The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network

Gail A. Carpenter†
Department of Mathematics
Northeastern University
Boston, MA 02115
and
Center for Adaptive Systems
111 Cummington Street
Boston University
Boston, MA 02215

and

Stephen Grossberg‡
Center for Adaptive Systems
111 Cummington Street
Boston University
Boston, MA 02215

August, 1987
Revised December 1987

Computer, 1988, **21**, 77–88

† Supported in part by the Air Force Office of Scientific Research (AFOSR F49620-86-C-0037 and AFOSR F49620-87-C-0018), the Army Research Office (ARO DAAG-29-85-K-0095), and the National Science Foundation (NSF DMS-86-11959).

‡ Supported in part by the Air Force Office of Scientific Research (AFOSR F49620-86-C-0037 and AFOSR F49620-87-C-0018), the Army Research Office (ARO DAAG-29-85-K-0095), and the National Science Foundation (NSF IRI-84-17756).

Acknowledgements: We wish to thank Cynthia Suchta and Carol Yanakakis for their valuable assistance in the preparation of the manuscript.

Attention and Expectation in Self-Organizing Learning and Recognition Systems

One of the central goals of computer science is to design intelligent machines capable of autonomous learning and skillful performance within complex environments that are not under strict external control. Many scientists have turned to a study of human capabilities as a source of new ideas for designing such machines. When one undertakes such a study, one is confronted by a number of basic issues of which we are all aware through our own personal experiences:

Why do we pay attention? Why do we learn expectations about the world? In particular, how do we cope so well with unexpected events, and how do we manage to do so as well as we do when we are on our own, and do not have a teacher as a guide? How do we learn what combinations of facts are useful for dealing with a given situation and what combinations of facts are irrelevant? How do we recognize familiar facts so quickly even though we may know many other things? How do we join together knowledge about the external world with information about our internal needs to quickly make decisions that have a good chance of satisfying these needs? Finally, what do all of these properties have in common?

The Stability-Plasticity Dilemma and Adaptive Resonance Theory

One answer to these questions has been found through the attempt to solve a basic design problem faced by all intelligent systems that are capable of autonomously adapting in real-time to unexpected changes in their world. This design problem is called the stability-plasticity dilemma and a theory called Adaptive Resonance Theory, or ART, is being developed that suggests a solution to this problem.

The stability-plasticity dilemma asks how a learning system can be designed to remain plastic, or adaptive, in response to significant events, yet also remain stable in response to irrelevant events? How does the system know how to switch between its stable and its plastic modes to achieve stability without rigidity and plasticity without chaos? In particular, how can it preserve its previously learned knowledge while continuing to learn new things; what prevents the new learning from washing away the memories of prior learning?

The ubiquity of this problem can be dramatized by imagining that you have grown up in Boston before moving to Los Angeles, but periodically return to Boston to visit your parents. Although you may need to learn many new things to enjoy life in Los Angeles, these new learning experiences do not prevent you from remembering how to find your parent's house or otherwise getting around Boston. A multitude of similar examples illustrate that we are designed to successfully adapt to environments whose rules may change—without necessarily forgetting our old skills. Moreover, we are designed to successfully adapt to environments whose rules may change *unpredictably*, and can do so even if no one tells us that the environment has changed. We can adapt, in short, without a teacher, and through a direct confrontation with our experiences. Such adaptation is called *self-organization* in the network modeling literature.

One of the key computational ideas that is rigorously demonstrated within Adaptive Resonance Theory is that top-down learned expectations focus attention upon bottom-up information in a way that protects previously learned memories from being washed away by new learning, and enables new learning to be automatically incorporated into the total knowledge base of the system in a globally self-consistent way.

The ART achitectures that are discussed herein are neural networks that self-organize stable recognition codes in real-time in response to arbitrary sequences of input patterns. Within such an ART architecture, the process of adaptive pattern recognition is a special case of the more general cognitive process of hypothesis discovery, testing, search, classification, and learning. This latter property opens the possibility of applying ART systems to more general problems of adaptively processing large abstract information sources and data bases. The present article outlines the main computational properties of these ART architectures, while comparing and contrasting these properties with those of alternative learning and recognition systems. Technical details are described in greater detail elsewhere[1,2], and articles wherein the theory was discovered through the analysis and prediction of interdisciplinary data about brain and behavior are collected in several books[3,4].

Competitive Learning Models

ART models grew out of an analysis of a simpler type of adaptive pattern recognition network which is often called a *competitive learning* model. Competitive learning models were developed in the early 1970's through contributions of Christoph von der Malsburg[5] and Stephen Grossberg, leading in 1976 to the description of these models in several forms in which they are used today[6]. Authors such as Shun-ichi Amari[7], Leon Cooper[8], and Teuvo Kohonen[9] have further developed these models. Kohonen[9] has made particularly strong use of competitive learning in his work on self-organizing maps. Grossberg[4] has provided a historical discussion of the development of competitive learning models.

In a competitive learning model (Figure 1), a stream of input patterns to a network F_1 can train the adaptive weights, or long term memory (LTM) traces, that multiply the signals in the pathways from F_1 to a coding level F_2. In the simplest such model, input patterns to F_1 are normalized before passing through the adaptive filter defined by the pathways from F_1 to F_2. Level F_2 is designed as a competitive network capable of choosing the node which receives the largest total input ("winner-take-all"). The winning population then triggers associative pattern learning within the vector of LTM traces which sent its inputs through the adaptive filter.

For example, as in Figure 1, let I_i denote the input to the ith node v_i of F_1, $i = 1, 2, \ldots, M$; let x_i denote the activity, or short term memory (STM) trace, of v_i; let x_j denote the activity, or STM trace, of the jth node v_j of F_2, $j = M+1, \ldots, N$; and let z_{ij} denote the adaptive weight, or long term memory (LTM) trace, of the pathway from v_i to v_j. Then let

$$x_i = \frac{I_i}{\sum_{k=1}^{M} I_k} \tag{1}$$

be the normalized activity of v_i in response to the input pattern $I = (I_1, I_2, \ldots, I_M)$. For simplicity, let the output signal S_i of v_i equal x_i. Let

$$T_j = \sum_{i=1}^{M} x_i z_{ij} \tag{2}$$

be the total signal received at v_j from F_1, let

$$x_j = \begin{cases} 1 & \text{if } T_j > \max(T_k : k \neq j) \\ 0 & \text{if } T_j < \max(T_k : k \neq j) \end{cases} \tag{3}$$

summarize the fact that the node x_j in F_2 which receives the largest signal is chosen for short-term memory storage, and let a differential equation

$$\frac{d}{dt} z_{ij} = \epsilon x_j(-z_{ij} + x_i) \tag{4}$$

specify that only the vector $Z_j = (z_{1j}, z_{2j}, \ldots, z_{Mj})$ of adaptive weights which abut the winning node v_j is changed due to learning. Vector Z_j learns by reducing the error between itself and the normalized vector $X = (x_1, x_2, \ldots, x_M)$ in the direction of steepest descent.

There are several equivalent ways to describe how such a system recognizes input patterns I that are presented to F_1. The winning node v_j in F_2 is said to code, classify, cluster, partition, compress, or orthogonalize these input patterns. In engineering, such a scheme is said to perform adaptive vector quantization. In cognitive psychology, it is said to perform categorical perception[3]. In categorical perception, input patterns are classified into mutually exclusive recognition categories which are separated by sharp categorical boundaries. A sudden switch in pattern classification can occur if an input pattern is deformed so much that it crosses one of these boundaries and thereby causes a different node v_j to win the competition within F_2. Categorical perception, in the strict sense of the word, occurs only if F_2 makes a choice. In more general competitive learning models, compressed but distributed recognition codes are generated by the model's coding level, or levels[3,4,9].

In response to certain input environments, a competitive learning model possesses very appealing properties. It has been mathematically proved[6] that, if not too many input patterns are presented to F_1, or if the input patterns form not too many clusters, relative to the number of coding nodes in F_2, then learning of the recognition code eventually stabilizes and the learning process elicits the best distribution of LTM traces that is consistent with the structure of the input environment.

Despite the demonstration of input environments that can be stably coded, it has also been shown, through explicit counterexamples[1,2,6], that a competitive learning model does not always learn a temporally stable code in response to an arbitrary input environment. In these counterexamples, as a list of input patterns perturbs level F_1 through time, the response of level F_2 to the *same* input pattern can be different on each successive presentation of that input pattern. Moreover, the F_2 response to a given input pattern might never settle down as learning proceeds.

Such unstable learning in response to a prescribed input is due to the learning that occurs in response to the other, intervening, inputs. In other words, the network's adaptability, or plasticity, enables prior learning to be washed away by more recent learning in response to a wide variety of input environments. In fact, there exist infinitely many input environments in which periodic presentation of just four input patterns can cause temporally unstable learning[1,2]. Learning can also become unstable due to simple changes in an input environment. Changes in the probabilities of inputs, or in the deterministic sequencing of inputs, can readily wash away prior learning. This instability problem is not, moreover, peculiar to competitive learning models. The problem is a basic one because it arises from a combination of the very features of an adaptive coding model that, on the surface, seem so desirable: its ability to learn from experience and its ability to code, compress, or categorize many patterns into a compact internal representation. Due to these properties, when a new input pattern I retrains a vector Z_j of LTM traces, the set of *all* input patterns coded by v_j changes too, because a change in Z_j in (2) can reverse the inequalities in (3) in response to many of the input patterns that were previously coded by v_j.

Learning systems which can become unstable in response to many input environments cannot safely be used in autonomous machines which may be unexpectedly confronted by one of these environments on the job. Adaptive Resonance Theory was introduced in 1976 to show how to embed a competitive learning model into a *self-regulating control structure* whose autonomous learning and recognition proceed stably and efficiently in response to an *arbitrary* sequence of input patterns.

Self-Stabilized Learning by an ART Architecture in an Arbitrary Input Environment

Figure 2 schematizes a typical example from a class of architectures called ART 1. It has been mathematically proved[1] that an ART 1 architecture is capable of stably learning a recognition code in response to an arbitrary sequence of *binary* input patterns until it utilizes its full memory capacity. Moreover, the adaptive weights, or LTM traces, of an ART 1 system oscillate at most once during learning in response to an arbitrary binary input sequence, yet do not get trapped in spurious memory states or local minima. After learning self-stabilizes, the input patterns directly activate the F_2 codes that represent them best.

As in a competitive learning model, an ART architecture encodes a new input pattern, in part, by changing the adaptive weights, or LTM traces, of a bottom-up adaptive filter. This filter is contained in the pathways leading from a feature representation field F_1 to a category representation field F_2. In an ART network, however, it is a second, top-down adaptive filter, contained in the pathways from F_2 to F_1, that leads to the crucial property of code self-stabilization. Such top-down adaptive signals play the role of learned expectations in an ART system. Before considering details about how the ART control structure automatically stabilizes the learning process, we will sketch how self-stabilization occurs in intuitive terms.

