Corollary 1: (a)Ω is stable.

(b) part of $C^{(X)}(k) \Rightarrow R^{(X)}(k+1)$ and/or part of $C^{(X)}(k) \Rightarrow H^{(X)}(k+1)$

(c)

$$R^{(X)}_{R_C}(k) \Rightarrow R^{(X)}_{C}(k+1) \text{ and } R^{(X)}_{C}(k) \Rightarrow C^{(X)}(k+1),$$

$$R^{(X)}_{R_H}(k) \Rightarrow R^{(X)}_{H}(k+1) \text{ and } R^{(X)}_{H}(k) \Rightarrow H^{(X)}(k+1),$$

$$R^{(X)}_{R_R}(k) \Rightarrow R^{(X)}_{R_R}(k+1),$$

where $R^{(X)}(k) = R^{(X)}_{CH}(k) \bigcup R^{(X)}_{R}(k)$, $R^{(X)}_{CH}(k) = R^{(X)}_{C}(k) \bigcup R^{(X)}_{H}(k)$, and $R^{(X)}_{R}(k) = R^{(X)}_{R_R}(k) \bigcup R^{(X)}_{R_C}(k) \bigcup R^{(X)}_{R_H}(k)$. The units of $R^{(X)}_{CH}(k)$ have no connections with the units active at time k, whereas the units of $R^{(X)}_{R}(k)$ have connections with the units active at time k.

(d)$H^{(X)}(k) \Rightarrow H^{(X)}(k+1)$ with activity $1+\delta$ at any time k.

By *Corollary 1*, we know that if there is no $R^{(X)}_{C}(k)$ at each time k, $C^{(X)}(k)$ must reduce with time k increasing and finally only $H^{(X)}(k)$ and $R^{(X)}_{R_R}(k)$ remain, which is a stable state. So, the behavior of $R^{(X)}_{C}(k)$ is the key point to see the stability of AAM. If $R^{(X)}_{C}(k)$ exists, we know $R^{(X)}_{C}(k) \Rightarrow C^{(X)}(k+1)$ by *Corollary 1*. Here, however, we make (define) a new set denoted by $\Gamma^{(X)}_{new}(k+1)$ at time $k+1$ whenever $R^{(X)}_{C}(k)$ exists at time k and regard that the reactivated units are accommodated in the new set.

Definition of Γ *set:* Whenever $R^{(X)}_{C}(k)$ exists at time k, a new Γ set is formed (defined) at time $k+1$ so that

$$R^{(X)}_{C}(k) \Rightarrow \Gamma^{(X)}_{new}(k+1) \text{ and } R^{(X)}_{C}(k)=\Gamma^{(X)}_{new}(k+1) \quad (18)$$

Definition of $\tilde{C}$ *set:* Supposing there are h Γ sets at time k, $\tilde{C}^{(X)}(k)$ is defined as

$$\tilde{C}^{(X)}(k)=C^{(X)}(k)-\bigcup_{l=1}^{h}\Gamma^{(X)}_{l}(k) \qquad (19)$$

From the two definitions, we can easily obtain following properties.

Properties of Γ *and* $\tilde{C}$ *set:* (i)Once a Γ set is formed at time k, no unit of R_C can be added to this Γ set after k. (ii)No unit of R_C can be added to $\tilde{C}$ set all the time. (iii)Once a Γ set is formed, we can regard that it (and $\tilde{C}$ set) will remain all the time, by introducing the concept of empty set.

Theorem 3: (a)When a new Γ set within set X is formed, a corresponding new Γ set within set Y is formed at the same time, i.e., $\Gamma^{(X)}_{new}(k)$ and $\Gamma^{(Y)}_{new}(k)$ are formed at time k as a pair; (b)there are connections between this two Γ sets.

Proof: (a)Supposing $\Gamma^{(Y)}_{new}(k)$ is not formed when $\Gamma^{(X)}_{new}(k)$ is formed, we know $C^{(Y)}(k) \subseteq C^{(Y)}(k-1)$ by *Corollary 1*. Since $\Gamma^{(X)}_{new}(k) \subseteq C^{(X)}(k)$, $\Gamma^{(X)}_{new}(k)$ has connections with $C^{(Y)}(k)$. $\Gamma^{(X)}_{new}(k)$ therefore has connections with $C^{(Y)}(k-1)$. This means $R^{(X)}_{C}(k-1)$ has connections with $C^{(Y)}(k-1)$, which is contradictory to the difinition of R_C set. So, $\Gamma^{(Y)}_{new}(k)$ is formed when $\Gamma^{(X)}_{new}(k)$ is formed.

(b)Supposing $\Gamma^{(X)}_{new}(k)$ has no connections with $\Gamma^{(Y)}_{new}(k)$, it has connections with $C^{(Y)}(k-1)$ (note $C^{(Y)}(k) \subseteq C^{(Y)}(k-1) + \Gamma^{(Y)}_{new}(k)$). This means $R^{(X)}_C(k-1)$ having connections with $C^{(Y)}(k-1)$, which is contradictory to the difinition of R_C set. So, there are connections between this two Γ sets. $\diamondsuit$

Definition of Γ*-unit's transformation:* A unit i of $\Gamma^{(X)}_p(k)$ is regarded to be transformed into $\Gamma^{(X)}_q(k+1)$ $(p \neq q)$ at time $k+1$ if the i satisfies the following conditions: (1)it is active at time $k+1$; (2)it has no connections with $\Gamma^{(Y)}_p(k+1)$; (3)it has connections with $\Gamma^{(Y)}_q(k+1)$. Here, $\Gamma^{(X)}_p(k)$ (and $\Gamma^{(X)}_q(k+1)$) is Γ set or $\tilde{C}$ set.

Theorem 4: If there are no connections between any two Γ sets, $\Gamma^{(X)}_p(k)$ and $\Gamma^{(Y)}_q(k)$ for $p \neq q$, and between any Γ set and $\tilde{C}$ set, $\Gamma^{(X)}_p(k)$ and $\tilde{C}^{(Y)}(k)$, supposing $\Gamma^{(X)}_1(k)$ is a Γ set or $\tilde{C}$ set and $\Gamma^{(X)}_O(k)$ is the sum of all other Γ sets and $\tilde{C}$ set within the set X, any unit of $\Gamma^{(X)}_O(k)$ cannot be transformed into $\Gamma^{(X)}_1(k+1)$.

Proof: Here, we consider a unit i of $\Gamma^{(X)}_O(k)$. If the i is inactive at time $k+1$, it is transformed into $R^{(X)}(k+1)$. If the i is active and has connections with $\Gamma^{(Y)}_O(k+1)$ at time $k+1$, it remains in $\Gamma^{(X)}_O(k+1)$. If the i is active and has no connections with $\Gamma^{(Y)}_O(k+1)$, this means any $j \in \Gamma^{(Y)}_O(k)$ that connected with the i at time k becomes inactive at time $k+1$ and therefore cannot be transformed into $\Gamma^{(Y)}_1(k+1)$ at time $k+1$. By considering that the i has no connections with $\Gamma^{(Y)}_1(k)$ at k and $\Gamma^{(Y)}_1(k)$ cannot obtain new members from other sets ($R^{(Y)}_C(k)$) at time $k+1$, we know that the i has no connections with $\Gamma^{(Y)}_1(k+1)$ at time $k+1$. That is, the i is transformed into $H^{(X)}(k+1)$. So, the i in any case cannot be transformed into $\Gamma^{(X)}_1(k+1)$ at time $k+1$.

Theorem 5: Supposing there are h $(h \geq 1)$ pair of Γ sets, $\Gamma^{(X)}_l(k)$ and $\Gamma^{(Y)}_l(k)$ $(l = 1, 2, \cdots, h)$, and a pair of $\tilde{C}$ sets, $\tilde{C}^{(X)}(k)$ and $\tilde{C}^{(Y)}(k)$, (a)there are no connections between $\Gamma^{(X)}_p(k)$ and $\Gamma^{(Y)}_q(k)$, if $p \neq q$; (b)there are no connections between $\Gamma^{(X)}_l(k)$ and $\tilde{C}^{(Y)}(k)$; (c)there are connections between $\Gamma^{(X)}_l(k)$ and $\Gamma^{(Y)}_l(k)$ and between $\tilde{C}^{(X)}(k)$ and $\tilde{C}^{(Y)}(k)$.

Proof: Without loss of generality, we use denotation $\Gamma^{(X)}_0(k)$ to represent $\tilde{C}^{(X)}(k)$ and suppose the lth Γ set, $\Gamma^{(X)}_l(k)$ $(l = 1, 2, \cdots, h)$, is made at time k_l $(k_l \geq 2)$. Firstly, we consider the case that two pair of Γ sets, $\Gamma^{(X)}_0(k_1)$ and $\Gamma^{(X)}_1(k_1)$ $(X = A, B)$, are formed at time k_1. By *Corollary 1*, we have

$$(C^{(X)}(k_1-1) \cup R^{(X)}_C(k_1-1)) \supseteq C^{(X)}(k_1) = (\Gamma^{(X)}_0(k_1) \cup \Gamma^{(X)}_1(k_1)) \quad (T5-1)$$