Suppose that an input pattern I activates F_1. Let F_1 in turn, activate the code, or hypothesis, symbolized by the node v_{j_1} at F_2 which receives the largest total signal from F_1. Then F_2 quickly reads-out its learned top-down expectation to F_1, whereupon the bottom-up input pattern and top-down learned expectation are matched across F_1. If these patterns are badly matched, then a mismatch event takes place at F_1 which triggers a reset burst to F_2. This reset burst shuts off node v_{j_1} for the remainder of the coding cycle, and thereby deactivates the top-down expectation controlled by v_{j_1}. Then F_1 quickly reactivates essentially the same bottom-up signal pattern to F_2 as before. Level F_2 reinterprets this signal pattern, conditioned on the hypothesis that the earlier choice v_{j_1} was incorrect, and another node v_{j_2} is automatically chosen. The parallel search, or hypothesis testing, cycle of bottom-up adaptive filtering from F_1 to F_2, code (or hypothesis) selection at F_2, read-out of a top-down learned expectation from F_2 to F_1, matching at F_1, and code reset at F_2 now repeats itself automatically at a very fast rate until one of three possibilities occurs: a node v_{j_m} is chosen whose top-down expectation approximately matches input I, or a previously uncommitted F_2 node is selected, or the full capacity of the system is used and cannot accomodate input I. Until one of these outcomes prevails, essentially no learning occurs, because all the STM computations of the hypothesis testing cycle proceed so quickly that the more slowly varying LTM traces in the bottom-up and top-down adaptive filters cannot change in response to them. Significant learning occurs in response to an input pattern only after the hypothesis testing cycle that it generates comes to an end.

If the hypothesis testing cycle ends in an approximate match, then the bottom-up input pattern and the top-down expectation quickly deform the activity pattern $X = (x_1, x_2, \ldots, x_M)$ across F_1 into a net pattern that computes a fusion, or consensus, between the bottom-up and top-down information. This fused pattern represents the attentional focus of the system. When fusion occurs, the bottom-up and top-down signal patterns mutually reinforce each other via feedback and the system gets locked into a resonant state of STM activation. Only then can the LTM traces learn. What they learn is any new information about the input pattern that is represented within the fused activation pattern across F_1. The fact that learning occurs only in the resonant state suggested the name Adaptive Resonance Theory. Thus the system allows one of its prior learned codes to be altered only if an input pattern is sufficiently similar to what it already knows to risk a further refinement of its knowledge.

If the hypothesis testing cycle ends by selecting an uncommitted node at F_2, then the bottom-up and top-down adaptive filters that are linked to this node learn the F_1 activation pattern that is generated directly by the input. No top-down alteration of the

F_1 activation pattern occurs in this case. If the full capacity has been exhausted and no adequate match exists, learning is automatically inhibited.

In summary, an ART network either refines its already learned codes based upon new information that can be safely accomodated into them via approximate matches, or selects new nodes for initiating learning of novel recognition categories, or defends its fully committed memory capacity against being washed away by the incessant flux of new input events.

Comparison with Alternative Learning Schemes

Many computational details have been worked out to make this scheme work well in an autonomous setting[1,2]. Before describing some of these details, we can already see that ART architectures differ from other popular neural network learning schemes, such as autoassociators, the Boltzmann machine, and back propagation[9-11] in a number of basic ways. These differences are schematized in Table 1.

The most robust differences are that an ART architecture is designed to learn quickly and stably in real-time in response to a possibly nonstationary world with an unlimited number of inputs until it utilizes its full memory capacity. Many alternative learning schemes become unstable unless they learn slowly in a controlled stationary environment with a carefully selected total number of inputs and do not use their full memory capacity[12]. For example, a learning system that is not self-stabilizing experiences a capacity catastrophe in response to an unlimited number of inputs: new learning washes away memories of prior learning if too many inputs perturb the system. To prevent this from happening, either the total number of input patterns that perturbs the system needs to be restricted, or the learning process itself must be shut off, before the capacity catastrophe occurs.

Shutting off the world is not possible in many real-time applications. In particular, how does such a system subsequently allow a familiar input to be processed and recognized, but block the processing of a novel input pattern before it can destabilize its prior learning? In the absence of a self-stabilization mechanism, an external teacher must act as the system's front end to independently recognize the inputs and make the decision. Shutting off learning at just the right time to prevent either a capacity catastrophe or a premature termination of learning from occurring would also require an external teacher. In either case the external teacher must be able to carry out the recognition tasks that the learning system was supposed to carry out. Hence non-self-stabilizing learning systems are not capable of functioning autonomously in ill-controlled environments.

In learning systems wherein an external teacher is needed to supply the correct representation to be learned, the learning process is often driven by mismatch between desired and actual outputs[10,11]. Such schemes must learn slowly, and in a stationary environment, or else risk unstable oscillations in response to the mismatches. They can also be destabilized if the external teaching signal is noisy, because such noise creates spurious mismatches. These learning models are also prone to getting trapped in local minima, or globally incorrect solutions. Models such as simulated annealing and the Boltzmann machine[10] use internal system noise to escape local minima and to thereby approach a more global minimum. An externally controlled (temperature) parameter regulates this process as it is made to converge ever more slowly to a critical value. In ART, by contrast, approximate matches, rather than mismatches, drive the learning process. Learning in the approximate-match mode enables rapid and stable learning to occur while buffering the system's memory against external noise. The hypothesis testing cycle replaces internal system noise as a scheme for discovering a globally correct solution, and does not utilize an externally controlled temperature parameter or teacher.

Attentional Priming and Prediction: Matching by the 2/3 Rule

One of the key constraints upon the design of the ART 1 architecture is its rule for matching a bottom-up input pattern with a top-down expectation at F_1. This rule is called the 2/3 Rule[1]. The 2/3 Rule is necessary to regulate both the hypothesis testing cycle and the self-stabilization of learning in an ART 1 system.

The 2/3 Rule reconciles two properties whose simplicity tends to conceal their fundamental nature: In response to an arbitrary bottom-up input pattern, F_1 nodes can be *supraliminally* activated; that is, activated enough to generate output signals to other parts of the network and thereby to initiate the hypothesis testing cycle. In response to an arbitrary top-down expectation, however, F_1 nodes are only *subliminally* activated; they sensitize, prepare, or attentionally *prime* F_1 for future input patterns that may or may not generate an approximate match with this expectation, but do not, in themselves, generate output signals. Such a subliminal reaction enables an ART system to anticipate future events and to thereby function as an "intentional" machine. In particular, if an attentional prime is locked into place by a high gain top-down signal source, then an ART system can automatically suppress all inputs that do not fall into a sought-after recognition category, yet amplify and hasten the processing of all inputs that do[3].

In order to implement the 2/3 Rule, it is necessary to assure that F_1 can distinguish between bottom-up and top-down signals, so that it can supraliminally react to the former and subliminally react to the latter. In ART 1, this distinction is carried out by a third F_1 input source, called an *attentional gain control* channel, that responds differently to bottom-up and top-down signals.

Figure 2 describes how this gain control source works. When it is activated, it excites each F_1 node equally. The 2/3 Rule says that at least 2 out of 3 input sources are needed to supraliminally activate an F_1 node: a bottom-up input, a top-down input, and a gain control input. In the top-down processing mode (Figure 2a), each F_1 node receives a signal from at most one input source, hence is only subliminally activated. In the bottom-up processing mode (Figure 2b), each active bottom-up pathway can turn on the gain control node, whose output, once on, is independent of the total number of active bottom-up pathways. Then all F_1 nodes receive at least a gain control input, but only those nodes that also receive a bottom-up input are supraliminally activated. When both bottom-up and top-down inputs reach F_1 (Figure 2c), the gain control source is shut off, so that only those F_1 nodes which receive top-down confirmation of the bottom-up input are supraliminally activated. In this case, the 2/3 Rule maintains supraliminal activity only within the spatial intersection of the bottom-up input pattern and the top-down expectation. Consequently, if a bottom-up input pattern, as in Figure 2b, causes the read-out of a badly matched top-down expectation, as in Figure 2c, then the total number of supraliminally active F_1 nodes can suddenly decrease, and thereby cause a decrease in the total ouput signal emitted by F_1. This property is used heavily in controlling the hypothesis testing and self-stabilization processes, as we now show.

Automatic Control of Hypothesis Testing by Attentional-Orienting Interactions

An ART architecture automates its hypothesis testing cycle through interactions between an attentional subsystem and an orienting subsystem. These subsystems in the ART 1 architecture are schematized in Figure 3.

The orienting subsystem A generates an output signal only when a mismatch occurs between bottom-up input pattern and top-down expectation at level F_1 of the attentional subsystem. Thus A functions like a novelty detector. The output signal from A is called an *STM reset wave* because it selectively inhibits the active node(s) at level F_2 of the attentional subsystem. The novelty detector A hereby disconfirms the F_2 hypothesis that led to the F_1 mismatch.

The 2/3 Rule controls the reset wave emitted by A as follows. When a bottom-up input pattern is presented, each of the active input pathways to F_1 also sends a signal to the orienting subsystem A, where all of these signals are added up. When the input pattern activates F_1, each of the activated F_1 nodes sends an inhibitory signal to A. The system is designed so that the total inhibitory signal is larger than the total excitatory signal. Thus in the bottom-up mode, there is a balance between active F_1 nodes and active input lines that prevents a reset wave from being triggered. (Note that level F_1 in ART 1 is not normalized as it was in equation (1) of the competitive learning model and in the ART 2 systems discussed below. The decision of whether and how to normalize depends upon the design of the whole system.)

This balance is upset when a top-down expectation is read-out that mismatches the bottom-up input pattern at F_1. As in Figure 2c, the total output from F_1 then decreases by an amount that grows with the severity of the mismatch. If the attenuation is sufficiently great, then inhibition from F_1 to A can no longer prevent A from emitting a reset wave. A parameter ρ called the *vigilance parameter* determines how large a mismatch will be tolerated before A emits a reset wave. High vigilance forces the system to search for new categories in response to small differences between input and expectation. Then the system learns to classify input patterns into a large number of fine categories. Low vigilance enables the system to tolerate large mismatches and to thus group together input patterns according to a coarse measure of mutual similarity. The vigilance parameter may be placed under external control, being increased, for example, when the network is "punished" for failing to distinguish two inputs that give rise to different consequences[1,3].

Figure 4 schematizes learning in response to the first twenty input presentations of a computer simulation of alphabet learning. After presenting the twentieth input, 9 recognition categories have formed when $\rho = .8$, but only 4 categories have been formed when $\rho = .5$. In this computer experiment, learning self-stabilized after at most 3 presentations of the 26 letters at any level of vigilance, and the learned LTM codes were more abstract—that is, less letter-like—at lower levels of vigilance.