By (18) and $(T5-1)$, we obtain

$$C^{(X)}(k_1-1) \supseteq \Gamma^{(X)}_0(k_1) \quad (T5-2)$$

Since $R^{(Y)}_C(k_1-1)$ has no connections with $C^{(X)}(k_1-1)$, from (18) and $(T5-2)$ we know that there are no connections between $\Gamma^{(Y)}_1(k_1)$ and $\Gamma^{(X)}_0(k_1)$. Thus, by *Theorem 4* and the properties (i) and (ii) of Γ and $\tilde{C}$ sets, we can obtain $\Gamma^{(Y)}_1(k) \supseteq \Gamma^{(Y)}_1(k+1)$ and $\Gamma^{(X)}_0(k) \supseteq \Gamma^{(X)}_0(k+1)$ for $k_1 \leq k \leq k_2$, where k_2 is the time, at which the next pair of Γ sets is formed. So, there are no connections between $\Gamma^{(Y)}_1(k)$ and $\Gamma^{(X)}_0(k)$ at any time until the next new pair of Γ sets is made.

Next, supposing there are no connections between $\Gamma^{(X)}_p(k)$ and $\Gamma^{(Y)}_q(k)$ for $p, q \leq M-1$ and $p \neq q$ in the case that M $(M \geq 2)$ pair of sets, $\Gamma^{(X)}_l(k)$ $(l = 0, 1, \cdots, M-1)$, exist, we will show that the $(M+1)th$ pair of sets, $\Gamma^{(Y)}_M(k)$, has no connections with the previous M pair of sets. Before the $(M+1)th$ pair of sets is made, since the M pair of sets have no connections each other, similar with the previous case, by *Theorem 4* and the properties (i) and (ii) of Γ and $\tilde{C}$ sets, we have

$$\Gamma^{(X)}_l(k) \supseteq \Gamma^{(X)}_l(k+1), \quad (T5-3)$$

for any time $k_{M-1} \leq k \leq k_M$. Since $R^{(Y)}_C(k_M-1)$ has no connections with $\Gamma^{(X)}_l(k_M-1)$, by (18) and $(T5-3)$ we know that $\Gamma^{(Y)}_M(k_M)$ has no connection with $\Gamma^{(X)}_l(k_M)$. So, there are no connections between any two Γ sets, $\Gamma^{(X)}_p(k_M)$ and $\Gamma^{(Y)}_q(k_M)$ for $p \neq q$, among the $M+1$ pair of sets at time k_M. Samely, by *Theorem 4* and the properties (i) and (ii) of Γ and $\tilde{C}$ sets, we can know that the previous conclusion can hold for any time k until the next new pair of Γ sets is made. Thus, the conclusions (a) and (b) of this theorem are proved. By the conclusions (a) and (b) and the definition of C set, we can easily obtain the conclusion (c). $\diamondsuit$

By *Theorem 4* and *Theorem 5*, we know that the behavior of each pair of Γ sets or $\tilde{C}$ sets is independent each other. Therefore, *Theorem 2* and the (b) of *Corollary 1* can also be applied to each pair of Γ sets or $\tilde{C}$ sets by regarding them as the pair of C sets. And, each pair of Γ sets or $\tilde{C}$ sets must also coexist like the pair of C sets.

Theorem 6: For any pair of Γ sets or $\tilde{C}$ sets, $\Gamma^{(X)}_l(k)$ and $\Gamma^{(Y)}_l(k)$, there is

$$(\Gamma^{(X)}_l(k) \cup \Gamma^{(Y)}_l(k)) \supset (\Gamma^{(X)}_l(k+1) \cup \Gamma^{(Y)}_l(k+1))$$

Proof: By the following two reasons, we obtain the previous relationship: (a)some units of $\Gamma^{(X)}_l(k)$ and $\Gamma^{(Y)}_l(k)$ must be transformed into R and/or H sets at time $k+1$ by *Corollary 1*; (b)No units can be added to $\Gamma^{(X)}_l(k+1)$ and $\Gamma^{(Y)}_l(k+1)$ by the properties (i) and (ii) of Γ and $\tilde{C}$ sets and *Theorem 4*. $\diamondsuit$

Note that by *Theorem 6* we cannot infer that $C^{(X)}(k) \supset C^{(X)}(k+1)$, since there may be new Γ set at time $k+1$.

Theorem 7: Supposing $\Gamma^{(X)}_l(k)$ and $\Gamma^{(Y)}_l(k)$ is a pair of Γ sets or $\tilde{C}$ sets, it is impossible that $\Gamma^{(X)}_l(k) \Rightarrow R^{(X)}(k+1)$ and $\Gamma^{(Y)}_l(k) \Rightarrow R^{(Y)}(k+1)$.

Proof: By $T2-2$ of *Theorem 2* and (15), it is impossible that all units of $\Gamma^{(X)}_l(k)$ and $\Gamma^{(Y)}_l(k)$ are suppressed at same

time (time $k+1$). That is, it is impossible that $\Gamma_l^{(X)}(k)$ and $\Gamma_l^{(Y)}(k)$ are completely transformed into $R^{(X)}(k+1)$ and $R^{(Y)}(k+1)$ at time $k+1$. $\diamondsuit$

By *Theorem 6*, we know that some Γ sets and/or $\tilde{C}$ sets must become empty ones after finite steps. Note that each pair of sets always becomes empty at the same time by their property of coexistence. The appearance of empty sets means the increasing of the size of $H^{(X)}(k)$, which cannot be transformed into other sets, by *Theorem 7*. The increasing of the size of $H^{(X)}(k)$ also equates to the decreasing of the size of $C^{(X)}(k)$ and $R^{(X)}(k)$. Since *Theorem 6* is always available when $C^{(X)}(k)$ exists and there are $R_{R_C}^{(X)}(k) \Rightarrow R_C^{(X)}(k+1)$ and $R_C^{(X)}(k) \Rightarrow C^{(X)}(k+1)$, until both $C^{(X)}(k)$ and $R_{R_C}^{(X)}(k) \cup R_C^{(X)}(k)$ become empty ones, the size of $C^{(X)}(k)$ and $R^{(X)}(k)$ will reduce with time k increasing. So, a finite time k_S must exists, at this time AAM reaches a stable state, at which both $C^{(X)}(k_S)$ and $R_{R_C}^{(X)}(k) \cup R_C^{(X)}(k)$ disappear and only $H^{(X)}(k_S)$ and $R_{R_R}^{(X)}(k_S)$ remain (Note that $R_{R_H}^{(X)}(k) \Rightarrow H^{(X)}(k+1)$).

Corollary 2: (a)When a single stored (0-1 binary) pattern is input, the input itself appears as a stable state. (b)When a composite pattern is inputed, an AAM can reach a stable state (stable solution) that is a (0-1 binary) sub-pattern of the input, in which there are no connections among 1-bits (active units).

To verify the ability of AAM to find a desired solution, we divide the stable solutions into three types. In general, a stable solution can be written as $H^{(X)} \cup R_{R_R}^{(X)} \cup \Omega$ and the time variable k_S is omitted. Since the activities of units of $R_{R_R}^{(X)}$ are 0 and Ω is the set of common units between $P^{(A)}$ and $P^{(B)}$, which is not changable all the time, we can divide the types of stable solutions only using $H^{(X)}$. Here, we also use the two-pattern case mentioned above. The first type of solution, *solution I*, is the case where $H^{(X)} = X$ & $H^{(Y)} = \phi$, where ϕ denotes the empty set. The recalled pattern $(X + \Omega)$ is a "winner". The second type of solution, *solution II*, is the case where $H^{(X)} = X$ & $H^{(Y)} = part(Y)$. The partly-recalled pattern, $part(Y)$, is cross-talk noise. The third type of solution, *solution III*, is the case where $H^{(X)} = part(X)$ & $H^{(Y)} = part(Y)$. Of these three types, it is clear that only *solution I* is desirable. We call *solution II* and *solution III* undesired solutions or spurious memories.

IV. PROBABILITY OF UNDESIRED SOLUTIONS

This section explains why the recall of spurious memories (undesired solutions) can be largely reduced compaired with HASP, and gives two conditions that guarantuee the non-existence of undesired solutions.

Since stable solutions should not have any interaction (inhibitory connection) between the active units according to *Corollary 2*, *Solution II* cannot exist if each unit in A and B has at least one connection with the competitor (B and A) when a composite pattern of A and B is input. Therefore, if any two stored patterns satisfy this condition, it can easily be inferred that *Solution II* does not exist for input composed of any combination of stored patterns. That is, when M patterns, $P^{(X_1)}, P^{(X_2)}, \cdots, P^{(X_M)}$, are stored, a necessary and sufficient condition that guarantees the non-existence of *Solution II* is as follows:

$$\prod_{i \in X_h} \sum_{j \in X_l} W_{ij} \neq 0, \qquad \text{for } h, l = 1, 2, \cdots, M \ (h \neq l)$$

Here $X_h = P^{(X_h)} - \Omega_{hl}$ and $X_l = P^{(X_l)} - \Omega_{hl}$, where Ω_{hl} is the common part between $P^{(X_h)}$ and $P^{(X_l)}$. This condition means that each sum of the weights of connections between each unit of X_h and the units of X_l is non-zero, that is, there is at least one connection between each unit of X_h and the units of X_l. It is clear that this condition is only related to the learning algorithm $(L-1)$ itself and will be satisfied easily if the number of stored patterns is not too large or the density of l-bits of the stored patterns is not too high.