Figure 5 illustrates how these properties of the interaction between levels F_1, F_2, and A regulate the hypothesis testing cycle of the ART 1 system. In Figure 5a, an input pattern I generates an STM activity pattern X across F_1. The input pattern I also excites the orienting subsystem A, but pattern X at F_1 inhibits A before it can generate an output signal. Activity pattern X also elicits an output pattern S which activates the bottom-up adaptive filter $T = ZS$, where Z is the matrix of bottom-up LTM traces. As a result an STM pattern Y becomes active at F_2. In Figure 5b, pattern Y generates a top-down output U through the adaptive filter $V = \hat{Z}U$, where $\hat{Z}$ is the matrix of top-down LTM traces. Vector V is the top-down expectation that is read into F_1. Expectation V mismatches input I, thereby significantly inhibiting STM activity across F_1. The amount by which activity in X is attenuated to generate the activity pattern X^* depends upon how much of the input pattern I is encoded within the expectation V, via the 2/3 Rule.

When a mismatch attenuates STM activity across F_1, the total size of the inhibitory signal from F_1 to A is also attenuated. If the attenuation is sufficiently great, inhibition from F_1 to A can no longer prevent the arousal source A from firing. Figure 5c depicts how disinhibition of A releases an arousal burst to F_2 which equally, or nonspecifically, excites all the F_2 cells. The cell populations of F_2 react to such an arousal signal in a state-dependent fashion. In the special case that F_2 chooses a single population for STM storage, the arousal burst selectively inhibits, or resets, the active population in F_2. This inhibition is long-lasting.

In Figure 5c, inhibition of Y leads to removal of the top-down expectation V, and thereby terminates the mismatch between I and V. Input pattern I can thus reinstate the original activity pattern X across F_1, which again generates the output pattern S from F_1 and the input pattern T to F_2. Due to the enduring inhibition at F_2, the input pattern T can no longer activate the original pattern Y at F_2. Level F_2 has been conditioned by the disconfirmation of the original hypothesis. A new pattern Y^* is thus generated at F_2 by I (Figure 5d).

The new activity pattern Y^* reads-out a new top-down expectation V^*. If a mismatch again occurs at F_1, the orienting subsystem is again engaged, thereby leading to another arousal-mediated reset of STM at F_2. In this way, a rapid series of STM matching and reset events may occur. Such an STM matching and reset series controls the system's hypothesis testing and search of LTM by sequentially engaging the novelty-sensitive orienting subsystem. Although STM is reset sequentially in time via this mismatch-mediated, self-terminating LTM search process, the mechanisms that control the LTM search are all parallel network interactions, rather than serial algorithms. Such a parallel search scheme continuously adjusts itself to the system's evolving LTM codes. The LTM code depends upon both the system's initial configuration and its unique learning history, and hence cannot be predicted *a priori* by a pre-wired search algorithm. Instead, the mismatch-mediated engagement of the orienting subsystem triggers a process of parallel self-adjusting search that tests only the hypotheses most likely to succeed, given the system's unique learning history.

The mismatch-mediated search of LTM ends when an STM pattern across F_2 reads-out a top-down expectation that approximately matches I, to the degree of accuracy required by the level of attentional vigilance, or that has not yet undergone any prior learning. In the former case, the accessed recognition code is refined based upon any novel information contained in the input I; that is, based upon the activity pattern resonating at F_1 that fuses together bottom-up and top-down information according to the 2/3 Rule. In the latter case, a new recognition category is then established as a new bottom-up code and top-down template are learned.

ART 2: Learning to Recognize an Analog World

Although self-organized recognition of binary patterns is useful in many applications, such as recognition of printed or written text, as in Figure 4, many other applications require the ability to categorize arbitrary sequences of analog (including binary) input patterns. A class of architectures, generically called ART 2, has been developed for this purpose[2]. Given the enhanced capabilities of ART 2 architectures, a sequence of arbitrary input patterns can be fed through an arbitrary preprocessor before the output patterns of the preprocessor are fed as inputs into an ART 2 system for automatic classification. Figure 6 illustrates how an ART 2 architecture has quickly learned to stably classify 50 analog input patterns, chosen to challenge the architecture in multiple ways, into 34 recognition categories after a single learning trial. Figure 7 illustrates how the same 50 input patterns have been quickly classified into 23 coarser categories after a single learning trial, using a smaller setting of the vigilance parameter.

ART 2 architectures can autonomously classify arbitrary sequences of analog input patterns into categories of arbitrary coarseness while suppressing arbitrary levels of noise. They accomplish this by modifying the ART 1 architecture to incorporate solutions of several additional design problems into their circuitry. In particular, level F_1 is split into separate sublevels for receiving bottom-up input patterns, for receiving top-down expectations, and for matching the bottom-up and top-down data, as in Figure 8. Three versions of the ART 2 architecture are now being applied to problems such as visual pattern recognition, speech perception, and radar classification. In addition to research on ART undertaken at several universities, applications are also being developed at government laboratories and industrial firms including M.I.T. Lincoln Laboratory; Booz-Allen and Hamilton, Inc.; Hecht-Nielsen Neurocomputer Corp.; Science Applications International Corp.; the U.S. Army Research Center at Redstone Arsenal; and Wright-Patterson Air Force Base.

Invariant Visual Pattern Recognition

An application to invariant visual pattern recognition is being carried out in a collaboration between Boston University and M.I.T. Lincoln Laboratory. This application uses a three stage preprocessor, summarized in Figure 9. First, the image figure to be recognized is detached from the image background using laser radar sensors. This can be accomplished by intersecting the images formed by two laser sensors: the image formed by a range detector focussed at the distance of the figure, with the image formed by another laser detector that is capable of differentiating figure from background, such as a doppler image when the figure is moving or the intensity of laser return when the figure is stationary[13]. The second stage of the preprocessor contains a neural network, called a Boundary Contour System[3,4], that detects, sharpens, regularizes, and completes the boundaries within noisy images. The third stage of the preprocessor contains a Fourier-Mellin filter, whose output spectra are invariant under such image transformations as 2-D spatial translation, dilation, and rotation[14]. Thus the input patterns to ART 2 are the invariant spectra of completed boundary segmentations of laser radar sensors. By setting ART 2 parameters to suppress (up to) a prescribed level of input noise and to tolerate (up to) a prescribed level of input deformation, this system defines a compact circuit capable of autonomously learning to recognize visual targets that are deformed, rotated, dilated, and shifted. Although this preprocessor does not purport to provide a biological solution to the problem of invariant visual object recognition, it is known that the mammalian visual cortex does carry out computations that are analogous to aspects of the second and third stages of this preprocessor[3,4,15].

The Three R's: Recognition, Reinforcement, and Recall

Recognition is only one of several processes whereby an intelligent system can learn a correct solution to a problem. Reinforcement and recall are no less important in designing an autonomous intelligent system.

Reinforcement, notably reward and punishment, provides additional information in the form of environmental feedback based on the success or failure of actions triggered by a recognition event. Reward and punishment calibrate whether the action has or has not satisfied internal needs, which in the biological case include hunger, thirst, sex, and pain reduction, but may in machine applications include a wide variety of internal cost functions. Reinforcement can modify the formation of recognition codes and can shift attention to focus upon those codes whose activation promises to satisfy internal needs based upon past experience. For example, both green and yellow bananas may be recognized as part of a single recognition category until reinforcement signals, contingent upon eating these bananas, differentiates them into separate categories.

Recall can generate equivalent responses or actions to input events that are classified by different recognition codes. For example, printed and script letters may generate distinct recognition codes, yet can also elicit identical learned naming responses.

Our own research program at Boston University during the past two decades has been devoted to discovering and implementing models of self-organizing biological systems wherein all the ingredients of recognition, reinforcement, and recall are joined together into a single integrated circuit[3,4]. The system depicted in Figure 10 provides a framework for implementing some of these circuit designs. In particular, as ART 2 self-organizes recognition categories in response to the preprocessed inputs, its categorical choices at the F_2 classifying level self-stabilize through time. In examples wherein F_2 makes a choice, it can be used as the first level of an ART 1 architecture, or yet another ART 2 architecture, should one prefer. Let us call the classifying level of this latter architecture F_3. Level F_3 can be used as a source of pre-wired priming inputs to F_2. Alternatively, as in Figure 10, self-stabilizing choices by F_3 can quickly be learned in response to the choices made at F_2. Then F_3 can be used as a source of self-organized priming inputs to F_2, and a source of priming patterns can be associated with each of the F_3 choices via mechanisms of associative pattern learning[3]. After learning of these primes takes place, turning on a particular prime can activate a learned $F_3 \rightarrow F_2$ top-down expectation. Then F_2 can be supraliminally activated only by an input exemplar which is a member of the recognition category of the primed F_2 node. The architecture ignores all but the primed set of input patterns. In other words, the prime causes the architecture to pay attention only to expected sources of input information. Due to the spatial invariance properties of the preprocessor, the expected input patterns can be translated, dilated, or rotated in 2-D without damaging recognition. Due to the similarity grouping properties of ART 2 at a fixed level of vigilance, suitable deformations of these input patterns, including deformations due to no more than anticipated levels of noise, can also be recognized.

The output pathways from level F_2 of ART 2 to the postprocessor can learn to recall any spatial pattern or spatiotemporal pattern of outputs by applying theorems about associative learning in a type of circuit called an avalanche[3]. In particular, distinct recognition categories can learn to generate identical recall responses. Thus the architecture as a whole can stably self-organize an invariant recognition code and an associative map to an arbitrary format of output patterns.

The interactions (priming → ART) and (ART → postprocessor) in Figure 10 can be modified so that output patterns are read out only if the input patterns have yielded rewards in the past and if the machine's internal needs for these rewards have not yet been satisfied[3,4]. In this variation of the architecture, the priming patterns supply motivational signals for releasing outputs only if an input exemplar from an internally desired recognition category is detected. The total circuit forms a neural network architecture which can stably self-organize an invariant pattern recognition code in response to a sequence of analog or binary input patterns; be attentionally primed to ignore all but a designated category of input patterns; automatically shift its prime as it satisfies internal criteria in response to actions based upon the recognition of a previously primed category of input patterns; and learn to generate an arbitrary spatiotemporal output pattern in response to any input pattern exemplar of an activated recognition category. Such circuits, and their real-time adaptive autonomous descendents, may prove to be useful in some of the many applications where preprogrammed rule-based systems, and systems requiring external teachers not naturally found in the applications environments, fear to tread.