For *solution III*, we can show that its probability of occurring in AAM is much smaller than in HASP as follows. As mentioned in section II, in the dynamics of the competitive network of HASP the adaptive weight α in (4) is 0 (i.e., $W_{ii}(t) = 0$ in (1)). Therefore, from (1) we can find that the smaller the inhibitions received by a HASP unit, the stronger the unit will become. This competing style is somewhat like a game for the comparison of perseverance. On the other hand, (1) or (13) show that the larger the inhibitions received by an AAM unit, the stronger the unit will fight (if the unit still survives) because of the first term of (1) or (13). Moreover, because the first term of (1) or (13) forms a self-loop, the stronger units may become much stronger and the weaker ones may become much weaker. This competing style is somewhat like "first come, first served".

Here, we also use the diagram shown in Fig. 4 to show the stable state. In the figure, i is a unit of $H^{(A)}$ and IB_i denotes the group of units in B that have connections with unit i. From the difference in the competing styles of AAM and HASP, groups such as IB_i that inhibit the strong units are much larger in AAM than that in HASP. Since the strong units will last out until the end in competition, we can obtain the following relation for IB_i in the stable state:

$$size(\bigcup_{i=1}^{n_S^{(A)}} IB_i^{(AAM)}) \gg size(\bigcup_{i=1}^{n_S^{(A)}} IB_i^{(HASP)}), \qquad (20)$$

where $n_S^{(A)}$ is the number of active units of A in the stable state. For IA_j of A, which is not shown in the figure, we can also obtain a similar relation as follows.

$$size(\bigcup_{j=1}^{n_S^{(B)}} IA_j^{(AAM)}) \gg size(\bigcup_{j=1}^{n_S^{(B)}} IA_j^{(HASP)}), \qquad (21)$$

where $n_S^{(B)}$ is the number of active units of B in the stable state. Since

$$Prob\{solutionIII\} = Prob\{H^{(B)} \neq \phi | H^{(A)} \neq \phi\} = Prob\{H^{(A)} \neq$$

$$= Prob\{(B - \bigcup_{i=1}^{n_S^{(A)}} IB_i) \neq \phi\} = Prob\{(A - \bigcup_{j=1}^{n_S^{(B)}} IA_j) \neq \phi\}, \quad (22)$$

where ϕ is an empty set, from (20)-(22), we obtain

$$Prob^{(AAM)}\{solutionIII\} \ll Prob^{(HASP)}\{solutionIII\} \quad (23)$$

From (22) we can also see that if $\bigcup_{i=1}^{n_S^{(A)}} IB_i = B$ (or $\bigcup_{j=1}^{n_S^{(B)}} IA_j = A$), *solution III* will not exist with probability 1 and pattern A (or pattern B) will be the winner. For AAM, especially if there is at least one unit i and j respectively in pattern A and B that have connections from all units of their competitors (B and A), they will last out until the end in competition. That is, $IB_i = B$ or $IA_j = A$ at the stable state. Therefore, *solution III* will not exist with probability 1. If we consider pattern A as an arbitrary stored pattern and pattern B as the supperposition of the remaining stored patterns, the above mentioned condition for AAM can be extended to all cases of input composed of any combination of stored patterns. That is, when M patterns, $P^{(X_1)}, P^{(X_2)}, \cdots, P^{(X_M)}$, are stored, a sufficient condition that guarantees the non-existence of *solution III* is as follows:

$$\text{for } P^{(A)} \in \{P^{(X_1)}, P^{(X_2)}, \cdots, P^{(X_M)}\},$$

$$\exists i \in P^{(A)} \quad \text{such that} \quad \prod_{j \in P^{(B)}} W_{ij} \neq 0, \quad (C-2)$$

where $P^{(B)} = \bigcup_{i=1}^{M} P^{(X_i)} - P^{(A)}$. This $(C-2)$ means that a unit (i) should exist in each stored pattern $(P^{(A)})$ such that all the connections between the unit (i) and the units of any other stored patterns excluding the common part with $P^{(A)}$ are non-zero. Evidently, condition $(C-2)$ includes condition $(C-1)$ and does not have any effect on a HASP for the probability of *solution III*. In general, this is a strict condition imposed on stored patterns. However, if we introduce sparse coding or a dummy augmentation encoding strategy that was used in [7][12][13] to stored patterns, it is not difficult to make them satisfy $(C-2)$.

By summarizing the analyses described in sections III and IV, we can see that AAM is stable when a composite pattern is input and the response converges to one of the three types of stable solutions . If condition $(C-1)$ is satisfied, one type of undesired solution will not exist . If condition $(C-2)$ is satisfied , both the types of undesired solutions will not exist. Even if condition $(C-2)$ is not satisfied, the probability of undesired solutions existing in AAM is much smaller than in HASP.

V. COMPUTER SIMULATIONS

To evaluate the performance of AAM and compare it with that of HASP, we conducted a large number of computer simulations. The AAM and HASP used in these experiments both consisted of 225 units. Their other parameters are as follows.

(1) *AAM*: The adaptive weight α is set to 1.2 (In fact, we can obtain almost unchangeable experimental results by setting α to a value between 1.0 and 2.0). The bias b is set to 0 in the experiments described in the first two subsections and 1.0 in the two following subsections. The initial weight of the learnable connections, W, is set to $1.5(b+1)$. The gain of RCU, W_{RI}, is set to the same value as W, rcs is set to "on" (1.0) for one step to suppress the competitive power of the currently recalled pattern, and rcs^* is set to 1.0 when rcs is on.

(2) *HASP*: The time constant is set to 4.0. The bias b is set to 0 in the experiments described in the first two subsections and 1.0 in the two following subsections. Each bit of the binary input pattern, P_i, is transfered into $(4.0b+1)P_i$. The initial weight of learnable connections, W, is set to $1.5(4.0b+1)$. The gain of RCU, W_{RI}, is set to the same value as W, rcs is set to "on" (1.0) for 40 steps, and rcs^* is set to 1.0 for one step when rcs is on.

Furthermore, the connecting coefficient from the input to neuron E, c_i, is set randomly to a value between 1.0 and 1.1 both in AAM and HASP.

A. Storage Capacity

Earlier studies measured the capacity of associative memory by equating it to a tight upper bound on the pattern ratio (*number of patterns/number of units*) or by directly measuring the number of patterns (fixing the number of units) for which all stored memories are stable states with a probability approaching 1 [7][10][11]. However, for AAM we know that all stored patterns are stable from the *Theorem 1* described in section III. We must therefore use another way of measuring capacity. In this paper, we define the storage capacity as the number of stored patterns that can gurantee the correct separation in the simplest case where the composite inputs are composed by only two stored patterns. In this experiment, we used randomly generated patterns, in which the sparse coding is introduced and the probability that a bit equals 1, $Prob\{P_i = 1\}$, is set to 0.2. We varied the number of stored patterns, M, from 2 to 30 in two increments and conducted 100 trials for each M. In each trial, two stored patterns were randomly selected to constitute a composite input and if the first separated pattern (stable solution) is the desired result, we registered it as a success. The percentages of success are plotted in Fig. 5. As can be seen, until $M = 16$, AAM can perform separation with a success rate of over 92% and its capacity is much larger than that of HASP.

B. Pattern Segmentation

In the first experiment of this subsection, we used 16 character patterns shown in Fig. 6. Examples of recall from a composite input by AAM and HASP when the 16 character patterns are stored are shown in Fig. 7 and Fig. 8. As can be seen, when a composite pattern composed by S (= 4) stored patterns is input, AAM can recall the S stored patterns one by one. On the other hand, the HASP recalls two spurious memories, although all the desired patterns were also recalled. By varying S from 2 to 16 and randomly combining the stored patterns to constitute in-

puts, we performed a large number of tests. Throughout these tests, the AAM could recall all the desired patterns from any combination of them without any spurious memory. In fact, if we examine the sufficient condition, $(C-2)$, that guarantees the non-existence of undesired solutions in AAM, we can see that it holds for the 16 character patterns. The HASP also recalled all the desired patterns, but was also highly likely to recall spurious memories, even if $S = 2$.

In the second experiment, we used randomly generated patterns, in which $Prob\{P_i = 1\}$ is set to 0.15. We fixed the number of stored patterns at $M = 16$ and varied S from 2 to M, performing 50 trials for each set of (M, S). In each trial, patterns were recalled one by one until the number of recalled patterns reached S. In order to have a unifying measure to evaluate the performance of AAM and HASP throughout the tests to be described later, we define a measure called *recall capability* as the ratio between the number of the recalled desired patterns and S when recall is performed S times. The experimenal results are shown in Table 1. From the results, we can see that the *recall capability* of AAM in all cases is over 0.92 and is much larger than that of HASP. In other words, AAM can recall more than $0.92S$ desired patterns from an input consisting of S patterns in all cases when recall is performed S times. On the other hand, the number of desired patterns recalled by HASP does not exceed $0.8S$ even in the best case, and it decreases to $0.5S$ when $S = 4$ or $S = 8$.

C. Pattern Recovery & Segmentation from Deficient Inputs

Pattern recovery from deficient inputs can be performed by separating the desired pattern from the input, in which a positive bias is added to each (0-1) bit (i.e., a positive bias, b, is set to AAM). Since pattern completion is a basic function of associative memory , we performed an experiment to recover incomplete patterns and to separate patterns from deficient composite inputs independently from the experiment described in the next subsection. In order to merely show the recovery capability, we used the 16 character patterns, for which the condition $(C-2)$ holds. M was fixed at 16 and S was varied from 1 to 16. Three perfect patterns were constructed by randomly combining S stored patterns for each S. For each perfect pattern, 20 deficient inputs were generated by negating 1-bits of the perfect pattern with a certain probability, called the deficiency rate $(D.R.)$. For each deficient input composed of S components, recall was performed one by one until the number of recalled patterns reached S. The experimental results are shown in Table 2. This table shows that AAM can recover single incomplete inputs with no errors when $D.R. \leq 0.4$, separate deficient composite inputs composed of two patterns with no errors when $D.R. \leq 0.2$, and separate deficient composite inputs composed of more than two patterns with high *Recall Capability* when $D.R. = 0.1$. From these results we can also see that the *Recall Capability* of AAM is much higher than that of HASP. Figure 9 shows an example of recall from a composite input with $D.R. = 0.2$.