Self-Stabilization of Speech Perception and Production Codes: New Light on Motor Theory

The insights gleaned from the design of ART 2 have also begun to clarify how hierarchical learning systems with multiple ART levels can be designed. Figure 11 schematizes a hierarchical ART system for learning to recognize and produce speech which self-stabilizes its learning in real-time without using a teacher. This ART architecture is being developed at Boston University by Michael Cohen, Stephen Grossberg, and David Stork. Top-down ART expectation mechanisms at several levels of this architecture help to self-stabilize learned codes and to self-organize the selection of invariant recognition properties. Of particular interest in this speech architecture is the role of top-down expectation signals from the architecture's articulatory, or motor, system to its auditory, or perception, system. These expectations help to explain classical results from motor theory, which state that speech is perceived in terms of how it would have been produced, even during passive listening.

The key insights of the motor theory take on new meaning in the theory through the self-stabilizing properties of top-down articulatory-to-auditory expectations. These expectations self-stabilize the learned imitative associative map that transforms the perceptual codes which represent heard speech into motor codes for generating spoken speech. In so doing, the articulatory-to-auditory expectations deform the bottom-up auditory STM patterns, via 2/3 Rule-like matching, into activation patterns which are consistent with invariant properties of the motor commands. These motorically-modified STM codes are then ecoded in long-term memory within a bottom-up adaptive filter within the auditory system itself. This bottom-up adaptive filter activates a partially compressed speech code at the auditory system's next processing level. The motorically-modified speech code is thus activated during passive listening as well as during active imitation.

Psychophysiological and Neurophysiological Predictions of ART

Although applications of ART to computer science depend upon the computational power of these systems for solving real-world problems, ART systems are also models of the biological processes whose analysis led to their discovery. In fact, in addition to suggesting mechanistic explanations of many interdisciplinary data about mind and brain, the theory has also made a number of predictions which have since been partially supported by experiments. For example, in 1976, it was predicted that both norepinephrine (NE) mechanisms and attentional mechanisms modulate the adaptive development of thalamocortical visual feature detectors. In 1976 and 1978, Kasamatsu and Pettigrew described NE modulation of feature detector development. Wolf Singer reported attentional modulation in 1982. In 1978, a word length effect in word recognition paradigms was predicted. In 1982 and 1983, Samuel, van Santen, and Johnston reported a word length effect in word superiority experiments. In 1978 and 1980, a hippocampal generator of the P300 event-related potential was predicted. In 1980, Halgren and his colleagues reported the existence of a hippocampal P300 generator in humans. The existence and correlations between other event-related potentials, such as processing negativity (PN), early positive wave (P120), and N200 were also predicted in these theoretical articles. These predictions and supportive data are described in several recent books[3,4].

Thus ART systems provide a fertile ground for gaining a new understanding of biological intelligence while suggesting novel computational theories and real-time adaptive neural network architectures with promising properties for tackling some of the outstanding problems in computer science and technology today.

REFERENCES

1. G.A. Carpenter and S. Grossberg, A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, **37**, 1987, 54–115.

2. G.A. Carpenter and S. Grossberg, ART 2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, December 1, 1987.

3. S. Grossberg (Ed.), **The Adaptive Brain, Volumes I and II.** Amsterdam: Elsevier/North-Holland, 1987.

4. S. Grossberg (Ed.), **Neural Networks and Natural Intelligence.** Cambridge, MA: MIT Press, 1988.

5. C. von der Malsburg, Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, **14**, 1973, 85–100.

6. S. Grossberg, Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors. *Biological Cybernetics*, **23**, 1976, 121–134.

7. S. Amari and A. Takeuchi, Mathematical theory on formation of category detecting nerve cells. *Biological Cybernetics*, **29**, 1978, 127–136.

8. E.L. Bienenstock, L.N. Cooper, and P.W. Munro, Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, **2**, 1982, 32–48.

9. T. Kohonen, **Self-Organization and Associative Memory.** New York: Springer-Verlag, 1984.

10. D.H. Ackley, G.E. Hinton, and T.J. Sejnowski, A learning algorithm for Boltzmann machines. *Cognitive Science*, **9**, 1985, 147–169.

11. D.E. Rumelhart, G.E. Hinton, and R.J. Williams, Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland (Eds.), **Parallel Distributed Processing.** Cambridge, MA: MIT Press, 1986.

12. J.S. Denker (Ed.), **Neural Networks for Computing.** New York: American Institute of Physics, 1986.

13. A.B. Gschwendtner, R.C. Harney, and R.J. Hull, Coherent IR radar technology. In D.K. Killinger and A. Mooradian (Eds.), **Optical and Laser Remote Sensing.** New York: Springer-Verlag, 1983.

14. D. Casasent and D. Psaltis, Position, rotation, and scale invariant optical correlations. *Applied Optics*, **15**, 1976, 1793–1799.

15. E.L. Schwartz, Computational anatomy and functional architecture of striate cortex: A spatial mapping approach to perceptual coding. *Vision Research*, **20**, 1980, 645–669.

TABLE 1

ART Architecture	Alternative Learning Properties
Real-time (on-line) learning	Lab-time (off-line) learning
Nonstationary world	Stationary world
Self-organizing (unsupervised)	Teacher supplies correct answer (supervised)
Memory self-stabilizes in response to arbitrarily many inputs	Capacity catastrophe in response to arbitrarily many inputs
Effective use of full memory capacity	Can only use partial memory capacity
Maintain plasticity in an unexpected world	Externally shut off plasticity to prevent capacity catastrophe
Learn internal top-down expectations	Externally impose costs
Active attentional focus regulates learning	Passive learning
Slow or fast learning	Slow learning or oscillation catastrophe
Learn in approximate-match phase	Learn in mismatch phase
Use self-regulating hypothesis testing to globally reorganize the energy landscape	Use noise to perturb system out of local minima in a fixed energy landscape
Fast adaptive search for best match	Search tree
Rapid direct access to codes of familiar events	Recognition time increases with code complexity
Variable error criterion (vigilance parameter) sets coarseness of recognition code in response to environmental feedback	Fixed error criterion in response to environmental feedback
All properties scale to arbitrarily large system capacities	Key properties deteriorate as system capacity is increased

FIGURE CAPTIONS

Figure 1. Stages of bottom-up activation: The input pattern I generates a pattern of STM activation $X = (x_1, x_2, \ldots, x_M)$ across F_1. Sufficiently active F_1 nodes emit bottom-up signals to F_2. This signal pattern S is multiplied, or gated, by long term memory (LTM) traces z_{ij} within the $F_1 \rightarrow F_2$ pathways. The LTM gated signals are summed before activating their target nodes in F_2. This LTM-gated and summed signal pattern T, where $T_j = \sum_i S_i z_{ij}$, generates a pattern of STM activation $Y = (x_{M+1}, \ldots, x_N)$ across F_2.

Figure 2. Matching by the 2/3 Rule: (a) A top-down expectation from F_2 inhibits the attentional gain control source as it subliminally primes target F_1 cells. Dotted outline depicts primed activation pattern. (b) Only F_1 cells that receive bottom-up inputs and gain control signals can become supraliminally active. (c) When a bottom-up input pattern and a top-down template are simultaneously active, only those F_1 cells that receive inputs from both sources can become supraliminally active. (d) Intermodality inhibition can shut off the F_1 gain control source and thereby prevent a bottom-up input from supraliminally activating F_1, as when attention shifts to a different input channel. Similarly, disinhibition of the F_1 gain control source in (a) may cause a top-down prime to become supraliminal, as during an internally willed fantasy.

Figure 3. ART 1 system: Two successive stages, F_1 and F_2, of the attentional subsystem encode patterns of activation in short term memory (STM). Bottom-up and top-down pathways between F_1 and F_2 contain adaptive long term memory (LTM) traces which multiply the signals in these pathways. The remainder of the circuit modulates these STM and LTM processes. Modulation by gain control enables F_1 to distinguish between bottom-up input patterns and top-down priming, or expectation, patterns, as well as to match these bottom-up and top-down patterns by the 2/3 Rule. Gain control signals also enable F_2 to react supraliminally to signals from F_1 while an input pattern is on. The orienting subsystem generates a reset wave to F_2 when sufficiently large mismatches between bottom-up and top-down patterns occur at F_1. This reset wave selectively and enduringly inhibits previously active F_2 cells until the input is shut off.

Figure 4. Alphabet learning: Code learning by ART 1 in response to the first presentation of the first 20 letters of the alphabet is shown. Two different vigilance levels were used, $\rho = .5$ and $\rho = .8$. Each row represents the total code that is learned after the letter at the left-hand column of the row is presented at F_1. Each column represents the learning, through time, of the top-down LTM vector, or expectation, corresponding to the F_2 node whose index is listed at the top of the column. These LTM vectors do not, in general, equal the input patterns which change them through learning. Instead, each expectation acts like a novel type of prototype for the entire set of practiced input patterns coded by that node, as well as for unfamiliar input patterns that share invariant properties with this set. The simulation illustrates the "fast learning" case, in which the altered LTM traces reach a new equilibrium in response to each new stimulus. Slow learning is more gradual than this.

Figure 5. ART 1 hypothesis testing cycle: (a) The input pattern I generates the STM activity pattern X at F_1 as it activates A. Pattern X both inhibits A and generates the bottom-up signal pattern S. Signal pattern S is transformed via the adaptive filter into the input pattern $T = ZS$, which activates the compressed STM pattern Y across F_2. (b) Pattern Y generates the top-down signal pattern U which is transformed by the top-down adaptive filter $V = \hat{Z}U$ into the expectation pattern V. If V mismatches I at F_1, then a new STM activity pattern X^* is generated at F_1. The reduction in total STM activity which occurs when X is transformed into X^* causes a decrease in the total inhibition from F_1 to A. (c) Then the input-driven activation of A can release a nonspecific

arousal wave to F_2, which resets the STM pattern Y at F_2. (d) After Y is inhibited, its top-down expectation is eliminated, and X can be reinstated at F_1. Now X once again generates input pattern T to F_2, but since Y remains inhibited T can activate a different STM pattern Y^* at F_2. If the top-down expectation due to Y^* also mismatches I at F_1, then the rapid search for an appropriate F_2 code continues.

Figure 6. Category grouping by ART 2 of 50 analog input patterns into 34 recognition categories. Each input pattern I is depicted by a graph as a function of abscissa values i $(i = 1 \ldots M)$, with successive ordinate I_i values connected by straight lines. The category structure established upon one complete presentation of the 50 inputs remains stable thereafter if the same inputs are presented again.

Figure 7. Lower vigilance implies coarser grouping. The same ART 2 system as used in Figure 6 has here grouped the same 50 inputs into 23 recognition categories. Note, for example, that Categories 4 and 32 of Figure 6 are joined in Category 4; Categories 8, 29, and 3 are joined in Category 28; and Categories 14, 15, and 31 are joined in Category 12.