D. Pattern Recovery & Segmentation from Noisy Inputs

Noisy input can be regarded as a pattern that is degraded by both deficiency and additive noise. It's recovery includes two processes: pattern completion and (additive) noise elimination. Since noise elimination is also performed by pattern segmentation (between desired pattern and the pattern that consists of noise), the competing style of "first come, first served" (α is large) should be considered for reducing the recall of spurious memories. On the other hand, from the fact that additive noise bits in input is fewer than 1-bits of desired pattern in general, it is clear that the competing style of the comparison of perseverance (α is small) is advantageous to making the desired pattern have stronger competitive power than the pattern that consists of noise.

This subsection shows that AAM not only has high recovery capability for noisy single patterns but also has relatively strong noise immunity for the noisy composite inputs, by selectting suitable α. We used the 16 character patterns in the experiment for the same reasons as in the previous subsection. Like before, M was fixed at 16 and S was varied from 1 to 16. Three perfect patterns were constructed by randomly combining S stored patterns for each S. Each of these perfect patterns was then modified to produce 20 noisy inputs by negating bits of the perfect pattern with a certain probability, called the noise rate $(N.R)$. For each noisy input composed of S components, recall is performed one by one until the number of recalled patterns reached S. In this experiment, α is set to 0.6 when $S = 1$, 0.8 when $S = 2$ or $S = 4$, and 1.2 when $S \geq 8$. The experimental results are shown in Table 3. This table shows that AAM can recover single noisy inputs with no errors when $N.R. \leq 0.1$ and separate noisy composite inputs of $S \geq 8$ with relatively high *Recall Capability* when $N.R. \leq 0.05$. From the results we can also see that the *Recall Capability* of AAM is much higher than that of HASP. Figure 10 shows an example of recall from a noisy input with $N.R. = 0.1$.

It should be noted that the *Recall Capability* of AAM is not so good when $S = 2$ or $S = 4$. As described in the first of the subsection, we know that α should be relatively small for noise elimination and relatively large for pattern segmentation. Therefore, when $S = 1$ or S is large, we can mainly consider either noise elimination or pattern segmentation, that is, we can make α as small as possible or as large as possible. In the other cases, however, a trade-off situation arises because noise immunity and pattern segmentation both have to be heavily considered. This is why the *Recall Capability* is not so good in such cases.

E. Speed of Recall

Another objective of AAM is to reduce the computation time for recall. As shown in above, AAM can work well in the case that the time constant is 0, while HASP must have a non-zero time constant, which has to be increased as S increases for stability [22]. As a result, the recall by AAM is much faster than by HASP. In the experiments described above, it takes about 8 iterations to recall one pattern by AAM, but about 48 iterations by HASP.

VI. CONCLUSION

We have proposed an adaptive type of associative memory, AAM, that is capable of separating patterns from composite inputs which might be degraded by deficiency or noise, and can also recover incomplete or noisy single patterns. We have analyzed the behavior of AAM in terms of stability, giving three types of solutions and we have shown that the recall of spurious memories can be greatly reduced compaired with an earlier form of associative memory, HASP. Furthermore, we have given two conditions that guarantees the non-existence of undesired solutions. Computer simulations have shown that the performance of AAM is much better than that of HASP in terms of both pattern segmentation and pattern recovery.

ACKNOWLEDGEMENTS

The author would like to thank Dr. Hideki Noda for his advice on this paper and Professor Yuzo Hirai for his encouragement during this work.

References

[1] T. Kohonen, "Correlation matrix memories," *IEEE Trans. Comput.*, vol. 21, no. 4, pp. 353-359, 1972.

[2] K. Nakano, "Associatron – A model of associative memory," *IEEE Trans. Sys., Man, Cybern.*, vol. 2, no. 3, pp. 380-388, 1972.

[3] T. Kohonen, *Associative Memory: A system-Theoretical Approach.* Berlin, Heidelberg, New York: Spring-Verlag, 1977.

[4] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. U.S.A.*, vol. 79, pp. 2554-2558, 1982.

[5] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Nat. Acad. Sci. U.S.A.*, vol. 81, pp. 3088-3092, 1984.

[6] B. Kosko, "Bidirectional associative memories," *IEEE Trans. Sys., Man, Cybern.*, vol. 18, no. 1, pp. 49-60, 1988.

[7] Y. F. Wang, J. B. Cruz, and J. H. Mulligan, "Two coding strategies for bidirectional associative memory", *IEEE Trans. Neural Networks*, vol. 1, No. 1. 81-92, 1990.

[8] M. H. Hassoun, "Dynamic heteroassociative neural memories," *Neural Networks*, vol. 2, pp. 275-287, 1989.

[9] M. H. Hassoun, *Associative Neural Memories–Theory and Implementation*, Oxford University Press, 1993.

[10] T. Wang, X. Zhuang, and X. Xing, "Weighted learning of bidirectional associative memories by global minimization," *IEEE Trans. Neural Networks*, vol. 3, No. 6, pp. 1010-1018, 1992.

[11] X. Zhuang, Y. Huang, and S. S. Chen, "Better learning for bidirectional associative memory", *Neural Networks*, vol. 6, pp. 1131-1146, 1993.

[12] Y. Hirai, "A learning network resolving multiple match in associative memory," *Proc. 6th Int. Conf. Pattern Recognition*, pp. 1049-1052, 1982.

[13] Y. Hirai, "A model of human associative processor (HASP)", *IEEE Trans . Sys., Man, Cybern.*, vol. 13 , no. 5, pp. 851-857 , 1983.

[14] D. Wang, J. Buhmann, and C. von der Malsburg, "Pattern segmentation in associative memory", *Neural Computation*, vol. 2, pp. 94-106 , 1990.

[15] D. Horn and M. Usher, "Parallel activation of memories in an oscillatory neural network", *Neural Computation*, vol. 3, pp.31-43, 1991.

[16] H. Sompolinsky and M. Tsodyks, "Segmentation by a network of oscillators with stored memories", *Neural Computation*, vol. 6, pp.642-657, 1994.

[17] C. von der Malsburg, "How are nervous structures organized?" In *Synergetics of the Brain. Proceedings of the International Symposium on Synergetics*, Springer, Berlin, Heidelberg, pp. 238-249, 1983-5.

[18] C. von der Malsburg, "Synaptic plasticity as basis of brain organization," In *The Neural and Molecular Bases of Learning*, Wiley, Chichester, pp. 411-431, 1987.

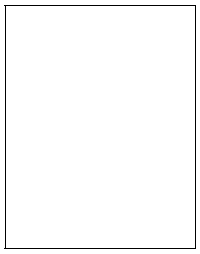
[19] Y. Hirai, "Representation of control structures by a model of associative processor, HASP," *Proc. 1st Annual International Conference on Neural Networks*, 1987.

[20] Y. Hirai and Q. Ma, "Modeling the process of problem-solving by associative networks capable of improving the performance," *Biol. Cybern.*, vol. 59, pp. 353-365, 1988.

[21] Q. Ma and Y. Hirai, "Modeling the acqusition of counting with an associative network," *Biol. Cybern.*, vol. 61, pp. 271-278, 1989.

[22] Q. Ma, "An adaptive type of competitive network for pattern segmentation in associative memory," *Technical Report of IEICE of Japan*, NC93-64, 1993-12.

[23] Q. Ma, "Adaptive type of associative memories with capability of pattern segmentation and noise immunity," *Proceedings of the 3d International Conference on Fuzzy Logic, Neural Nets and Soft Computing*, Japan, pp. 65-68, 1994-8.



Qing Ma Was born in Shaoxing, Zhengjiang, China, on Dec. 4. 1962. He received B.S. degree in electrical engineering from Beijing University of Aeronautics and Astronautics, China, in 1983, and the M.S. and Dr. Eng. degree in computer science from University of Tsukuba, Japan, in 1987 and 1990.

Since 1993 he has been working at the Communications Research Laboratory of the Japan Ministry of Posts and Telecommunications, currently as a Senior Research Scientist. His research interests include neural networks and its applications in knowledge representation and natural language processing.

Figure Captions

Fig. 1 The unit structure of AAM
Fig. 2 The structure of a readout control unit
Fig. 3 A diagram for representing intermediate competing states when a composite pattern composed of $P^{(A)}$ and $P^{(B)}$ is input
Fig. 4 A diagram for representing stable states when a composite pattern composed of $P^{(A)}$ and $P^{(B)}$ is input
Fig. 5 Simulated storage capacity. Each point on the curve represents 100 trials
Fig. 6 The 16 character patterns used in the tests
Fig. 7 An example of recall by the AAM from a composite input
Fig. 8 An example of recall by the HASP from a composite input
Fig. 9 An example of recall from a composite input with a deficiency rate of 0.2
Fig. 10 An example of recall from a noisy input with a noise rate of 0.1