Figure 8. A typical ART 2 architecture. Open arrows indicate specific patterned inputs to target nodes. Filled arrows indicate nonspecific gain control inputs. The gain control nuclei (large filled circles) nonspecifically inhibit target nodes in proportion to the L_2-norm of STM activity in their source fields. As in ART 1, gain control (not shown) coordinates STM processing with input presentation rate.

Figure 9. A three stage preprocessor for the ART 2 system enables input patterns that are deformed, shifted, dilated, and rotated to be recognized as exemplars of the same category. The preprocessor passes laser radar images that separate figure from background through a boundary segmentation network and then through a Fourier-Mellin transform. The Fourier-Mellin spectra are the inputs to ART 2.

Figure 10. A self-organizing architecture for invariant pattern recognition and recall that can be expanded, as noted in the text, to include reinforcement mechanisms capable of focussing attention upon internally desired classes of external events.

Figure 11. Schematic of some processing stages in an architecture for a self-organizing speech perception and production system. The left hand side of the figure depicts five stages of the auditory model, the right hand side four stages of the motor model. The pathways from the partially compressed auditory code to the motor system learn an imitative associative map which joins auditory feedback patterns to the motor commands that generated them. These motor commands are compressed via bottom-up and top-down adaptive filters within the motor system into motor synergies. The synergies read out top-down learned articulatory-to-auditory expectations, which select the motorically consistent auditory data for incorporation into the learned speech codes of the auditory system.

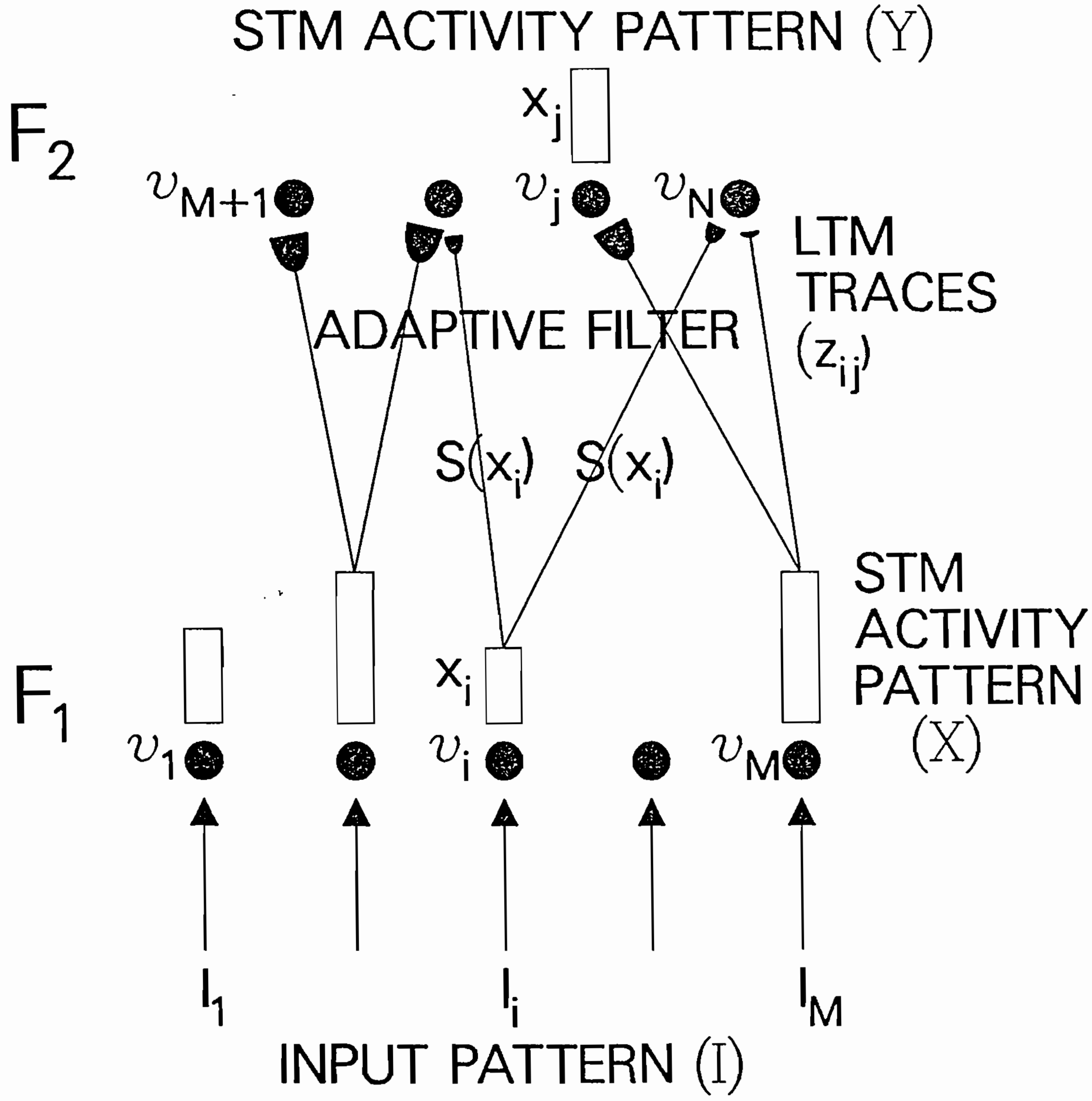

STM ACTIVITY PATTERN (Y)
x_j
F_2
v_{M+1}
v_j
v_N
LTM TRACES (z_{ij})
ADAPTIVE FILTER
$S(x_i)$
$S(x_i)$
STM ACTIVITY PATTERN (X)
x_i
F_1
v_1
v_i
v_M
I_1
I_i
I_M
INPUT PATTERN (I)

Figure 1

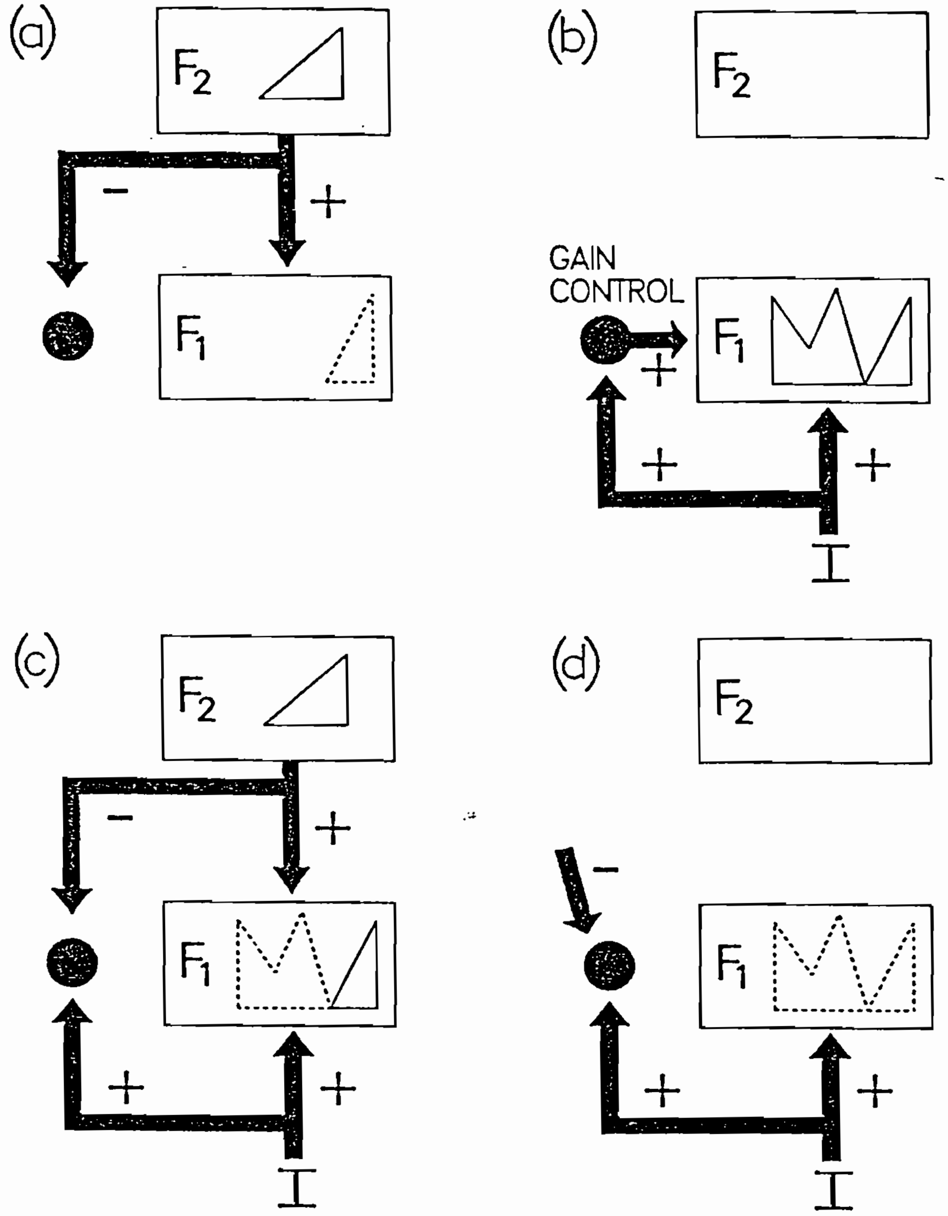


Figure 2

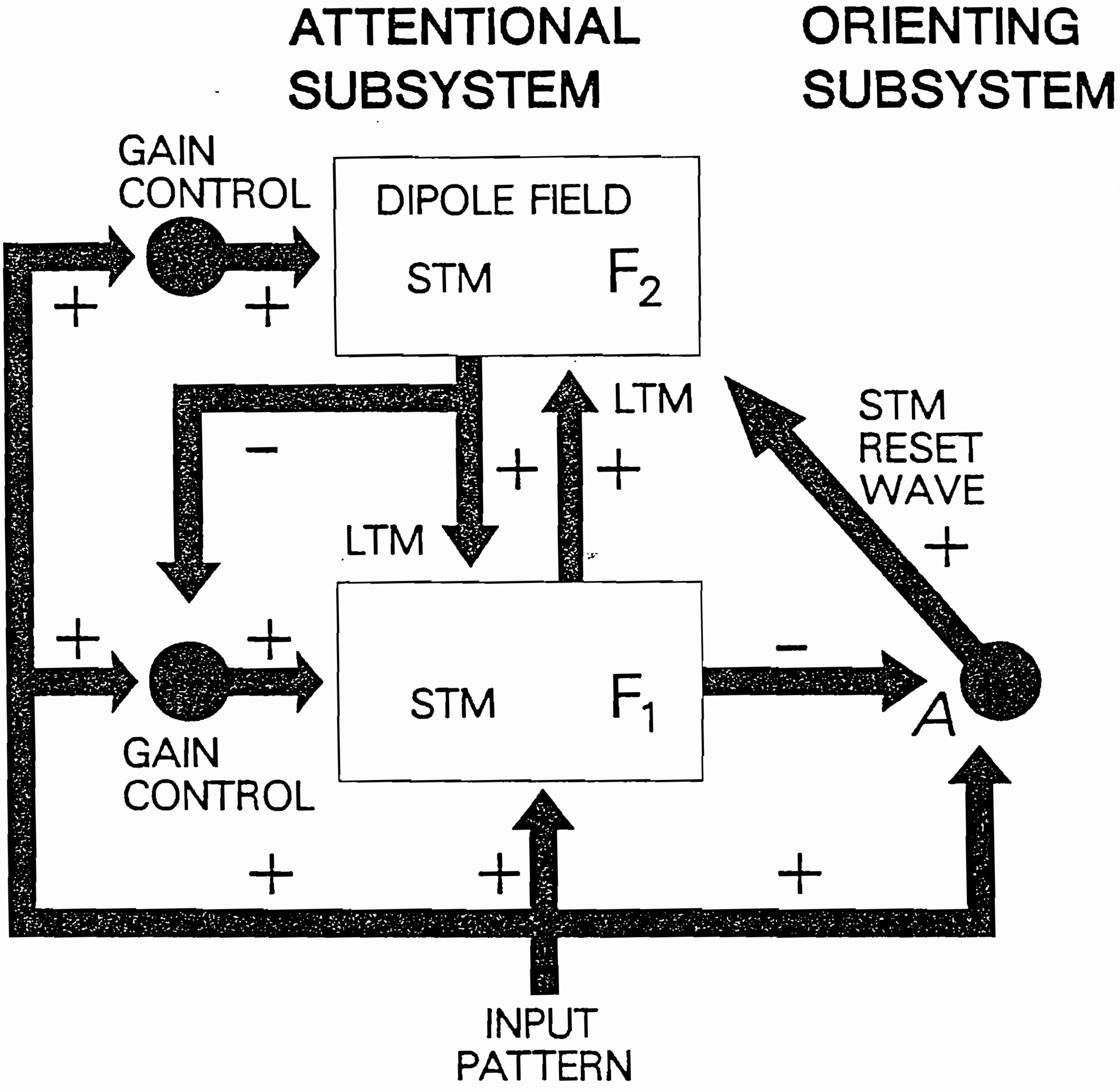
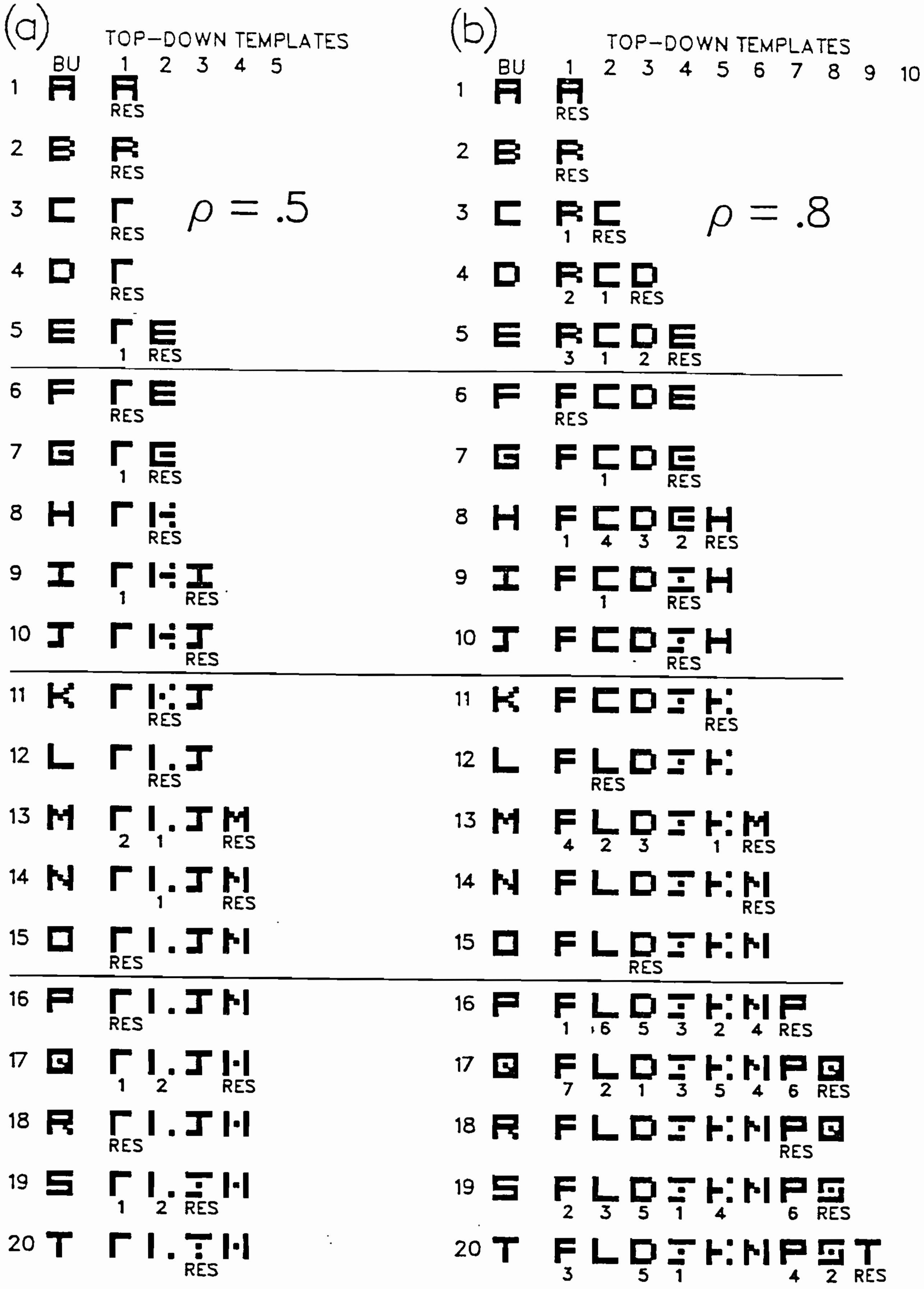

ATTENTIONAL SUBSYSTEM
ORIENTING SUBSYSTEM
GAIN CONTROL
DIPOLE FIELD
STM
F_2
+
+
−
+
+
LTM
LTM
STM RESET WAVE
+
+
+
−
STM
F_1
A
GAIN CONTROL
+
+
+
INPUT PATTERN

Figure 3



(a)
TOP-DOWN TEMPLATES
BU
1 2 3 4 5
ρ = .5
RES
(b)
TOP-DOWN TEMPLATES
BU
1 2 3 4 5 6 7 8 9 10
ρ = .8
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Figure 4

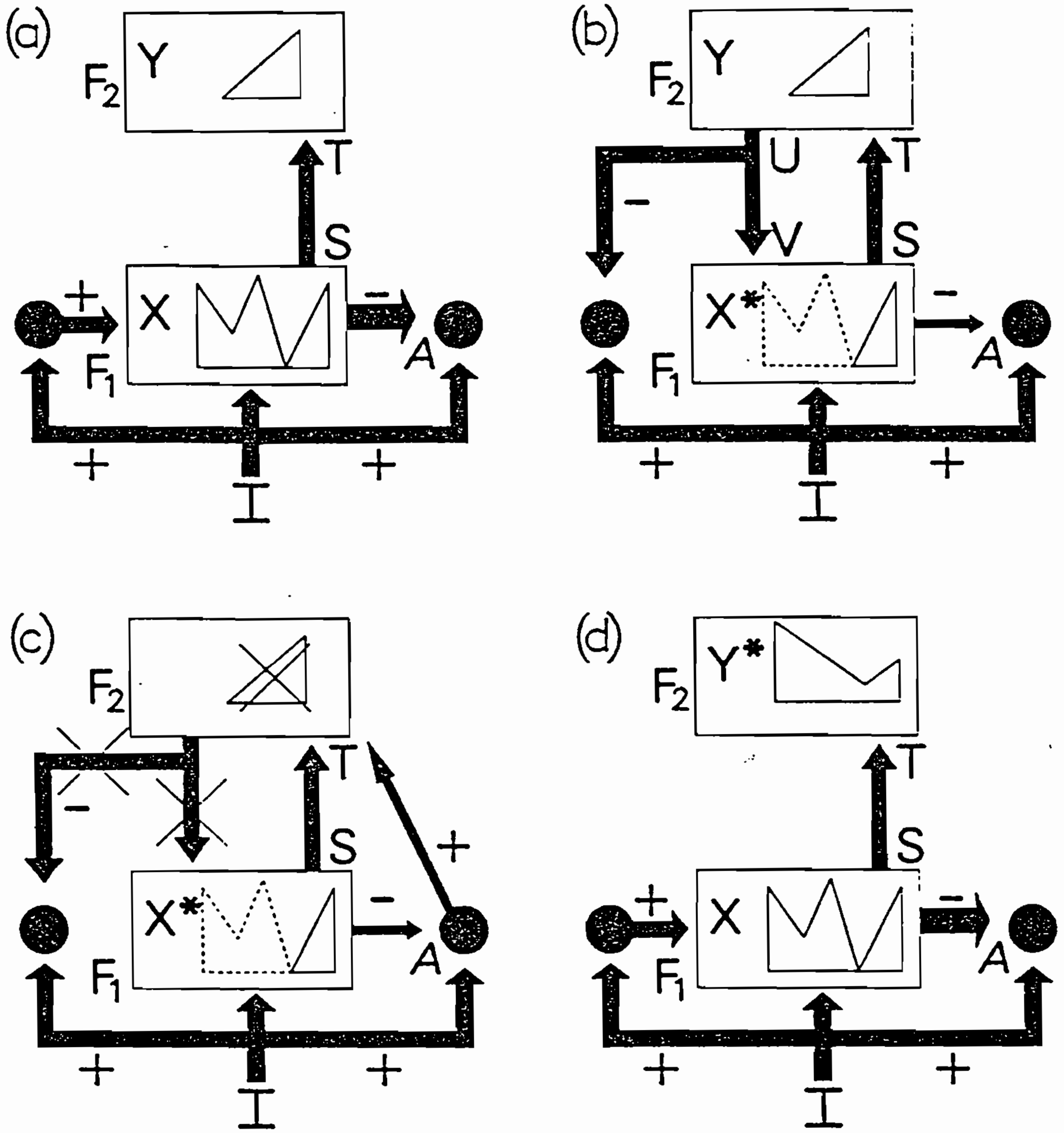


Figure 5

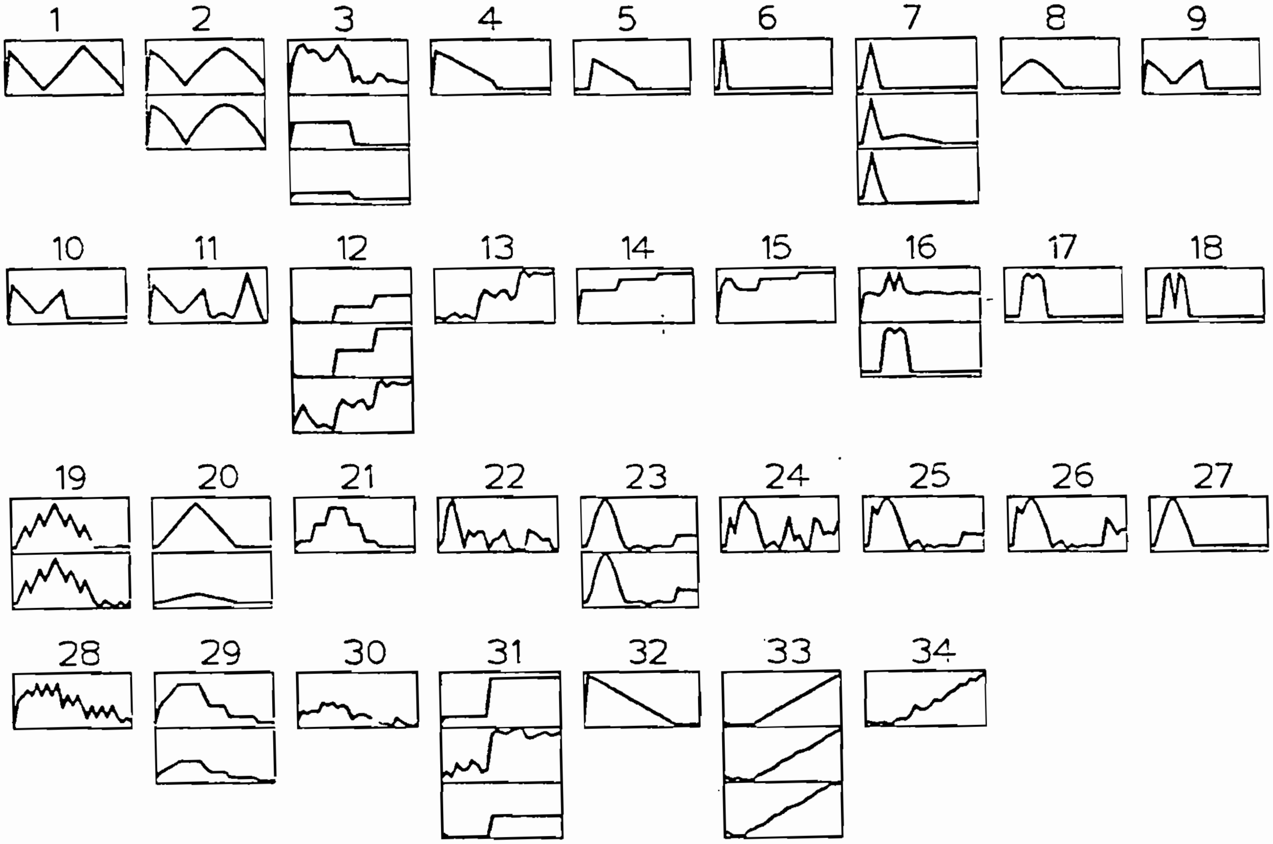

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34

Figure 6

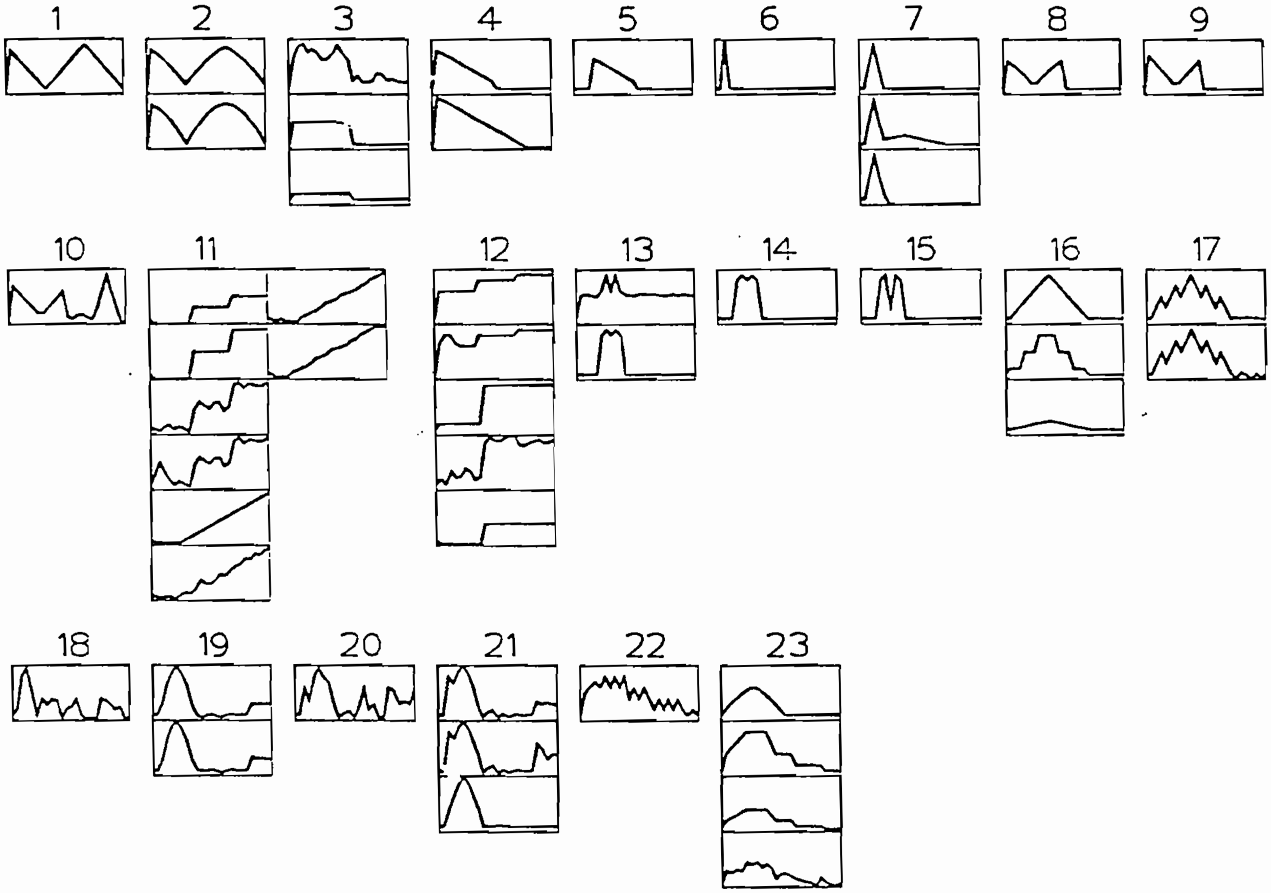


Figure 7

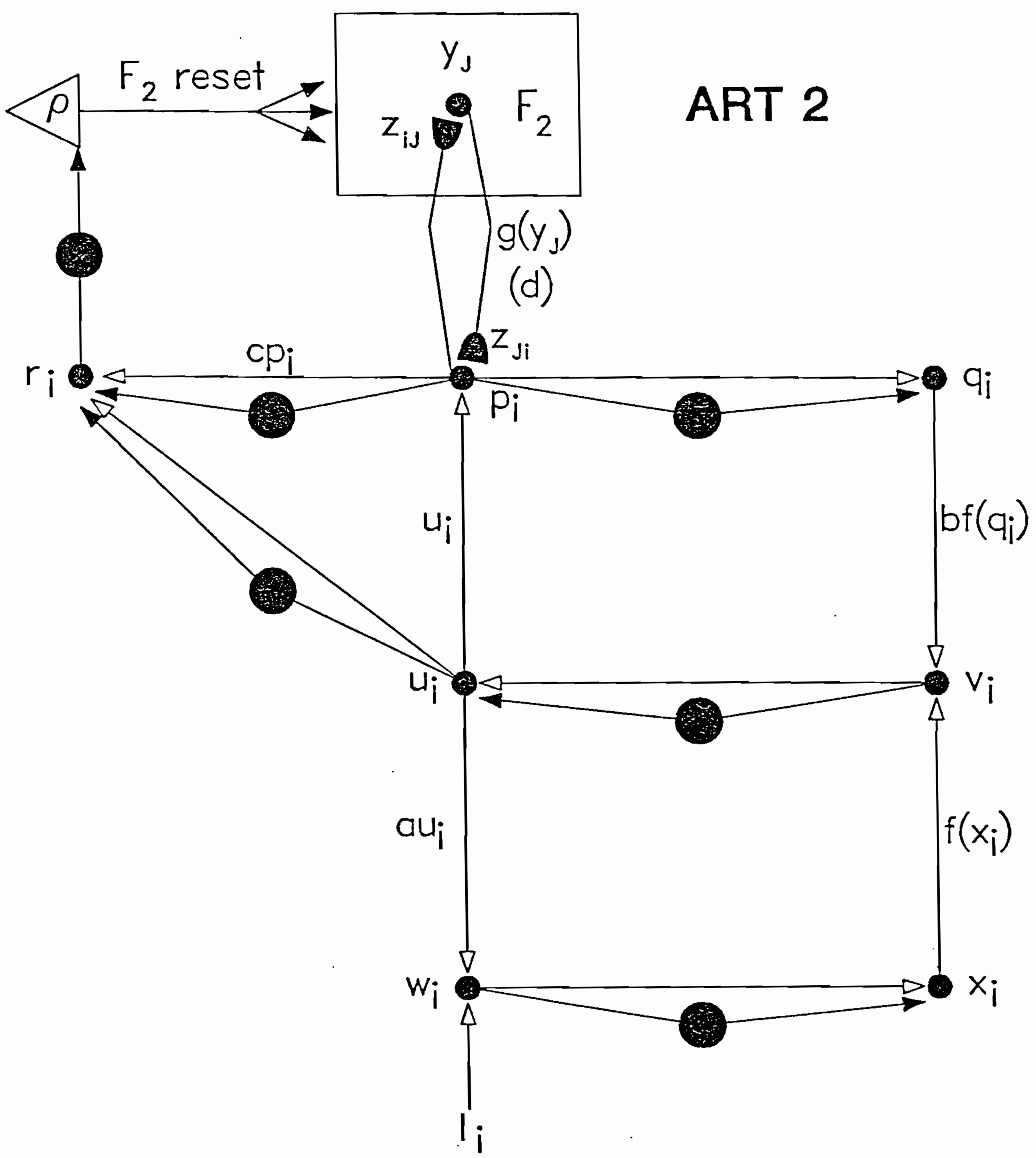

ART 2
F$_2$ reset
ρ
y_J
F_2
z_{iJ}
$g(y_J)$
(d)
z_{Ji}
cp_i
r_i
p_i
q_i
u_i
$bf(q_i)$
u_i
v_i
au_i
$f(x_i)$
w_i
x_i
I_i

Figure 8

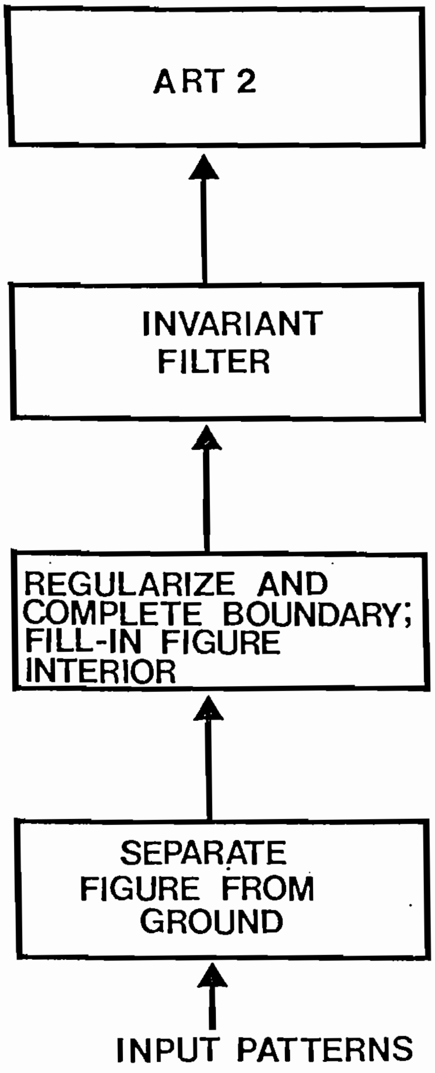

ART 2
INVARIANT FILTER
REGULARIZE AND COMPLETE BOUNDARY; FILL-IN FIGURE INTERIOR
SEPARATE FIGURE FROM GROUND
INPUT PATTERNS

Figure 9

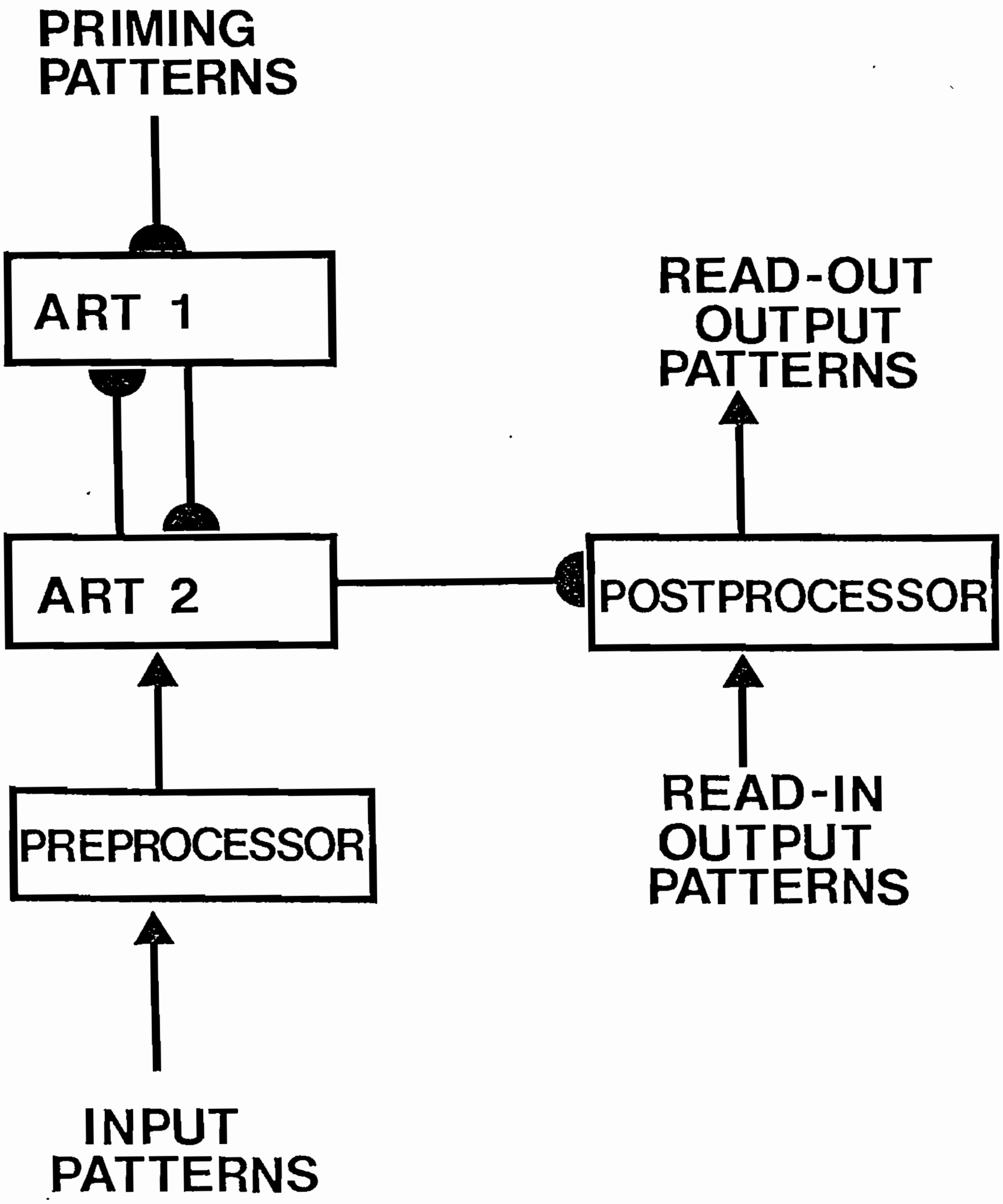

PRIMING PATTERNS
ART 1
READ-OUT OUTPUT PATTERNS
ART 2
POSTPROCESSOR
PREPROCESSOR
READ-IN OUTPUT PATTERNS
INPUT PATTERNS

Figure 10

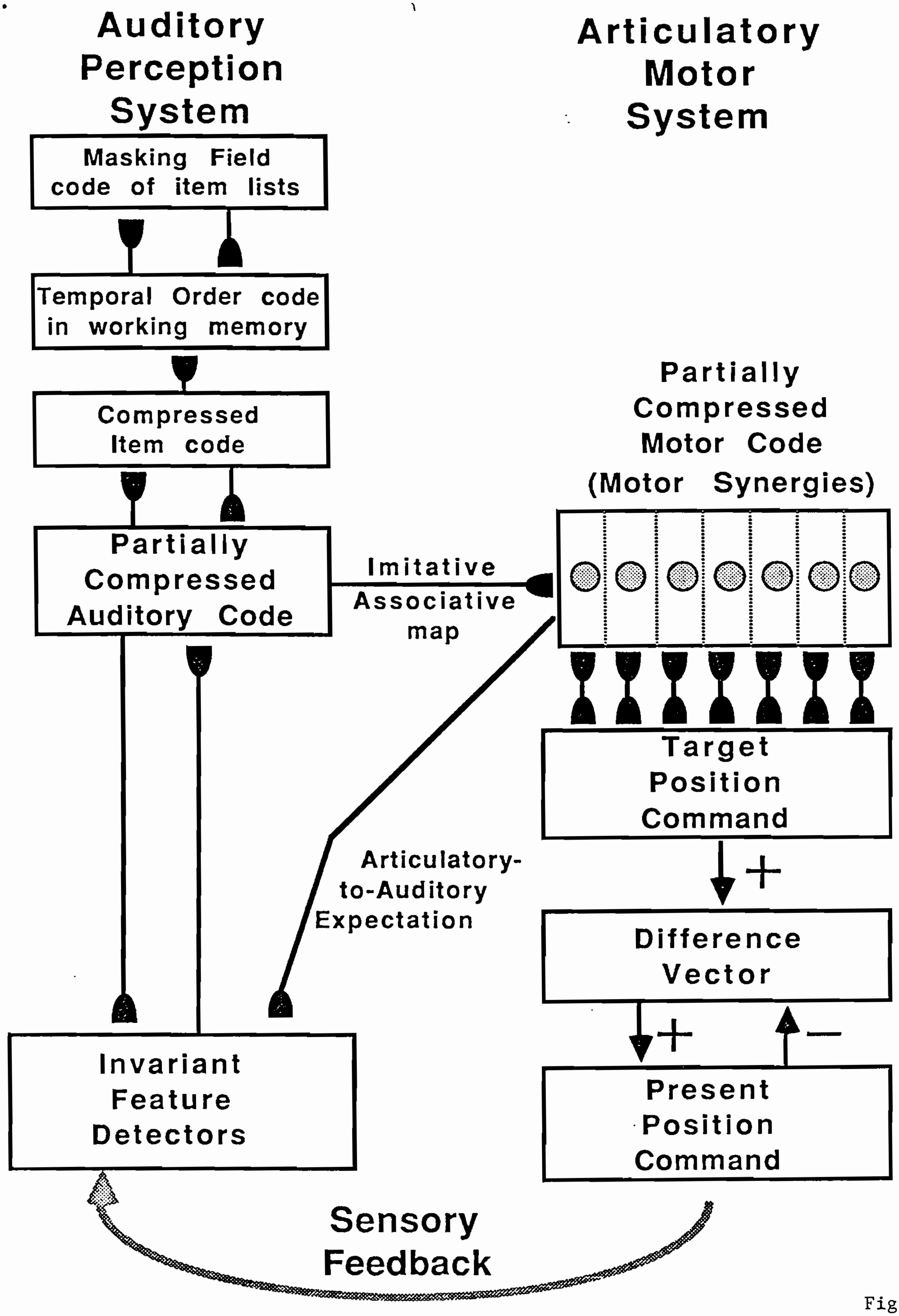

Auditory Perception System
Articulatory Motor System
Masking Field code of item lists
Temporal Order code in working memory
Compressed Item code
Partially Compressed Auditory Code
Imitative Associative map
Partially Compressed Motor Code (Motor Synergies)
Target Position Command
+
Difference Vector
+
−
Present Position Command
Articulatory-to-Auditory Expectation
Invariant Feature Detectors
Sensory Feedback

Figure 11